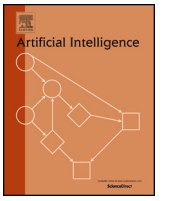
Artificial Intelligence 330 (2024) 104082

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



An extended view on lifting Gaussian Bayesian networks

Mattis Hartwig [a,b,*], Ralf Möller [a,c], Tanya Braun [d]

[a] German Research Centre for Artificial Intelligence, Ratzeburger Allee 160, Lübeck, 23562, Schleswig-Holstein, Germany
[b] singularIT GmbH, Hainstraße 1-3, Leipzig, 04109, Saxony, Germany
[c] Institute of Information Science, University of Lübeck, Ratzeburger Allee 1602, Lübeck, 23562, Schleswig-Holstein, Germany
[d] Computer Science Department, University of Münster, Einsteinstraße 62, Münster, 48149, North-Rhine Westfalia, Germany

ARTICLE INFO

Keywords:
Lifting
Gaussian Bayesian networks
Exact inference

ABSTRACT

Lifting probabilistic graphical models and developing lifted inference algorithms aim to use higher level groups of random variables instead of individual instances. In the past, many inference algorithms for discrete probabilistic graphical models have been lifted. Lifting continuous probabilistic graphical models has played a minor role. Since many real-world applications involve continuous random variables, this article turns its focus to lifting approaches for Gaussian Bayesian networks. Specifically, we present algorithms for constructing a lifted joint distribution for scenarios of sequences of overlapping and non-overlapping logical variables. We present operations that work in a fully lifted way including addition, multiplication, and inversion. We present how the operations can be used for lifted query answering algorithms and extend the existing query answering algorithm by a new way of evidence handling. The new way of evidence handling groups evidence that has the same effect on its neighboring variables in cases of partial overlap between the logical-variable sequences. In the theoretical complexity analysis and the experimental evaluation, we show under which conditions the existing lifted approach and the new lifted approach including evidence grouping lead to the most time savings compared to the grounded approach.

1. Introduction

Probabilistic graphical models (PGMs) use the language of graphs to represent probability distributions over random variables (randvars), where the graph structure encodes conditional (in)dependencies between the randvars. Bayesian networks (BNs) are the most used subclass of PGMs [1]. When real-world applications require PGMs to model the behavior of individuals (as randvar instances), the graphs get very large and hard to handle. The current pandemic of COVID-19 gives a good example in which we would ideally model the behavior of individual patients or persons to estimate the risk of infection for each person individually. However, no inference algorithm for a PGM could handle a model representing eight billion people, each having possibly multiple randvars and connections to other people in a timely and accurate manner. Therefore, Poole has introduced first-order probabilistic inference [2] as a mechanism to exploit symmetries in a model: Indistinguishable instances are combined into groups using logical variables (logvars). First-order inference then works with representatives of these groups. Using a compact first-order representation is also referred to as lifting and has been an active research field in the past years [3–5].

* Corresponding author.
E-mail address: mattis.hartwig@singular-it.de (M. Hartwig).

https://doi.org/10.1016/j.artint.2024.104082
Received 30 July 2021; Received in revised form 26 February 2023; Accepted 21 January 2024

Available online 6 February 2024

Lifted inference algorithms have been first developed for discrete PGMs. Since many real-world use cases require the use of continuous variables, we have transferred the lifting approach to continuous, specifically Gaussian, Bayesian networks. In this article, we bring together the first published approach that does not allow overlaps between logvar sequences [6] with the more general approach that can also handle logvar sequences [7] and extend it with a new evidence grouping mechanism in the query answering algorithm. In this article, we make the following contributions:

1. We expose the connection between the non-matrix notation used in [6] with the matrix notation used in [7].
2. We showcase all operations developed in the article using a full running example.
3. We provide an extended derivation of existing algorithms allowing for better readability and reproducibility.
4. We present a new version of the query answering algorithm that groups evidence in cases of partial overlaps, including a definition of liftability for queries and evidence in the continuous setting as well as an updated complexity analysis and empirical evaluation.

The first three contributions allow for better access to the research around lifting in Gaussian Bayesian networks. The fourth contribution is a new algorithm allowing for more efficient query answering in certain scenarios (see Section 9) including an updated complexity analysis and empirical evaluation. In the empirical evaluation, we confirm our theoretical complexity analysis by conducting four experiments on different sub-graphs and queries derived from our running example. The experiments focus on the influence of increasing domain sizes on runtime using both ground and lifted approaches in four scenarios. The liftability definition, the first formal one for continuous inference to the best of our knowledge, provides an interesting case as it varies from liftability notions for queries in the discrete setting. After its acceptance with minor reviews, this paper has also become part of a dissertation.

The remainder of this paper is structured as follows. We begin by introducing the preliminaries for GBNs and lifting in Section 2, followed by an overview of related work in Section 3. In Section 4, we introduce a running example used as an illustration in all upcoming sections. In Section 5, we describe the algorithm for constructing the lifted joint distribution with special emphasis on the connection between the two approaches in [6] and [7]. In Section 6, we present operations for addition, multiplication and inversion to work with the lifted joint distribution. Then, we introduce the algorithms for query answering in lifted GBNs and present our new algorithms that uses evidence grouping to speed up the run-time in partial overlap cases in Section 7. We evaluate our developed algorithms with a complexity analysis in Section 8 and an experimental evaluation along the running example in Section 9. We conclude with a discussion and outlook to further research in Section 10.

2. Fundamental concepts and notation

This section covers preliminaries of lifting GBNs. In slight abuse of notation, we use set operations for sequences, where we apply the operation to the set containing the sequence elements but keep the sequence order intact.

2.1. Probabilistic graphical models

The fundamental structure of a PGM is a graph. PGMs model the stochastic behavior of randvars, i.e., the vertices in PGMs represent these randvars and the edges describe the relationships between randvars. In general, PGMs serve as a compact representation of the joint probability distribution over randvars. Since we are only using graphs in the form of PGMs, we use the same symbol X for vertices and randvars. There are different types of PGMs but the one used in this article is the BN. For the formal definitions, we follow [8]. At the end of this part, we provide a small example.

Definition 1 *(Factorization, Bayesian network).* Let $\mathcal{G}$ be a directed acyclic graph whose vertices represent randvars $\boldsymbol{X} = \{X_1, ..., X_N\}$ that each have a range $R(X_i)$ of values that can be assigned to X_i. Analogously, we use $R(\boldsymbol{X})$ to refer to the cross product of the ranges of $\boldsymbol{X}$. A probability distribution P over the same space $\boldsymbol{X}$ factorizes according to $\mathcal{G}$ if P can be expressed as a product

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i | \boldsymbol{Pa}(X_i)). \tag{1}$$

where $\boldsymbol{Pa}(X_i)$ refers to the *parents* of a randvar X_i, i.e., all X_k that have a directed edge to X_i. Let each vertex in $\mathcal{G}$ be annotated with a conditional probability table representing $P(X_i | \boldsymbol{Pa}(X_i))$. Then, a *Bayesian network* is a pair $(\mathcal{G}, P)$, where P factorizes over $\mathcal{G}$.

The BN structure encodes conditional independencies between randvars. Any randvar X_i is conditionally independent of all non-descending randvars given its parents $\boldsymbol{Pa}(X_i)$. For iterating over a set of randvars that are part of a BN, we use a topological ordering of the randvars.

Definition 2 *(Topological ordering).* Let $\boldsymbol{X}$ be a set of randvars within a BN. A topological ordering of the randvars in $\boldsymbol{X}$ is any ordering that ensures that a child node X_i comes after its parent node X_k. In a BN, there always exists at least one topological ordering.

Having introduced a representation for a joint probability distribution, we now define how to use the distribution with query answering.

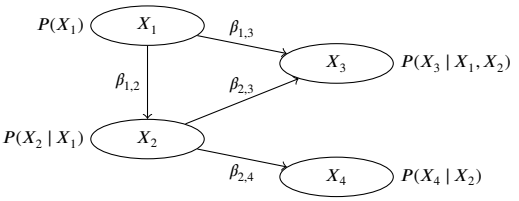


Fig. 1. Example (G)BN with four randvars.

Definition 3 *(Query).* A *query* $P(\boldsymbol{Q}|\boldsymbol{E}=\boldsymbol{e})$ consists of a query set $\boldsymbol{Q} \subseteq \{X_1, ...X_N\}$, and a set of events $\boldsymbol{E}=\boldsymbol{e}$ with $\boldsymbol{E} \subseteq \{X_1, ...X_N\}$ and $\boldsymbol{e} \in R(\boldsymbol{E})$.

Based on Definition 3, query answering in PGMs allows for two different types of queries: We use the term *marginal query* for queries that do not contain any evidence, i.e., $P(\boldsymbol{Q})$, and the term *conditional query* for queries that contain evidence, i.e., $P(\boldsymbol{Q}|\boldsymbol{E}=\boldsymbol{e})$. To answer a query for a BN, a possible way is to eliminate all randvars that do not occur in the query by summing over their respective range values (marginalizing out).

Example 1. Fig. 1 shows a BN with four randvars $X_1, \ldots, X_4$ and corresponding conditional probability distributions assigned to each node. There exist two possible topological orderings in the BN, (X_1, X_2, X_3, X_4) and (X_1, X_2, X_4, X_3). Assume that the randvar ranges are Boolean. A marginal query in the BN is given by $P(X_4)$, which requires marginalizing out X_1, X_2, X_3 by summing over their Boolean ranges. A conditional query is given by $P(X_2 \mid X_3 = \mathit{true})$, requiring the marginalization of X_1, X_4.

2.2. The (multivariate) Gaussian distribution

The distribution of a single Gaussian randvar X_i follows the probability density function:

$$f_{X_i}(x_i) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left(-\frac{1}{2\sigma^2}(x_i - \mu)^2\right) \tag{2}$$

where μ is the mean and σ^2 the standard deviation. In short, we write $X_i \sim \mathcal{N}\left(\mu, \sigma^2\right)$. The N-dimensional multivariate Gaussian distribution over the values $\boldsymbol{x}$ of a set of N randvars $\boldsymbol{X} = \{X_1, ..., X_N\}$ follows the multivariate probability density function:

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{N}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right) \tag{3}$$

with $\boldsymbol{\mu}$ being a N-dimensional mean vector and $\boldsymbol{\Sigma}$ being a $N \times N$-dimensional symmetric covariance matrix. In short, we write again $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Gaussian distributions are easy to use for calculations because they are closed under multiplication and the integral can be calculated analytically. Both characteristics are useful for working with probabilities. The answer for a conditional probability query $P(\boldsymbol{Q} \mid \boldsymbol{E} = \boldsymbol{e})$ is again a Gaussian distribution $P(\boldsymbol{Q} \mid \boldsymbol{E} = \boldsymbol{e}) = \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$, where $\boldsymbol{\mu}^*$ and $\boldsymbol{\Sigma}^*$ can be calculated analytically with

$$\boldsymbol{\mu}^* = \boldsymbol{\mu}_{\boldsymbol{Q}} + \boldsymbol{\Sigma}_{\boldsymbol{QE}} \boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1} (\boldsymbol{e} - \boldsymbol{\mu}_{\boldsymbol{E}}), \tag{4}$$

$$\boldsymbol{\Sigma}^* = \boldsymbol{\Sigma}_{\boldsymbol{QQ}} - \boldsymbol{\Sigma}_{\boldsymbol{QE}} \boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{EQ}}. \tag{5}$$

The subscripts containing sets refer to the subset of entries in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ corresponding to the randvars in given sets.

2.3. Gaussian probabilistic graphical models

Using continuous randvars in a BN is not trivial. Marginalizing out randvars in the continuous case translates to integrating out the randvars that are to be marginalized. In the general form, the problem of solving those integrals quickly gets intractable, which motivates more restrictive forms that avoid intractability. Shachter and Kenley have introduced Gaussian Bayesian networks (GBNs) under the name Gaussian influence diagrams as a possibility to have continuously distributed randvars in a BN that allows for tractable inference [9], which we define next.

Definition 4 *(Gaussian Bayesian network).* A *Gaussian Bayesian network* is a BN where all randvars are continuous, i.e., $R(X_i) = \mathbb{R}$, and normally distributed. The edges represent linear relationships between the randvars. As in BNs, the joint density can be factorized using the conditional probability densities of X_i with $i = 1, ..., N$ given it parents $\boldsymbol{Pa}(X_i)$:

$$P(X_i | \boldsymbol{Pa}(X_i) = \boldsymbol{x}) \sim \mathcal{N}\left(\mu_i + \sum_{X_k \in \boldsymbol{Pa}(X_i)} \beta_{k,i}(x_k - \mu_k), \sigma_i^2\right), \tag{6}$$

where μ_k and μ_i are node means, σ_i^2 is the node variance, $\beta_{k,i}$ represents the influence of parent X_k on its child X_i, and x_k refers to the observed value for X_k.

Example 2. Interpreting the BN in Fig. 1 as a GBN means that the randvars $X_1, \dots, X_4$ are continuous and normally distributed, with the ranges of $X_1, \dots, X_4$ no longer being Boolean but $\mathbb{R}$. Each X_i has its own mean μ_i and variance σ_i^2. Linear dependencies between parent and child nodes exist as follows: $\beta_{1,2}, \beta_{1,3}, \beta_{2,3}, \beta_{2,4}$.

Later in the article, we also write μ_{X_i} instead of μ_i. Whenever it is clear to which randvar the mean value is referring, we use the short form. Remark: The linear influence β is here multiplied with the parent's deviation from the mean $(x_k - \mu_k)$, which is following the structure in [9]. Other authors multiply the linear factor directly with the variable value μ_k [8]. Using the deviation has the benefit that the marginal means are equivalent to the node means, whereas using the variable name requires a further conversion step. Semantically, both structures can be converted into multivariate Gaussians as given in Eq. (3). Before we show how to convert a Gaussian BN into a multivariate Gaussian, we need to define the following function, which becomes important for the conversion:

Definition 5 *(Kronecker delta).* Given two items A and B, such as indexes or randvars, the Kronecker delta function $\delta_{A,B}$ is defined as

$$\delta_{A,B} = \begin{cases} 1 & \text{if } A = B \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

Analogously to a discrete BN, a GBN is a sparse representation of the multivariate Gaussian distribution in Eq. (3), which can be constructed from a GBN using an algorithm by Shachter and Kenley [9], which can be formulated in matrix and in non-matrix notation. Using non-matrix notation, we iterate two times over the randvars in the GBN and use the following equation for calculating the entries within the covariance matrix inductively,

$$\Sigma_{i,j} = \sum_{X_k \in \boldsymbol{Pa}(X_j)} \Sigma_{i,k}\beta_{k,j} + \delta_{i,j}\sigma_i^2, \tag{8}$$

where $i, j \in 1, \dots, N$, $i < j$, and δ is the Kronecker delta function as defined in Definition 5.

Example 3 *(Construction of the covariance matrix).* Given the GBN of Fig. 1, we use Eq. (8) to calculate the joint distribution inductively:

$$\begin{aligned}
&\Sigma_{1,1} = \sigma_1^2 && \Sigma_{2,1} = \Sigma_{1,2} = \Sigma_{1,1}\beta_{1,2} && \Sigma_{3,1} = \Sigma_{1,3} = \Sigma_{1,1}\beta_{1,3} + \Sigma_{1,2}\beta_{2,3} && \Sigma_{4,1} = \Sigma_{1,4} = \Sigma_{1,2}\beta_{2,4} \\
&&& \Sigma_{2,2} = \Sigma_{2,1}\beta_{1,2} + \sigma_2^2 && \Sigma_{3,2} = \Sigma_{2,3} = \Sigma_{2,1}\beta_{1,3} + \Sigma_{2,2}\beta_{2,3} && \Sigma_{4,2} = \Sigma_{2,4} = \Sigma_{2,2}\beta_{2,4} \\
&&&&& \Sigma_{3,3} = \Sigma_{3,1}\beta_{1,3} + \Sigma_{3,2}\beta_{2,3} + \sigma_3^2 && \Sigma_{4,3} = \Sigma_{3,4} = \Sigma_{3,2}\beta_{2,4} \\
&&&&&&& \Sigma_{4,4} = \Sigma_{4,2}\beta_{2,4} + \sigma_4^2
\end{aligned}$$

Using matrix notation as an alternative, we store the linear dependencies in a matrix $\boldsymbol{T}$, where an entry $T_{i,j}$ contains the value of $\beta_{i,j}$, describing the linear relationship between parent X_i and child X_j. A zero entry is equivalent to no direct edge between the randvars in the GBN. In matrix notation, we can then calculate the off-diagonal entries of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{i},j}$ belonging to a randvar X_j in one step by

$$\boldsymbol{\Sigma}_{\boldsymbol{i},j} = \boldsymbol{\Sigma}_{\boldsymbol{i},\boldsymbol{i}}\boldsymbol{T}_{\boldsymbol{i},j} \tag{9}$$

and $\boldsymbol{\Sigma}_{j,\boldsymbol{i}}$ by its transpose

$$\boldsymbol{\Sigma}_{j,\boldsymbol{i}} = \boldsymbol{\Sigma}_{\boldsymbol{i},j}^T, \tag{10}$$

where $\boldsymbol{i} = \{1, ..., j-1\}$. The on-diagonal entries are calculated with

$$\Sigma_{j,j} = \boldsymbol{\Sigma}_{j,\boldsymbol{i}}\boldsymbol{T}_{\boldsymbol{i},j} + \sigma_j^2. \tag{11}$$

The starting point is

$$\Sigma_{1,1} = \sigma_1^2. \tag{12}$$

Algorithm 1 shows how to use Eqs. (9) to (12) for constructing the covariance matrix in a for-loop. In Example 4, we look at the BN in Fig. 1 from a matrix notation viewpoint.

Algorithm 1 Converting a GBN into a multivariate Gaussian joint distribution (matrix notation).

procedure CONSTRUCTJOINT(Topologically sorted randvars $\boldsymbol{X}$)
  $N \leftarrow |\boldsymbol{X}|$
  Initialize $N \times N$-dimensional covariance matrix $\boldsymbol{\Sigma}$
  $\Sigma_{1,1} \leftarrow \sigma_{X_1}$
  for $j \in 2, ..., N$ **do**
    $\boldsymbol{i} \leftarrow (1, \ldots, j-1)$
    $\boldsymbol{\Sigma}_{\boldsymbol{i},j} \leftarrow \boldsymbol{\Sigma}_{\boldsymbol{i},\boldsymbol{i}} \boldsymbol{T}_{\boldsymbol{i},j}$
    $\boldsymbol{\Sigma}_{j,\boldsymbol{i}} \leftarrow \boldsymbol{\Sigma}_{\boldsymbol{i},j}^T$
    $\Sigma_{j,j} \leftarrow \boldsymbol{\Sigma}_{j,\boldsymbol{i}} \boldsymbol{T}_{\boldsymbol{i},j} + \sigma_{X_j}$
  return $\boldsymbol{\Sigma}$

Example 4 *(Construction of the covariance matrix with matrix notation).* Converting Example 3 into matrix notation results in

$$\Sigma_{1,1} = \sigma_1^2 \qquad \Sigma_{2,1} = \Sigma_{1,2} = \Sigma_{1,1} T_{1,2} \qquad \boldsymbol{\Sigma}_{3,1:2} = \boldsymbol{\Sigma}_{1:2,3} = \boldsymbol{\Sigma}_{1:2,1:2} \boldsymbol{T}_{1:2,3} \qquad \boldsymbol{\Sigma}_{4,1:3} = \boldsymbol{\Sigma}_{1:3,4} = \boldsymbol{\Sigma}_{1:3,1:3} \boldsymbol{T}_{1:3,4}$$
$$\Sigma_{2,2} = \Sigma_{2,1} T_{1,2} + \sigma_2^2 \qquad \Sigma_{3,3} = \boldsymbol{\Sigma}_{3,1:2} \boldsymbol{T}_{1:2,3} + \sigma_3^2 \qquad \Sigma_{4,4} = \boldsymbol{\Sigma}_{4,1:3} \boldsymbol{T}_{1:3,4} + \sigma_4^2,$$

where

$$\boldsymbol{T} = \begin{bmatrix} 0 & \beta_{1,2} & \beta_{1,3} & 0 \\ 0 & 0 & \beta_{2,3} & \beta_{2,4} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

2.4. Parameterized random variables

Parameterized randvars (PRVs) allow for grouping indistinguishable randvars, which share the same characteristics in terms of node means, node variances and parent structure and as such are isomorphic in the BN. Logvars identify groups and PRVs to represent a set of indistinguishable randvars. The aim of lifting is to work solely with the PRVs as representatives for the randvars belonging to the PRV. The model where all PRVs are instantiated with the logvar constants is called the ground model and the model that only contains PRVs the lifted model.

Definition 6 *(Parameterized random variable, grounding).* Let $\mathcal{X}$ be a set of randvars and $\mathcal{L}$ be a sequence of logvars with a fixed ordering. A *parameterized random variable* Y is a syntactical construct of a randvar $X \in \mathcal{X}$ combined with a sequence of logvars $\boldsymbol{L} \subseteq \mathcal{L}$ that obey the fixed ordering in $\mathcal{L}$ into $X(\boldsymbol{L})$ to represent a set of randvars. If $\boldsymbol{L} = \emptyset$, the PRV is parameterless and constitutes a propositional randvar. The domain $\boldsymbol{D}(L) = \{l^1, ..., l^H\}$ contains the constants of logvar L. The domain of a sequence of logvars is defined as $\boldsymbol{D}(\boldsymbol{L}) = \times_{L \in \boldsymbol{L}} \boldsymbol{D}(L)$. The term $lv(Y)$ refers to the logvars of Y, i.e., $\boldsymbol{L}$. *Grounding* a PRV $Y = X(L)$ with a logvar L results in a set of randvars $\boldsymbol{gr}(Y) = \{X(l^1), ..., X(l^H)\}$.

The number of randvars represented by a PRV Y is determined by the domain size of its logvars, i.e., $|\boldsymbol{D}(lv(Y))|$. A propositional randvar X can be interpreted as a PRV $Y = X(L)$ with $|\boldsymbol{D}(L)| = 1$. The range of Y, $R(Y)$, contains the possible values for randvars $\boldsymbol{gr}(Y)$ analog to the range of individual randvars $R(X)$. The ordering of logvars places no restriction on the expressivity of a model as one can always introduce a new logvar with the same domain for a logvar that would usually occur out of order. To switch between randvars and PRVs, we use two helper functions.

Definition 7 *(One- and lif-function).* For a PRV Y and a randvar X with $X \in \boldsymbol{gr}(Y)$, the function $one(Y)$ returns any randvar in $\boldsymbol{gr}(Y)$, e.g., X. The function $lif(X)$ returns the corresponding PRV, here, $lif(X) = Y$.

2.5. Parameterized Gaussian Bayesian networks

Instead of N ground randvars $X_1, \ldots, X_N$, a parameterized GBN contains M PRVs $Y_1, \ldots, Y_M$. Each PRV Y_s, with $s = 1, ..., M$, has a lifted node mean η_s and lifted node variance λ_s that apply to all randvars in $\boldsymbol{gr}(Y_s)$, i.e.,

$$\forall X \in \boldsymbol{gr}(Y) : \eta_Y = \mu_X \wedge \lambda_Y = \sigma_X^2. \tag{13}$$

Edges are defined between PRV nodes. Grounding a topologically ordered list of PRVs results in a topologically ordered list of ground randvars because the parent-child relationships for each ground randvar are defined by the parent-child relationships of the corresponding PRVs. An edge between a parent Y_u and a node Y_s has a corresponding variable $\zeta_{Y_u,Y_s} \neq 0$ that describes the linear relationship between the PRVs analogously to the β in propositional GBNs of Definition 4. Given a node PRV Y_s and a child PRV Y_t, the logvar sequences can be either disjoint or overlapping resulting in different conditional independencies between parent and child randvars.

Disjoint logvar sequences: The logvars $\boldsymbol{L}_s$ of the parent PRV Y_s and the logvars $\boldsymbol{L}_t$ of the child PRV Y_t are disjoint, i.e. $\boldsymbol{L}_s \cap \boldsymbol{L}_t = \emptyset$. Disjoint logvar sets result in a relation from every randvar in $\boldsymbol{gr}(Y_s)$ to every randvar in $\boldsymbol{gr}(Y_t)$.

Overlapping logvar sequences: The logvars $\boldsymbol{L}_s$ of the parent PRV Y_s and the logvars $\boldsymbol{L}_t$ of the child PRV Y_t overlap, i.e., $\boldsymbol{L}_o = \boldsymbol{L}_s \cap \boldsymbol{L}_t \neq \emptyset$. Grounding results in a relation where each child node is influenced by all $|\boldsymbol{D}(\boldsymbol{L}_s \setminus \boldsymbol{L}_t)|$ parents that share the same grounding of $\boldsymbol{L}_o$.

In the course of this article, we need the following helper functions regarding logvar sequences.

Definition 8 *(dim- and ov-function).* Let L be a logvar and $\boldsymbol{L}_s \subseteq \mathcal{L}$ and $\boldsymbol{L}_t \subseteq \mathcal{L}$ two sequences of logvars. Then, we define

$$dim(L, \boldsymbol{L}_s) = \begin{cases} |\boldsymbol{D}(L)| & \text{if } L \in \boldsymbol{L}_s \\ 1 & \text{otherwise} \end{cases} \tag{14}$$

$$ov(L, \boldsymbol{L}_s, \boldsymbol{L}_t) = \begin{cases} 1 & \text{if } L \in (\boldsymbol{L}_s \cap \boldsymbol{L}_t) \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

3. Related work

The notion of lifting can be applied to the modeling formalism as well as the inference approach. Both aspects were first introduced to PGMs and probabilistic inference by Poole in 2003 [2], using so-called parfactor graphs, based on factor graphs, as a lifted model. Markov Logic networks [10] are another form of lifted models based on Markov networks using weighted first-order logic formulas. Formalisms that use similar ideas include relational BNs [11] as well as ProbLog [12].

Even though the field of lifted inference has only really started in 2003, the field is vast. We highlight a few ideas and areas here. For a complete survey, please consider a dedicated survey such as [3]. As far as history goes, Poole proposes a first version of lifted variable elimination (LVE) as an exact inference algorithm. In the following years, this first lifted inference algorithm has been extended by various researchers in terms of counting, constraint languages, or query language [13–19] for single-query inference. Holtzen et al. identify further symmetries on a propositional level in a factor graph for lifted inference [20]. For multi-query answering, researchers have introduced lifted versions of helper structures, which introduces overhead beforehand but allows for efficient, fast query answering online. Such algorithms include (i) the lifted junction tree algorithm (LJT) [21–23] lifting the junction tree algorithm [24–26] and using LVE as a subroutine, (ii) first-order knowledge compilation (FOKC) [27,28] lifting knowledge compilation [29], (iii) lifted first-order rules [30], and (iv) probabilistic theorem proving [31] as exact approaches to inference.

Approximate algorithms include deterministic algorithms such as lifted belief propagation [32–35] as well as sampling-based stochastic algorithms such as (i) lifted importance sampling [36], (ii) lifted Gibbs sampling [37,36,38], and (iii) lifted Markov Chain Monte Carlo sampling [39,40]. Bisimulation-inspired approximate inference allows for trading off the degree of compression of a graph with accuracy [41]. Another branch of approximate inference regards variational inference, which has also seen lifting extensions, e.g., for hybrid relational factor graphs [42], which incorporates discrete and continuous random variables, or using automorphism groups [43].

The above algorithms handle queries for conditional and marginal probability distributions or events. Another type of queries regards assignments, for all random variables without evidence, called most probable explanation (MPE), or for a subset of those, called maximum-a-posteriori (MAP) assignment (sometimes called MAP and marginal MAP, respectively). For parfactors, LVE has been extended to handle maxing out operations instead of summing out operations for elimination [14,44,45], which also enables assignment queries for LJT [46], while for MLNs, integer polynomial programming as well as linear programming has been used for efficient inference [47–49] and a new lifting rule enables eliminating logvars [50]. Another approach builds a lifted graph for an MLN with evidence that allows for a coarse-to-fine approximation.

Next to these episodic models, i.e., models without a temporal or sequential component, lifting has also been applied to dynamic factor graphs [51] and dynamic Markov networks [34]. The former comes with a dedicated exact inference algorithm for filtering, prediction, and hindsight queries [52], the latter uses lifted belief propagation [34]. A relational approach to dynamic Bayesian networks [53] enables to use some repeated structure [54]. A temporal version enables computing most probable state sequences in a lifted way [55].

Over the years, researchers have also worked on runtime complexity and completeness results. A major result shows that lifting allows for tractability with respect to domain sizes [56]. For certain model classes, a so-called domain-lifted algorithm runs in time polynomial regarding domain sizes. Domain-lifted algorithms include LVE [57], LJT [46], and FOKC [58] for models containing two logvars per parfactor or formulas or at most one logvar per PRV in the discrete setting. There exists a domain-lifted sampling scheme for the two-logvar class [59]. A new lifting rule has also opened up previously not liftable cases in the class of formulas containing three logvars [60]. For temporal models, the model class of two logvars is no longer liftable as a whole if keeping up temporal separation [61].

For decision making, a decision-theoretic parfactor model [62] as well as a Markov logic decision network [62] exist, which include parameterized actions and utilities. The former is accompanied by a lifted algorithm to solve a maximum expected utility problem. A temporal extension enables lifted online decision making over time [63]. Offline decision making with repeated structure is formalized using first-order Markov decision processes [64] based on the situation calculus [65]. Lifting has been incorporated as a means to prune the search space within symbolic dynamic programming [66]. In a more recent development, lifting has also been applied to the agent set in multi-agent decision making [67].

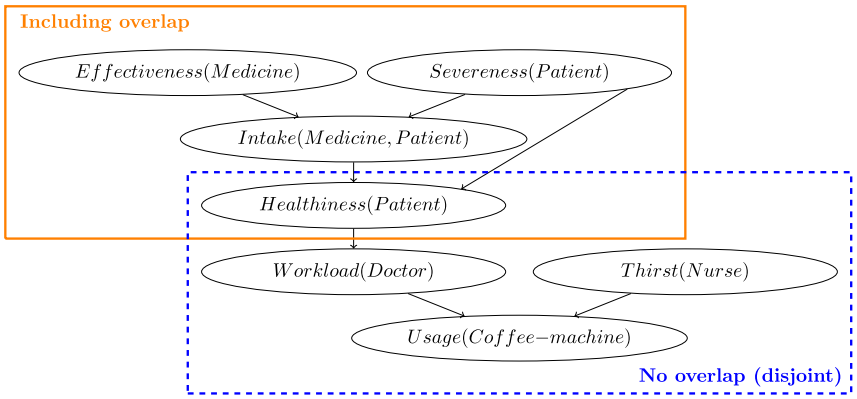


Fig. 2. Running example. (For interpretation in a black and white print, we differentiate not only by color but also by line structure (solid vs. dotted).

The above formalisms and algorithms work specifically for PRVs with discrete ranges. In terms of continuous models, Choi et al. have transferred lifting to the continuous setting where they have developed lifted inference for a pairwise potential factor graph [68]. Probabilistic soft logic enables models close to MLNs but also continuous modeling based on Hinge-loss Markov random fields [69]. The first ideas for this article have been published in [6] and [7], lifting GBNs as well as the inference on them based on a lifted full joint.

In the discrete setting, researchers have also taken a closer look at evidence that potentially breaks symmetries [70], uncertain evidence [71,72], and lifting evidence in lifted belief propagation [73]. Using a similar idea to lifting, clustering is used to handle evidence efficiently in Markov networks [74]. Symmetry-breaking evidence is especially a problem in temporal models as slightly diverging evidence over time for groups of otherwise indistinguishable objects is a common occurrence, grounding the model over time. To keep inference polynomial over time, clustering is used to merge groups again, which can be shown to exhibit an indefinitely bounded error [75]. All these approaches nominally deal with grouping of evidence but do so in an approximate manner while this article considers an exact approach in the continuous domain. To the best of our knowledge, this article provides a unified look on lifting and GBNs for the first time, combining the different approaches to transforming GBNs into a joint distribution under the lifting banner. Before diving into lifted construction and query answering after parameterized GBNs, we introduce a running example.

4. Introducing a running example

We setup a running example to illustrate the operations performed in this article, which can be seen in Fig. 2. The example is designed to showcase certain details or cruxes and not to be considered fully realistic (containing not too serious elements like modeling the coffee consumption of doctors and nurses).

The example contains the randvars *Effectiveness (E)*, *Intake (I)*, *Severity (S)*, *Health (H)*, *Workload (W)*, *Thirst (T)*, and *Usage (U)* and the logvars *Medicine (M)*, *Patient (P)*, *Doctor (D)*, *Nurse (N)*, and *Coffee-machine (C)*. Randvars and logvars are combined into PRVs and put into dependence as follows: The *Effectiveness* of a *Medicine*, $E(M)$, and the *Severity* of a *Patient*'s disease, $S(P)$, have an influence on the *Intake* of a specific *Medicine* for a *Patient*, $I(M,P)$. The *Severity* of a *Patient*'s disease and the *Intake* of a specific *Medicine* for a *Patient* have then an influence on the *Health* of a *Patient*, $H(P)$, after the treatment, which then influence the *Workload* of the *Doctors*, $W(D)$. The *Workload* of the *Doctors* and the *Thirst* of the *Nurses*, $T(N)$, have an influence on the *Usage* of the *Coffee-machine*, $U(C)$. In the following sections, we use individual parts of this network to demonstrate our methods. The starting point is the blue, dashed box, where PRVs do not contain a overlap between their logvar sequences, followed by a generalization for the parts that contain overlaps (orange, solid box). We use a topological ordering of $(E(M), S(P), I(M,P), H(P), T(N), W(D), U(C))$ for our example, together with a global logvar sequence $\mathcal{L} = (M, P, N, D, C)$ defining the global logvar ordering. In the formal descriptions, we always have an iterator (often denoted as $s = 1, \ldots, M$) over the PRVs. In example calculations, we use the PRV names as often as possible, but when iterating over example PRVs, e.g., within a sum, we use $Y_1, \ldots, Y_7$ as synonyms for $E(M)$, $S(P)$, $I(M,P)$, $H(P)$, $T(N)$, $W(D)$, $U(C)$, respectively, to ensure better readability. We also define two helper functions:

Definition 9 *(PRV position, preceding PRVs).* Given a set of PRVs $\boldsymbol{Y}$ and a ordering O, we define $pos(Y_s)$ of a PRV $Y_s \in \boldsymbol{Y}$ to return the position in the ordering

$$pos(Y_s) = s \tag{16}$$

and $pre(Y_s)$ to return all PRVs preceding Y_s

$$pre(Y_s) = \{Y_r | r < s\}. \tag{17}$$

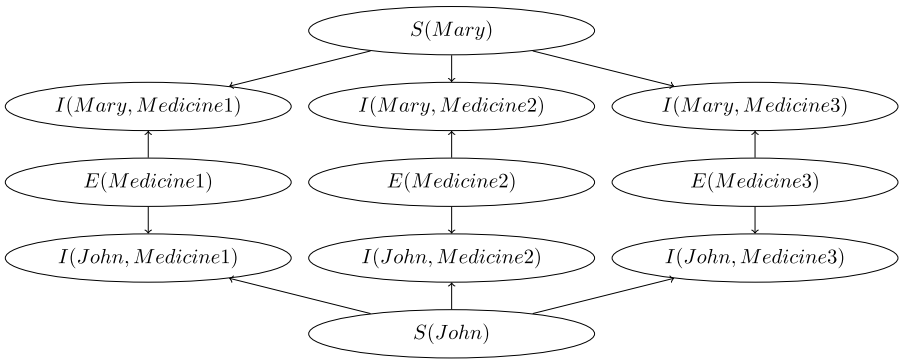


Fig. 3. Grounding parts of the running example.

The parameters $\lambda, \boldsymbol{\eta}$, and $\boldsymbol{\zeta}$ of the parameterized GBN are as follows:

$$\lambda = \begin{pmatrix} \lambda_{E(M)} \\ \lambda_{S(P)} \\ \lambda_{I(M,P)} \\ \lambda_{H(P)} \\ \lambda_{T(N)} \\ \lambda_{W(D)} \\ \lambda_{U(C)} \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}, \boldsymbol{\eta} = \begin{pmatrix} \eta_{E(M)} \\ \eta_{S(P)} \\ \eta_{I(M,P)} \\ \eta_{H(P)} \\ \eta_{T(N)} \\ \eta_{W(D)} \\ \eta_{U(C)} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 3 \\ 0 \\ 2 \\ 3 \\ 4 \end{pmatrix}, \text{ and } \begin{aligned} \zeta_{E(M),I(M,P)} &= 2 \\ \zeta_{S(P),I(M,P)} &= 5 \\ \zeta_{S(P),H(P)} &= -3 \\ \zeta_{I(M,P),H(P)} &= 4. \\ \zeta_{H(P),W(D)} &= -2 \\ \zeta_{T(N),U(C)} &= 2 \\ \zeta_{W(D),U(C)} &= 3 \end{aligned} \tag{18}$$

The domain size of the PRVs in the network is defined by the cardinality of the logvar domains, stored in what we later refer to as a cardinality vector $\boldsymbol{\tau}$. In example calculations, we use the following domain sizes:

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_M \\ \tau_P \\ \tau_N \\ \tau_D \\ \tau_C \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 3 \\ 2 \\ 3 \end{pmatrix}. \tag{19}$$

Behind this parameterized GBN lies a grounded GBN. If the logvar M has three constants, $\boldsymbol{D}(M) = \{Medicine1, Medicine2, Medicine3\}$, and the logvar P two constants, $\boldsymbol{D}(P) = \{John, Mary\}$, the resulting ground network of the first three PRVs $E(M)$, $S(P)$, and $I(M, P)$ would consist of 11 randvars, each with their own μ, σ^2, and β values to store. Fig. 3 shows a visualization of the ground network. With a domain size of ten medicines and 100 patients the ground network of only the first three PRVs would already contain 1,110 randvars. This explosion of the number of randvars shows how important efficient approaches to handling those networks are.

5. Constructing the lifted joint distribution

The starting point for constructing a lifted joint given a parameterized GBN is the grounded implementation of Algorithm 1 originally developed by Shachter and Kenley [9]. This section describes a lifted version of the algorithm. If the number of indistinguishable instances is high, the covariance matrix of the multivariate normal distribution is filled with many duplicate values. We present a lifted representation for more efficient memory usage and faster calculations. We start by explaining the basic case where PRVs are not allowed to have overlapping logvar sets using the non-matrix notation, followed by a matrix version of the same algorithms. Both approaches have their own intuition, adding their own contribution to the discussion of (parameterized) GBNs. The non-matrix notation gives better intuition of how conditional independencies in the network result in less summations, when summing up all influences caused by parent randvars. It also shows very clearly how equal values occur in the case of indistinguishable randvars and how equal values can simplify calculations. The matrix notation however is better suited for generalizing the algorithm for the overlapping case, because we can exploit block structure within the matrices to simplify equations, enabling scenarios such as one part of the running example, where the severity of a patient's illness only influences the intake of medicine for that specific patient.

5.1. The base case without overlaps: non-matrix notation

The algorithm by Shachter and Kenley [9] starts with a topologically ordered list of randvars in the GBN. As already described in Eq. (8), the symmetric covariance matrix $\boldsymbol{\Sigma}$ of the joint distribution $P(\mathcal{X}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is created inductively by

$$\Sigma_{X_j,X_i} = \Sigma_{X_i,X_j} = \sum_{X_k \in \boldsymbol{Pa}(X_j)} \Sigma_{X_i,X_k} \beta_{X_k,X_j} + \delta_{X_i,X_j} \sigma^2_{X_i}, \tag{20}$$

here indexed using full randvar names, with $i < j$ meaning X_i comes before X_j in the topological ordering.

Instead of looking at propositional randvars X_i and X_j, we now look at two sets of randvars $\boldsymbol{gr}(Y_s)$ and randvars $\boldsymbol{gr}(Y_t)$, each behaving indistinguishably, grouped into PRVs Y_s and Y_t with $s, t = 1, ..., M$, respectively. In the ground case, we would calculate the covariance between all $|\boldsymbol{D}(\boldsymbol{L}_s)|$ randvars $\boldsymbol{gr}(Y_s)$ and all $|\boldsymbol{D}(\boldsymbol{L}_t)|$ randvars $\boldsymbol{gr}(Y_t)$. If there exists a parent child relationship between Y_u and Y_s, all indistinguishable randvars $\boldsymbol{gr}(Y_u)$ are parents of the randvars $\boldsymbol{gr}(Y_s)$ due to the non-overlap in the logvar sequences. Based on this, we can reformulate Eq. (20) into

$$\Sigma_{X_i,X_j} = \sum_{Y_u \in \boldsymbol{Pa}(lif(X_j))} \left(\sum_{X_k \in \boldsymbol{gr}(Y_u)} \Sigma_{X_i,X_k} \beta_{X_k,X_j} \right) + \delta_{X_i,X_j} \sigma^2_{X_i}. \tag{21}$$

The sum in parentheses is working on ground level. Based on Section 2.5, all β_{X_k,X_j} for $X_k \in \boldsymbol{gr}(Y_u)$ are equal to $\zeta_{lif(X_k),lif(X_j)}$ in the non-overlapping case. If Σ_{X_i,X_k} was equal for all X_i and X_k, we could calculate it once and multiply it with the number of randvars $|\boldsymbol{D}(\boldsymbol{L}_u)|$ in $\boldsymbol{gr}(Y_u)$ to replace the sum. However, if X_i is equal to X_k, the recursive Σ_{X_i,X_k} contains a different value due to the Kronecker delta term resulting in an additional summand. The delta term prevents us from multiplying $\Sigma_{X_i,X_k} \zeta_{lif(X_k),lif(X_j)}$ with the number of randvars $|\boldsymbol{D}(\boldsymbol{L}_u)|$ to replace the sum. To get rid of the sum in parentheses nevertheless, we differentiate between X_i being a parent of X_j or not:

Case 1: If X_i is no parent of X_j, the $\delta_{X_i,X_k} \sigma^2_{X_i}$ term in the referenced Σ_{X_i,X_k} is always zero, resulting in a fully equal covariance Σ_{X_i,X_k} for all randvars $X_k \in \boldsymbol{gr}(Y_u)$, which reduces the second summation to a product between any covariance $\Sigma_{i,one(Y_u)}$, the number of parent randvars $|\boldsymbol{D}(\boldsymbol{L}_u)|$, and the linear dependency $\zeta_{Y_u,lif(X_j)}$:

$$\Sigma_{X_i,X_j} = \sum_{Y_u \in \boldsymbol{Pa}(lif(X_j))} \Sigma_{X_i,one(Y_u)} |\boldsymbol{D}(\boldsymbol{L}_u)| \zeta_{Y_u,lif(X_j)} + \delta_{X_i,X_j} \sigma^2_{X_i}. \tag{22}$$

Case 2: If X_i is a parent of X_j, then exactly one randvar in $\boldsymbol{gr}(Y_u)$ will be equal to X_i, which would result in a different covariance $\Sigma_{i,k}$. The key is that if X_i is a parent of X_j, all other randvars $\boldsymbol{gr}(lif(X_i))$ of the corresponding PRV $lif(X_i)$ are also parents of randvars $\boldsymbol{gr}(lif(X_j))$. This means that, independent of the specific randvar in the covariance function, there will always be exactly one covariance Σ_{X_i,X_k} where $\delta_{X_i,X_k} \sigma^2_{X_i}$ is nonzero.

Based on the two cases, we can reformulate Eq. (21) into

$$\Sigma_{X_i,X_j} = \sum_{Y_u \in \boldsymbol{Pa}(lif(X_j)} \left(\Sigma_{X_i,one(Y_u)} |\boldsymbol{D}(\boldsymbol{L}_u)| + \delta_{Y_u,lif(X_j)} \lambda_{lif(X_k)} \right) \zeta_{Y_u,lif(X_j)} + \delta_{X_i,X_j} \lambda_{lif(X_i)}. \tag{23}$$

Since the sum over $Y_u \in \boldsymbol{Pa}(lif(X_j))$ is equal for all combinations between randvars $\boldsymbol{gr}(lif(X_i))$ and randvars $\boldsymbol{gr}(lif(X_j))$, we introduce a new symbol ρ to store this value that is equal for all randvars belonging to the same PRV. $\boldsymbol{\rho}$ can be seen as a PRV covariance matrix. Introducing this new matrix as a result of the sum in Eq. (23) results in

$$\rho_{Y_s,Y_t} = \sum_{Y_u \in \boldsymbol{Pa}(Y_t)} \left(\rho_{Y_s,Y_u} |\boldsymbol{D}(\boldsymbol{L}_u)| + \delta_{Y_u,Y_t} \lambda_{Y_u} \right) \zeta_{Y_u,Y_t}. \tag{24}$$

The last term $\delta_{X_i,X_j} \lambda_{lif(X_i)}$ of Eq. (23) only needs to be added if the ground covariance between two equal randvars X_i and X_i is calculated. This additional value is already stored in the λ vector. We can use the PRV covariance $\boldsymbol{\rho}$ matrix and the λ vector to calculate a ground covariance with

$$\Sigma_{X_i,X_j} = \rho_{lif(X_i),lif(X_j)} + \delta_{X_i,X_j} \lambda_{lif(X_i)}. \tag{25}$$

Eqs. (24) and (25) show that we can calculate the ground covariance matrix only using $\boldsymbol{\rho}$ and λ. All lifted covariances ρ_{Y_s,Y_t} can be stored in a $M \times M$-dimensional matrix $\boldsymbol{\rho}$ and the variances λ_{X_s} can be stored in a M-dimensional vector $\boldsymbol{\lambda}$, no longer requiring storage depending on domain sizes.

In addition to the covariance matrix, the multivariate Gaussian also needs a mean vector $\boldsymbol{\mu}$. The ground mean-vector contains the means of all randvars. Since the randvars $\boldsymbol{gr}(Y_s)$ of a PRV Y_s have the same mean, the M-dimensional PRV mean vector $\boldsymbol{\eta}$ is a lifted version of $\boldsymbol{\mu}$. We get the mean for a ground randvar X_i by expanding the lifted mean vector $\boldsymbol{\eta}$ as $\mu_{X_i} = \eta_{lif(X_i)}$.

In summary, to store all information of the lifted joint distribution over M PRVs $Y_1, ..., Y_M$, we need an M-dimensional lifted mean vector $\boldsymbol{\eta}$, an $M \times M$-dimensional PRV covariance matrix $\boldsymbol{\rho}$, and an M-dimensional node variance vector $\boldsymbol{\lambda}$. Additionally, we need to store the cardinalities of the randvars, i.e., the domain sizes of the corresponding logvar sequences, needed in Eq. (24). In the case with no overlap, each PRV has its own set of logvars so we could store the cardinalities in an M-dimensional vector but in preparation of the general case, we store individual domain sizes in a $|\mathcal{L}|$-dimensional cardinality vector $\boldsymbol{\tau}$. Let us look at our example to see Eq. (24) at work.

Example 5. As a basis, we use the blue, dotted part with no overlapping logvars of the parameterized GBN visualized in Fig. 2. The parameters for λ and $\boldsymbol{\zeta}$ are defined in Eqs. (18) and (19). The lifted calculations from Eq. (24) to construct the joint covariance matrix are as follows:

$$\rho_{H(P),H(P)} = 0, \text{ as } H(P) \text{ has no parent}$$
$$\rho_{H(P),T(N)} = 0, \text{ as } T(N) \text{ has no parent}$$
$$\rho_{T(N),T(N)} = 0, \text{ as } T(N) \text{ has no parent}$$
$$\rho_{H(P),W(D)} = (\rho_{H(P),H(P)}\tau_P + \lambda_{H(P)})\zeta_{H(P),W(D)} = (0 \cdot 2 + 1)(-2) = -2$$
$$\rho_{T(N),W(D)} = (\rho_{T(N),H(P)}\tau_P + 0)\zeta_{H(P),W(D)} = (0 \cdot 2 + 0)(-2) = 0$$
$$\rho_{W(D),W(D)} = (\rho_{W(D),H(P)}\tau_P + 0)\zeta_{H(P),W(D)} = (-2 \cdot 2 + 0)(-2) = 8$$
$$\begin{aligned}\rho_{H(P),U(C)} &= (\rho_{H(P),T(N)}\tau_N + 0)\zeta_{T(N),U(C)} + (\rho_{H(P),W(D)}\tau_D + 0)\zeta_{W(D),U(C)} \\ &= (0 \cdot 3 + 0)2 + (-2 \cdot 2 + 0)3 = -12\end{aligned}$$
$$\begin{aligned}\rho_{T(N),U(C)} &= (\rho_{T(N),T(N)}\tau_N + \lambda_{T(N)})\zeta_{T(N),U(C)} + (\rho_{T(N),W(D)}\tau_D + 0)\zeta_{W(D),U(C)} \\ &= (0 \cdot 3 + 2)2 + (0 \cdot 2 + 0)3 = 4\end{aligned}$$
$$\begin{aligned}\rho_{W(D),U(C)} &= (\rho_{W(D),T(N)}\tau_N + 0)\zeta_{T(N),U(C)} + (\rho_{W(D),W(D)}\tau_D + \lambda_{W(D)})\zeta_{W(D),U(C)} \\ &= (0 \cdot 3 + 0)2 + (8 \cdot 2 + 3)3 = 57\end{aligned}$$
$$\begin{aligned}\rho_{U(C),U(C)} &= (\rho_{U(C),T(N)}\tau_N + 0)\zeta_{T(N),U(C)} + (\rho_{U(C),W(D)}\tau_D + 0)\zeta_{W(D),U(C)} \\ &= (4 \cdot 3 + 0)2 + (57 \cdot 2 + 0)3 = 366\end{aligned}$$

The resulting ρ matrix is then

$$\rho = \begin{bmatrix} 0 & 0 & -2 & -12 \\ 0 & 0 & 0 & 4 \\ -2 & 0 & 8 & 57 \\ -12 & 4 & 57 & 366 \end{bmatrix}.$$

Getting the variance of $H(Doctor1)$ results in a grounding of

$$\Sigma_{W(Doctor1),W(Doctor1)} = \rho_{W(D),W(D)} + \lambda_{W(D)} = 8 + 3 = 11.$$

Appendix A.1 contains the same example calculated on ground level using Eq. (8), where one can see the correspondence between the two calculations. In the next section, we look at the same case of a parameterized GBN without overlap using the matrix notation.

5.2. The base case without overlaps: matrix notation

In this section, we use the matrix notation of Shachter and Kenley's approach [9]. Algorithm 1 shows pseudocode for the approach. Before diving into the algorithm, we introduce two basic rules for working with identity and all-ones matrices that are essential for the matrix notation version. The first rule relates to the multiplication of all-ones matrices and has been covered by Timm [76]. The second rule follows directly from the first one.

Lemma 1. *Given two all-ones matrices $\boldsymbol{A} = \mathbf{J}_{E\times F}$ and $\boldsymbol{B} = \mathbf{J}_{F\times G}$, the product of the two matrices is again an all-ones matrix multiplied by F*

$$\boldsymbol{AB} = \mathbf{J}_{E\times F}\mathbf{J}_{F\times G} = F\mathbf{J}_{E\times G}$$

Proof. With matrix algebra, each entry at position (i, j), with $i \in \{1, ..., E\}$ and $j \in \{1, ..., G\}$ of the resulting matrix is calculated by the sum

$$\sum_{k=1}^{F} a_{i,k}b_{k,i} = \sum_{k=1}^{F} 1 = F.$$

An $E \times G$ dimensional matrix whose elements are all equal to F can be written as $F\mathbf{J}_{E\times G}$. □

Lemma 2. *Given matrices*

$$\boldsymbol{A} = p\mathbf{J}_{E\times E} + q\mathbf{I}_{E\times E},$$
$$\boldsymbol{B} = r\mathbf{J}_{E\times E} + s\mathbf{I}_{E\times E}, \text{ and}$$
$$\boldsymbol{C} = t\mathbf{J}_{E\times G},$$

it holds that

$$\boldsymbol{AB} = u\mathbf{J}_{E\times E} + v\mathbf{I}_{E\times E} \text{ and}$$
$$\boldsymbol{AC} = w\mathbf{J}_{E\times G}$$

with

$$u = Epr + ps + qr,$$
$$v = qs, \text{ and}$$
$$w = Ept + qt.$$

Proof. Using the distributivity characteristics of matrix multiplication and Lemma 1, it holds that

$$\begin{aligned}\boldsymbol{AB} &= u\mathbf{J}_{E\times E} + v\mathbf{I}_{E\times E}\\ &= (p\mathbf{J}_{E\times E} + q\mathbf{I}_{E\times E})(r\mathbf{J}_{E\times E} + s\mathbf{I}_{E\times E})\\ &= Epr\mathbf{J}_{E\times E} + ps\mathbf{J}_{E\times E} + qr\mathbf{J}_{E\times E} + qs\mathbf{I}_{E\times E}\\ &= (Epr + ps + qr)\mathbf{J}_{E\times E} + qs\mathbf{I}_{E\times E}\end{aligned}$$

and

$$\begin{aligned}\boldsymbol{AC} &= u\mathbf{J}_{E\times E} + v\mathbf{I}_{E\times E}\\ &= (p\mathbf{J}_{E\times E} + q\mathbf{I}_{E\times E})(t\mathbf{J}_{E\times G})\\ &= Ept\mathbf{J}_{E\times E} + qt\mathbf{J}_{E\times E}\\ &= (Ept + qt)\mathbf{J}_{E\times E}. \quad \square\end{aligned}$$

For better readability, we restate Eqs. (9) to (12) as the starting point for the lifted construction using matrix notation:

$$\Sigma_{X_1,X_1} = \sigma^2_{X_1}, \tag{26}$$
$$\boldsymbol{\Sigma}_{\boldsymbol{X_i},X_j} = \boldsymbol{\Sigma}_{\boldsymbol{X_i},\boldsymbol{X_i}}\boldsymbol{T}_{\boldsymbol{X_i},X_j}, \tag{27}$$
$$\boldsymbol{\Sigma}_{X_j,\boldsymbol{X_i}} = \boldsymbol{\Sigma}^T_{\boldsymbol{X_i},X_j}, \tag{28}$$
$$\Sigma_{X_j,X_j} = \boldsymbol{\Sigma}_{X_j,\boldsymbol{X_i}}\boldsymbol{T}_{\boldsymbol{X_i},X_j} + \sigma^2_{X_j}. \tag{29}$$

For lifting the Shachter and Kenley algorithm, we structure the covariance matrix $\boldsymbol{\Sigma}$ and the transition matrix $\boldsymbol{T}$ into $M \times M$ blocks. For $s,t \in \{1,\dots,M\}$, we identify the blocks with $\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_s),\boldsymbol{gr}(Y_t)}$ and $\boldsymbol{T}_{\boldsymbol{gr}(Y_s),\boldsymbol{gr}(Y_t)}$. In the non-overlapping scenario, the blocks of the $\boldsymbol{T}$ matrix are either filled with zero or filled with the corresponding ζ_{Y_s,Y_t} value for the relationship between the two PRVs Y_s and Y_t:

$$\boldsymbol{T}_{\boldsymbol{gr}(Y_s),\boldsymbol{gr}(Y_t)} = \zeta_{Y_s,Y_t} \cdot \mathbf{J}_{|\boldsymbol{D}(\boldsymbol{L}_s)|\times|\boldsymbol{D}(\boldsymbol{L}_t)|}, \tag{30}$$

where $\mathbf{J}_{|\boldsymbol{D}(\boldsymbol{L}_s)|\times|\boldsymbol{D}(\boldsymbol{L}_t)|}$ is the $|\boldsymbol{D}(\boldsymbol{L}_s)| \times |\boldsymbol{D}(\boldsymbol{L}_t)|$-dimensional all-ones matrix.

The diagonal blocks of $\boldsymbol{T}$ are always zero, because self-reference is not allowed in (parameterized) GBNs. This structure allows us to go blockwise through Eqs. (26) to (29). For any $t \in \{1,\dots,M\}$ and $\boldsymbol{s} = \{1,\dots,t-1\}$, it holds that

$$\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_1),\boldsymbol{gr}(Y_1)} = \lambda_{Y_1}\mathbf{I}_{|\boldsymbol{D}(\boldsymbol{L}_1)|\times|\boldsymbol{D}(\boldsymbol{L}_1)|}, \tag{31}$$
$$\boldsymbol{\Sigma}_{\boldsymbol{gr}(\boldsymbol{Y_s}),\boldsymbol{gr}(Y_t)} = \boldsymbol{\Sigma}_{\boldsymbol{gr}(\boldsymbol{Y_s}),\boldsymbol{gr}(\boldsymbol{Y_s})}\boldsymbol{T}_{\boldsymbol{gr}(\boldsymbol{Y_s}),\boldsymbol{gr}(Y_t)}, \tag{32}$$
$$\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_t),\boldsymbol{gr}(\boldsymbol{Y_s})} = \boldsymbol{\Sigma}^T_{\boldsymbol{gr}(\boldsymbol{Y_s}),\boldsymbol{gr}(Y_t)}, \tag{33}$$
$$\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_t),\boldsymbol{gr}(Y_t)} = \boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_t),\boldsymbol{gr}(\boldsymbol{Y_s})}\boldsymbol{T}_{\boldsymbol{gr}(\boldsymbol{Y_s}),\boldsymbol{gr}(Y_t)} + \lambda_{Y_t}\mathbf{I}_{|\boldsymbol{D}(\boldsymbol{L}_t)|\times|\boldsymbol{D}(\boldsymbol{L}_t)|}. \tag{34}$$

The initial block is always the identity matrix $\mathbf{I}$ multiplied with the node variance of the first PRV. We use an induction-like approach to show that the covariance matrix can be again calculated and stored in a lifted way. We show how the first iteration of Eqs. (31) to (34) works out and then generalize further.

Equation (32) is a multiplication of the first block $\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_1),\boldsymbol{gr}(Y_1)}$ with the transition matrix block $\boldsymbol{T}_{\boldsymbol{gr}(Y_1),\boldsymbol{gr}(Y_2)}$

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_1),\boldsymbol{gr}(Y_2)} &= \boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_1),\boldsymbol{gr}(Y_1)}\boldsymbol{T}_{\boldsymbol{gr}(Y_1),\boldsymbol{gr}(Y_2)}\\ &= \lambda_{Y_1}\mathbf{I}_{|\boldsymbol{D}(\boldsymbol{L}_1)|\times|\boldsymbol{D}(\boldsymbol{L}_1)|}\zeta_{Y_1,Y_2}\mathbf{J}_{|\boldsymbol{D}(\boldsymbol{L}_1)|\times|\boldsymbol{D}(\boldsymbol{L}_2)|}\\ &= \lambda_{Y_1}\zeta_{Y_1,Y_2}\mathbf{J}_{|\boldsymbol{D}(\boldsymbol{L}_1)|\times|\boldsymbol{D}(\boldsymbol{L}_2)|},\end{aligned} \tag{35}$$

where the matrix multiplication with the identity matrix (as the neutral element) has no effect. The next block on the diagonal using Eq. (34) is calculated by

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_2),\boldsymbol{gr}(Y_2)} &= \boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_2),\boldsymbol{gr}(Y_1)}\boldsymbol{T}_{\boldsymbol{gr}(Y_1),\boldsymbol{gr}(Y_2)} \\ &= \lambda_{Y_1}\zeta_{Y_1,Y_2}\mathbf{J}_{|\boldsymbol{D}(\boldsymbol{L}_2)|\times|\boldsymbol{D}(\boldsymbol{L}_1)|}\zeta_{Y_1,Y_2}\mathbf{J}_{|\boldsymbol{D}(\boldsymbol{L}_1)|\times|\boldsymbol{D}(\boldsymbol{L}_2)|} + \lambda_{Y_2}\mathbf{I}_{|\boldsymbol{D}(\boldsymbol{L}_2)|\times|\boldsymbol{D}(\boldsymbol{L}_2)|} \\ &= \lambda_{Y_1}\zeta_{Y_1,Y_2}^2|\boldsymbol{D}(\boldsymbol{L}_1)|\mathbf{J}_{|\boldsymbol{D}(\boldsymbol{L}_2)|\times|\boldsymbol{D}(\boldsymbol{L}_2)|} + \lambda_{Y_2}\mathbf{I}_{|\boldsymbol{D}(\boldsymbol{L}_2)|\times|\boldsymbol{D}(\boldsymbol{L}_2)|},\end{aligned} \tag{36}$$

where we use Lemma 1 in the multiplication of the two all-ones matrices.

In further steps, when calculating Eq. (32) or Eq. (34), there are several PRVs in $\boldsymbol{Y}_s$. With block matrix multiplication rules, we can calculate the resulting blocks $\boldsymbol{\Sigma}_{\boldsymbol{gr}(\boldsymbol{Y}_s),\boldsymbol{gr}(\boldsymbol{Y}_t)}$ individually. We use $s \in 1,...,t-1$ as an index to iterate over the blocks:

$$\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_s),\boldsymbol{gr}(Y_t)} = \sum_{k=1}^{t-1}\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_s),\boldsymbol{gr}(Y_k)}\boldsymbol{T}_{\boldsymbol{gr}(Y_k),\boldsymbol{gr}(Y_t)}. \tag{37}$$

Based on Lemma 2, we know that the resulting block $\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_s),\boldsymbol{gr}(Y_t)}$ can be expressed by a multiple of the all-ones matrix, if all $\boldsymbol{\Sigma}_{\boldsymbol{gr}(\boldsymbol{Y}_s),\boldsymbol{gr}(\boldsymbol{Y}_s)}$ have a structure of $a\mathbf{J} + b\mathbf{I}$. For the off-diagonal blocks, we have shown that they follow this structure if their preceding blocks follow the structure as well. The formula for the on-diagonal blocks can also be converted to a summation:

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_t),\boldsymbol{gr}(Y_t)} &= \boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_t),\boldsymbol{gr}(\boldsymbol{Y}_s)}\boldsymbol{T}_{\boldsymbol{gr}(\boldsymbol{Y}_s),\boldsymbol{gr}(Y_t)} + \lambda_{Y_t}\mathbf{I}_{|\boldsymbol{D}(\boldsymbol{L}_t)|\times|\boldsymbol{D}(\boldsymbol{L}_t)|} \\ &= \sum_{k=1}^{t-1}\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_t),\boldsymbol{gr}(Y_k)}\boldsymbol{T}_{\boldsymbol{gr}(Y_k),\boldsymbol{gr}(Y_t)} + \lambda_{Y_t}\mathbf{I}_{|\boldsymbol{D}(\boldsymbol{L}_t)|\times|\boldsymbol{D}(\boldsymbol{L}_t)|}.\end{aligned} \tag{38}$$

Based on Lemma 2, the summation over k results again in a multiple of the all-ones matrix. On the diagonal, there is the identity matrix addition. In this induction-like approach, we only need to show that the starting point is also following the structure, which is the case based on Eq. (31). The beauty of this approach lies in the closed form of the matrix structure under the operations performed in the algorithm. The reader might have wondered why we have put such an emphasis on the relatively simple Lemma 2, but finding a lifted structure that is closed under the operations performed in the algorithm by Shachter and Kenley [9] will also be key for the following generalizations.

Closing the loop to the previous section, we can now store the factor of the all-ones matrix of each block in the $M \times M$ dimensional matrix $\boldsymbol{\rho}$ and the factor of the identity matrix, only present on the diagonal blocks, in the M dimensional vector $\boldsymbol{\lambda}$ to have a lifted representation of the joint covariance matrix. Additionally, remember we store the domain size of each logvar in the cardinality vector $\boldsymbol{\tau}$. Grounding a full block can be done by

$$\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_s),\boldsymbol{gr}(Y_t)} = \rho_{Y_s,Y_t}\mathbf{J}_{\tau_s\times\tau_t} + \delta_{Y_s,Y_t}\lambda_s\mathbf{I}_{\tau_s\times\tau_s}. \tag{39}$$

To get a specific covariance between two ground randvars X_i and X_j, we select the entries of the block matrix in Eq. (39) by

$$\Sigma_{X_i,X_j} = \rho_{lif(X_i),lif(X_j)} + \delta_{X_i,X_j}\lambda_{lif(X_i)}, \tag{40}$$

which is the same equation as Eq. (25) for the non-matrix notation case. Let us revisit Example 5, but this time illustrate how calculations can be understood in terms of block matrix operations.

Example 6. The ground matrix $\boldsymbol{T}$ has the following structure

$$\boldsymbol{T} = \begin{bmatrix} 0&0&0&0&0&-2&-2&0&0&0\\ 0&0&0&0&0&-2&-2&0&0&0\\ 0&0&0&0&0&0&0&2&2&2\\ 0&0&0&0&0&0&0&2&2&2\\ 0&0&0&0&0&0&0&2&2&2\\ 0&0&0&0&0&0&0&3&3&3\\ 0&0&0&0&0&0&0&3&3&3\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{\tau_P\times\tau_P} & \mathbf{0}_{\tau_P\times\tau_N} & -2\mathbf{J}_{\tau_P\times\tau_D} & \mathbf{0}_{\tau_P\times\tau_C}\\ \mathbf{0}_{\tau_N\times\tau_P} & \mathbf{0}_{\tau_N\times\tau_N} & \mathbf{0}_{\tau_N\times\tau_D} & 2\mathbf{J}_{\tau_N\times\tau_C}\\ \mathbf{0}_{\tau_D\times\tau_P} & \mathbf{0}_{\tau_D\times\tau_N} & \mathbf{0}_{\tau_D\times\tau_D} & 3\mathbf{J}_{\tau_D\times\tau_C}\\ \mathbf{0}_{\tau_C\times\tau_P} & \mathbf{0}_{\tau_C\times\tau_N} & \mathbf{0}_{\tau_C\times\tau_D} & \mathbf{0}_{\tau_C\times\tau_C} \end{bmatrix}$$

We only show a few block matrix notations and do not repeat the full example here. The full example can be found in Appendix A.2.

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{gr}(H(P)),\boldsymbol{gr}(H(P))} &= \lambda_{H(P)}\mathbf{I}_{\tau_P\times\tau_P} = 1\mathbf{I}_{\tau_P\times\tau_P} \\ \boldsymbol{\Sigma}_{\boldsymbol{gr}(H(P)),\boldsymbol{gr}(T(N))} &= \boldsymbol{\Sigma}_{\boldsymbol{gr}(H(P)),\boldsymbol{gr}(H(P))}\boldsymbol{T}_{\boldsymbol{gr}(H(P)),\boldsymbol{gr}(T(N))} = \mathbf{I}_{\tau_P\times\tau_P}\mathbf{0}_{\tau_P\times\tau_N} = \mathbf{0}_{\tau_P\times\tau_N} \\ \boldsymbol{\Sigma}_{\boldsymbol{gr}(T(N)),\boldsymbol{gr}(T(N))} &= \boldsymbol{\Sigma}_{\boldsymbol{gr}(T(N)),\boldsymbol{gr}(H(P))}\boldsymbol{T}_{\boldsymbol{gr}(H(P)),\boldsymbol{gr}(T(N))} + \lambda_{T(N)}\mathbf{I}_{\tau_N\times\tau_N} \\ &= \mathbf{0}_{\tau_N\times\tau_P}\mathbf{0}_{\tau_N\times\tau_P} + \lambda_{T(N)}\mathbf{I}_{\tau_N\times\tau_N} = 0\mathbf{J}_{\tau_N\times\tau_N} + 2\mathbf{I}_{\tau_N\times\tau_N}\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\Sigma}_{gr(H(P)),gr(W(D))} &= \boldsymbol{\Sigma}_{gr(H(P)),gr(H(P))}\boldsymbol{T}_{gr(H(P)),gr(W(D))} \\
&\quad + \boldsymbol{\Sigma}_{gr(H(P)),gr(T(N))}\boldsymbol{T}_{gr(T(N)),gr(W(D))} \\
&= 1\mathbf{I}_{\tau_P\times\tau_P}(-2)\mathbf{J}_{\tau_P\times\tau_D} + \mathbf{0}_{\tau_P\times\tau_N}\mathbf{0}_{\tau_N\times\tau_D} = (-2)\mathbf{J}_{\tau_P\times\tau_D} \\
\boldsymbol{\Sigma}_{gr(T(N)),gr(W(D))} &= \boldsymbol{\Sigma}_{gr(T(N)),gr(H(P))}\boldsymbol{T}_{gr(H(P)),gr(W(D))} \\
&\quad + \boldsymbol{\Sigma}_{gr(T(N)),gr(T(N))}\boldsymbol{T}_{gr(T(N)),gr(W(D))} \\
&= \mathbf{0}_{\tau_N\times\tau_P}(-2)\mathbf{J}_{\tau_P\times\tau_D} + 2\mathbf{I}_{\tau_N\times\tau_N}\mathbf{0}_{\tau_N\times\tau_D} = 0\mathbf{J}_{\tau_N\times\tau_D} \\
\boldsymbol{\Sigma}_{gr(W(D)),gr(W(D))} &= \boldsymbol{\Sigma}_{gr(W(D)),gr(H(P))}\boldsymbol{T}_{gr(H(P)),gr(W(D))} \\
&\quad + \boldsymbol{\Sigma}_{gr(W(D)),gr(T(N))}\boldsymbol{T}_{gr(T(N)),gr(W(D))} \\
&\quad + \lambda_{W(D)}\mathbf{I}_{\tau_D\times\tau_D} \\
&= (-2)\mathbf{J}_{\tau_D\times\tau_P}(-2)\mathbf{J}_{\tau_P\times\tau_D} + \mathbf{0}_{\tau_D\times\tau_N}\mathbf{0}_{\tau_N\times\tau_D} + \lambda_{W(D)}\mathbf{I}_{\tau_D\times\tau_D} \\
&= (-2)\cdot(-2)\cdot\tau_P\mathbf{J}_{\tau_D\times\tau_D} + \lambda_{W(D)}\mathbf{I}_{\tau_D\times\tau_D} = 8\mathbf{J}_{\tau_D\times\tau_D} + 3\mathbf{I}_{\tau_D\times\tau_D}
\end{aligned}$$

The example shows how using Lemma 1 and Lemma 2 results in the same ρ values as in Example 5.

We have introduced two approaches for constructing the lifted representation of the covariance matrix of ρ and λ, allowing us to work with the blue, dotted part of our running example in Fig. 2. Next, we will focus on generalizing the matrix approach to allow overlaps between logvar sequences.

5.3. Allowing for overlaps

Up until this point, the logvar sequences of connected PRVs are not allowed to have overlaps. However, as the orange, solid box in Fig. 2 shows, overlaps are important to introduce further conditional independencies into the model and thus make the model more expressive. When allowing overlaps between the logvars of a parent and a child PRV, the connections on ground level only exist if the groundings share the same instances of the overlapping logvars. Referring to our previously discussed case, these additional independencies violate the equality assumption used in Eq. (24) for the non-matrix notation and the structure of the $\boldsymbol{T}$ matrix in Eq. (30) for the matrix notation. From here on, we will focus on the matrix notation. Therefore, we start with specifying the general form of the transition matrix $\boldsymbol{T}$ given overlaps. Afterwards, we construct a lifted representation of the covariance matrix anew.

5.3.1. General form of the transition matrix

For the new lifted representation, we need the Kronecker product of matrices.

Definition 10 *(Kronecker product).* Given a $m \times n$ matrix $\boldsymbol{A}$ and a $p \times q$ matrix $\boldsymbol{B}$, the Kronecker product is defined as

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} A_{1,1}\boldsymbol{B} & \dots & A_{1,n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ A_{m,1}\boldsymbol{B} & \dots & A_{m,n}\boldsymbol{B} \end{bmatrix}. \tag{41}$$

We use the same $M \times M$ dimensional block structure for the transition matrix as introduced in the previous section. Given a parent PRV Y_s and a child PRV Y_t, a full logvar overlap, i.e., $\boldsymbol{L}_s = \boldsymbol{L}_t$, would mean that each randvar in $gr(Y_s)$ has exactly one connection to only one randvar in $gr(Y_t)$, namely, where Y_s and Y_t are grounded with the same constants. Given a global ordering, a full overlap results in a block design where the ζ value is only present on the diagonal:

$$\boldsymbol{T}_{gr(Y_s),gr(Y_t)} = \zeta_{Y_s,Y_t}\mathbf{I}_{|\boldsymbol{D}(\boldsymbol{L}_s)|\times|\boldsymbol{D}(\boldsymbol{L}_t)|}. \tag{42}$$

If the full overlap logvar sequence $\boldsymbol{L}_s = \boldsymbol{L}_t$ contains more than one logvar, we can reformulate Eq. (42) into

$$\boldsymbol{T}_{gr(Y_s),gr(Y_t)} = \zeta_{Y_s,Y_t}\mathbf{I}_{|\boldsymbol{D}(\boldsymbol{L}_s)|\times|\boldsymbol{D}(\boldsymbol{L}_t)|} = \zeta_{Y_s,Y_t}\bigotimes_{L\in\boldsymbol{L}_s}\mathbf{I}_{|\boldsymbol{D}(L)|\times|\boldsymbol{D}(L)|}. \tag{43}$$

Adding any non-overlapping logvar L_p in the parent sequence implies that previously one randvar is now replaced by $|\boldsymbol{D}(L_p)|$ randvars, which leads to a Kronecker multiplication of a $\mathbf{J}^{|\boldsymbol{D}(L_p)|\times 1}$ column vector (parent variables correspond to rows). Analogously, a non-overlapping additional logvar L_c in the child PRV results in a Kronecker multiplication of a $\mathbf{J}^{1\times|\boldsymbol{D}(L_c)|}$ row vector. Depending on the global ordering of the logvars, we get the following general formula for $\boldsymbol{T}$, with $\mathbf{J}^0 = \mathbf{I}$:

$$\boldsymbol{T}_{gr(Y_s),gr(Y_t)} = \zeta_{Y_s,Y_t}\bigotimes_{L\in seq(\boldsymbol{L}_s,\boldsymbol{L}_t)}\mathbf{J}^{ov(L,\boldsymbol{L}_s,\boldsymbol{L}_t)}_{dim(L,\boldsymbol{L}_s)\times dim(L,\boldsymbol{L}_t)}. \tag{44}$$

The *ov*-function in Eq. (44) controls if a identity matrix (overlap) or an all-ones matrix (non-overlap) is needed. For the non-overlapping logvars, the Kronecker product contains either a column or row vector dependent on the logvar being in the child or parent logvar sequence.

Example 7. For the overlapping part (orange, solid box) of our example in Fig. 2, the transition matrix $\boldsymbol{T}$ has the following block structure

$$\boldsymbol{T} = \begin{bmatrix} \mathbf{0}_{|\boldsymbol{D}(M)|\times|\boldsymbol{D}(M)|} & \mathbf{0}_{|\boldsymbol{D}(M)|\times|\boldsymbol{D}(P)|} & \boldsymbol{T}_{\boldsymbol{gr}(E(M)),\boldsymbol{gr}(I(M,P)} & \mathbf{0}_{|\boldsymbol{D}(M)|\times|\boldsymbol{D}(P)|} \\ \mathbf{0}_{|\boldsymbol{D}(P)|\times|\boldsymbol{D}(M)|} & \mathbf{0}_{|\boldsymbol{D}(P)|\times|\boldsymbol{D}(P)|} & \boldsymbol{T}_{\boldsymbol{gr}(S(P)),\boldsymbol{gr}(I(M,P)} & \boldsymbol{T}_{\boldsymbol{gr}(S(P)),\boldsymbol{gr}(H(P))} \\ \mathbf{0}_{|\boldsymbol{D}(M,P)|\times|\boldsymbol{D}(M)|} & \mathbf{0}_{|\boldsymbol{D}(M,P)|\times|\boldsymbol{D}(P)|} & \mathbf{0}_{|\boldsymbol{D}(M,P)|\times|\boldsymbol{D}(M,P)|} & \boldsymbol{T}_{\boldsymbol{gr}(I(M,P)),\boldsymbol{gr}(H(P))} \\ \mathbf{0}_{|\boldsymbol{D}(P)|\times|\boldsymbol{D}(M)|} & \mathbf{0}_{|\boldsymbol{D}(P)|\times|\boldsymbol{D}(P)|} & \mathbf{0}_{|\boldsymbol{D}(P)|\times|\boldsymbol{D}(M,P)|} & \mathbf{0}_{|\boldsymbol{D}(P)|\times|\boldsymbol{D}(P)|} \end{bmatrix},$$

with

$$\begin{aligned} \boldsymbol{T}_{\boldsymbol{gr}(E(M)),\boldsymbol{gr}(I(M,P)} &= \zeta_{E(M),I(M,P)}\mathbf{I}_{|\boldsymbol{D}(M)|\times|\boldsymbol{D}(M)|} \otimes \mathbf{J}_{1\times|\boldsymbol{D}(P)|}, \\ \boldsymbol{T}_{\boldsymbol{gr}(S(P)),\boldsymbol{gr}(I(M,P)} &= \zeta_{S(P),I(M,P)}\mathbf{J}_{1\times|\boldsymbol{D}(M)|} \otimes \mathbf{I}_{|\boldsymbol{D}(P)|\times|\boldsymbol{D}(P)|}, \\ \boldsymbol{T}_{\boldsymbol{gr}(S(P)),\boldsymbol{gr}(H(P))} &= \zeta_{S(P),H(P)}\mathbf{I}_{|\boldsymbol{D}(P)|\times|\boldsymbol{D}(P)|} \text{ and} \\ \boldsymbol{T}_{\boldsymbol{gr}(I(M,P)),\boldsymbol{gr}(H(P))} &= \zeta_{I(M,P),H(P)}\mathbf{J}_{|\boldsymbol{D}(P)|\times 1} \otimes \mathbf{I}_{|\boldsymbol{D}(P)|\times|\boldsymbol{D}(P)|}. \end{aligned}$$

Since Eq. (44) gives us a fixed rule to create a ground version of the transition matrix $\boldsymbol{T}$, we can store all the information needed in the cardinality vector $\boldsymbol{\tau}$ and the lifted transition matrix $\boldsymbol{\zeta}$.

5.3.2. Constructing the covariance matrix

The new general form for the transition matrix $\boldsymbol{T}$ based on a lifted representation $\boldsymbol{\zeta}$ allows for overlaps between logvar sequences. We investigate how this new structure of $\boldsymbol{T}$ influences Eqs. (31) to (34). Analogously to the non-overlapping case, we use an induction-like approach to show that all blocks follow a fixed structure, which enables a lifted storage of the covariance matrix $\boldsymbol{\Sigma}$. For ease of argumentation, we need the concept of a Kronecker sequence and Kronecker factors.

Definition 11 *(Kronecker sequence, Kronecker factor).* Given a logvar sequence $\boldsymbol{L}$, we define a *Kronecker sequence* as any sequence of Kronecker products over $\boldsymbol{L}$:

$$\bigotimes_{L\in\boldsymbol{L}} \boldsymbol{H}_{|\boldsymbol{D}(L)|\times|\boldsymbol{D}(L)|} \tag{45}$$

where the matrix $\boldsymbol{H}$ is either an identity matrix $\mathbf{I}$ or an all-ones matrix $\mathbf{J}$. We call every factor $\boldsymbol{H}_{|\boldsymbol{D}(L)|\times|\boldsymbol{D}(L)|}$ in the Kronecker sequence a *Kronecker factor*.

For a logvar sequence $\boldsymbol{L}_s$, there are $2^{|\boldsymbol{L}_s|}$ different Kronecker sequences following Definition 11, because for each logvar $L \in \boldsymbol{L}_s$, there are two possible Kronecker factors ($\mathbf{I}_{|\boldsymbol{D}(L)|\times|\boldsymbol{D}(L)|}$ and $\mathbf{J}_{|\boldsymbol{D}(L)|\times|\boldsymbol{D}(L)|}$). To identify one of the possible Kronecker sequences, we use a $|\boldsymbol{L}_s|$-dimensional index vector of zeroes and ones, where a zero stands for an identity matrix and one for an all-ones matrix, and $\mathbf{J}$ as basis with $\mathbf{J}^0 = \mathbf{I}$ and $\mathbf{J}^1 = \mathbf{J}$. For example, the vector $\mathbf{0} = (0 \dots 0)$ references the Kronecker sequence of only identity matrices, whereas $\mathbf{1} = (1 \dots 1)$ references the Kronecker sequence of only all-ones matrices. Each position in the vector corresponds to one logvar $L \in \boldsymbol{L}_s$ and thus to one Kronecker factor. The set of possible vectors is referenced by $\boldsymbol{\Phi}_s$. To iterate over all possible Kronecker sequences, we iterate over all possible index vectors, i.e., combinations of zeroes and ones, denoted as $\boldsymbol{q}_s \in \boldsymbol{\Phi}_s$. To iterate over all possible Kronecker sequences of an overlapping logvar sequence $\boldsymbol{L}_s \cap \boldsymbol{L}_t$ in a logvar sequence $\boldsymbol{L}_s \cup \boldsymbol{L}_t$, we write $\boldsymbol{q}_{s,t} \in \boldsymbol{\Phi}_{s,t}$ for short, with $\boldsymbol{q}_{s,t}$ being padded with ones at those positions that do not correspond to overlapping logvars. That means that the set $\boldsymbol{\Phi}_{s,t}$ for two non-overlapping logvar sequences only contains a single vector of ones.

Lemma 3. *Given a transition matrix $\boldsymbol{T}$ with a structure from Eq. (44) and the inductive application of Eqs. (31) to (34), the diagonal blocks of the covariance matrix $\boldsymbol{\Sigma}$ follow the structure*

$$\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_s,Y_s)} = \sum_{\boldsymbol{q}_s\in\boldsymbol{\Phi}_s} \rho_{s,s}^{\boldsymbol{q}_s} \bigotimes_{L\in\boldsymbol{L}_s} \mathbf{J}^{\pi_L(\boldsymbol{q}_s)}_{|\boldsymbol{D}(L)|\times|\boldsymbol{D}(L)|}, \tag{46}$$

and off-diagonal blocks of the covariance matrix $\boldsymbol{\Sigma}$ follow

$$\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_s,Y_t)} = \sum_{\boldsymbol{q}_{s,t}\in\boldsymbol{\Phi}_{s,t}} \rho_{s,t}^{\boldsymbol{q}_{s,t}} \bigotimes_{L\in\boldsymbol{L}_s\cup\boldsymbol{L}_t} \mathbf{J}^{qexp(\boldsymbol{q}_{s,t},L)}_{dim(L,\boldsymbol{L}_s)\times dim(L,\boldsymbol{L}_t)}, \tag{47}$$

with

$$qexp(\boldsymbol{q}_{s,t}, L) = \begin{cases} \pi_L(\boldsymbol{q}_{s,t}) & \text{if } L \in (\boldsymbol{L}_s \cup \boldsymbol{L}_t), \\ 1 & \text{otherwise.} \end{cases} \tag{48}$$

The tensor $\boldsymbol{\rho}$ contains $M \cdot M$ vectors, with each vector $\boldsymbol{\rho}_{s,t}$ having a length of $2^{|\boldsymbol{D}(\boldsymbol{L}_s \cap \boldsymbol{L}_t)|}$. Each of the values can be indexed with one vector $\boldsymbol{q}_{s,t}$ referring to the corresponding Kronecker sequence.

Proof. We use an induction-like proof. First, we assume that Lemma 3 is true for any current block structure of the covariance matrix. Then we show that all following blocks of the covariance matrix using the condition from Lemma 3 stay in the format. Last, we show that Eq. (31) is a valid starting point that fulfills the structure as well.

Equation (46) is a special case of Eq. (47) as both logvar sequences involved are equal, i.e., $\boldsymbol{q}_{s,t} = \boldsymbol{q}_s = \boldsymbol{q}_t$ ($\boldsymbol{L}_s = \boldsymbol{L}_t$), and dim and $qexp$ simplifying to their first cases.

Inserting the general equation for a block from the transition matrix from Eq. (44) and the equation for a block from the covariance matrix from Eq. (47) into the equation for calculating a new block in the covariance matrix from Eq. (37) for one arbitrary summand k results in

$$\begin{aligned}&\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_s),\boldsymbol{gr}(Y_k)} \boldsymbol{T}_{\boldsymbol{gr}(Y_k),\boldsymbol{gr}(Y_t)} = \\ &\sum_{\boldsymbol{q}_{s,k} \in \boldsymbol{\Phi}_{s,k}} \left(\rho_{s,k}^{\boldsymbol{q}_{s,k}} \bigotimes_{L \in \boldsymbol{L}_s \cup \boldsymbol{L}_k} \mathbf{J}_{dim(L,\boldsymbol{L}_s) \times dim(L,\boldsymbol{L}_k)}^{qexp(\boldsymbol{q}_{s,k},L)} \right) \left(\zeta_{Y_k,Y_t} \bigotimes_{L \in \boldsymbol{L}_k \cup \boldsymbol{L}_t} \mathbf{J}_{dim(L,\boldsymbol{L}_k) \times dim(L,\boldsymbol{L}_t)}^{ov(L,\boldsymbol{L}_k,\boldsymbol{L}_t)} \right)\end{aligned} \tag{49}$$

Multiplying sequences of Kronecker products can be rewritten using the mixed-product property in its general form [77]:

$$(A_1 \otimes A_2 \otimes \ldots \otimes A_n)(B_1 \otimes B_2 \otimes \ldots \otimes B_n) = (A_1 B_1 \otimes A_2 B_2 \otimes \ldots \otimes A_n B_n). \tag{50}$$

We use Eq. (50) to align the Kronecker factors that correspond to the same logvar. For each logvar L (and thus Kronecker factor), there are seven possible Kronecker factor combinations resulting from the occurrence of the logvar in the sequences $\boldsymbol{L}_s$, $\boldsymbol{L}_k$, and $\boldsymbol{L}_t$:

1. $L \in \boldsymbol{L}_s$, $L \notin \boldsymbol{L}_t$ and $L \in \boldsymbol{L}_k$, result: $\mathbf{I}_{|\boldsymbol{D}(L)| \times |\boldsymbol{D}(L)|} \otimes \mathbf{J}_{|\boldsymbol{D}(L)| \times 1} = \mathbf{J}_{|\boldsymbol{D}(L)| \times 1}$
2. $L \in \boldsymbol{L}_s$, $L \notin \boldsymbol{L}_t$ and $L \notin \boldsymbol{L}_k$, result: $\mathbf{I}_{|\boldsymbol{D}(L)| \times |\boldsymbol{D}(L)|} \otimes \mathbf{J}_{|\boldsymbol{D}(L)| \times 1} = \mathbf{J}_{|\boldsymbol{D}(L)| \times 1}$
3. $L \notin \boldsymbol{L}_s$, $L \in \boldsymbol{L}_t$ and $L \in \boldsymbol{L}_k$, result: $\mathbf{J}_{1 \times |\boldsymbol{D}(L)|} \otimes \mathbf{I}_{|\boldsymbol{D}(L)| \times |\boldsymbol{D}(L)|} = \mathbf{J}_{1 \times |\boldsymbol{D}(L)|}$
4. $L \notin \boldsymbol{L}_s$, $L \in \boldsymbol{L}_t$ and $L \notin \boldsymbol{L}_k$, result: $1 \otimes \mathbf{J}_{1 \times |\boldsymbol{D}(L)|} = \mathbf{J}_{1 \times |\boldsymbol{D}(L)|}$
5. $L \notin \boldsymbol{L}_s$, $L \notin \boldsymbol{L}_t$ and $L \in \boldsymbol{L}_k$, result: $\mathbf{J}_{1 \times |\boldsymbol{D}(L)|} \otimes \mathbf{J}_{|\boldsymbol{D}(L)| \times 1} = |\boldsymbol{D}(L)|$
6. $L \in \boldsymbol{L}_s$, $L \in \boldsymbol{L}_t$ and $L \in \boldsymbol{L}_k$, result: $\mathbf{I}_{|\boldsymbol{D}(L)|} \otimes \mathbf{I}_{|\boldsymbol{D}(L)| \times |\boldsymbol{D}(L)|} = \mathbf{I}_{|\boldsymbol{D}(L)| \times |\boldsymbol{D}(L)|}$
7. $L \in \boldsymbol{L}_s$, $L \in \boldsymbol{L}_t$ and $L \notin \boldsymbol{L}_k$, result: $\mathbf{J}_{|\boldsymbol{D}(L)| \times 1} \otimes \mathbf{J}_{1 \times |\boldsymbol{D}(L)|} = \mathbf{J}_{|\boldsymbol{D}(L)| \times |\boldsymbol{D}(L)|}$

The structure in Eq. (49) defines that only permutations exist for logvars in the overlap between $\boldsymbol{L}_s$ and $\boldsymbol{L}_t$. The seven cases confirm this structure. Cases 1 and 2 result in the same Kronecker factor independently of L_k. The same holds for Cases 3 and 4. Case 5 adds a $|\boldsymbol{D}(L)|$ factor to all summands but does not affect the structure. Cases 6 and 7 are the only cases where L is part of the overlap between $\boldsymbol{L}_s$ and $\boldsymbol{L}_t$. Here, the structure is dependent on L being in $\boldsymbol{L}_k$, but L will also be part in the permutation set of s and t anyway. Each result is stored in the $\boldsymbol{\rho}$ vector at the position corresponding to the Kronecker sequence. Every entry of the new $\boldsymbol{\rho}^{(i,j)}$ vector can be calculated as follows:

$$\rho_{s,t}^{\boldsymbol{q}_{s,t}} = \sum_{n=1}^{t-1} \sum_{\boldsymbol{q} \in \sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho_{s,n}^{\boldsymbol{q}} \zeta_{n,t} \prod_{L \in (\boldsymbol{L}_n \setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})}, \tag{51}$$

where the selection in the second sum is controlling that the results are stored in the ρ-entry corresponding to the correct Kronecker sequence.

The transpose calculation in Eq. (33) stays in the same structure implied by Eq. (47):

$$\begin{aligned}&\boldsymbol{\Sigma}^T_{\boldsymbol{gr}(Y_s,Y_t)} \\ =&\left(\sum_{\boldsymbol{q}_{s,t} \in \boldsymbol{\Phi}_{s,t}} \rho_{s,t}^{\boldsymbol{q}_{s,t}} \bigotimes_{L \in seq(\boldsymbol{L}_s,\boldsymbol{L}_t)} \mathbf{J}_{dim(L,\boldsymbol{L}_s) \times dim(L,\boldsymbol{L}_t)}^{qexp(\boldsymbol{q}_{s,t},L)} \right)^T \\ =& \sum_{\boldsymbol{q}_{s,t} \in \boldsymbol{\Phi}_{s,t}} \rho_{s,t}^{\boldsymbol{q}_{s,t}} \bigotimes_{L \in seq(\boldsymbol{L}_s,\boldsymbol{L}_t)} \mathbf{J}_{dim(L,\boldsymbol{L}_t) \times dim(L,\boldsymbol{L}_s)}^{qexp(\boldsymbol{q}_{s,t},L)} \\ =&\boldsymbol{\Sigma}_{\boldsymbol{gr}(Y_t,Y_s)}.\end{aligned} \tag{52}$$

Based on the equation for calculating a new off-diagonal block, we apply Eq. (34) to calculate a new on-diagonal block. The only difference to the off-diagonal case is that we also add the PRV variance λ_{Y_t} to the diagonal after calculating the $\boldsymbol{\rho}$ vector using Eq. (51), meaning, we add the PRV variance λ_{Y_t} to the one summand indexed by $\mathbf{0}$:

$$\rho_{t,t}^{\mathbf{0}} \leftarrow \rho_{t,t}^{\mathbf{0}} + \lambda_{Y_t}. \tag{53}$$

The last part of the proof is to show that the starting point is in line with our assumptions. Based on Eq. (31), the vector $\boldsymbol{\rho}_{Y_1,Y_1}$ contains only one non-zero value λ_{Y_1} at position $\boldsymbol{\rho}_{Y_1,Y_1}^{\mathbf{0}}$. This position refers to the Kronecker sequence full of identity matrices.

Table 1
Variables in ground and lifted representation of the joint distribution.

	Grounded		Lifted	
	Variable	Dimensionality	Variable	Dimensionality
Randvars/PRVs	$\mathcal{X}$	N	$\mathcal{Y}$	M
Covariance	$\boldsymbol{\Sigma}$	$N \times N$	$\boldsymbol{\rho}$	$M \times M \times 2^{\lvert\mathcal{L}\rvert}$
Mean	$\boldsymbol{\mu}$	N	$\boldsymbol{\eta}$	M
Domain sizes	implicit		$\boldsymbol{\tau}$	$\lvert\mathcal{L}\rvert$

Because of being in the structure of Eq. (46), the vector $\boldsymbol{\rho}_{Y_1,Y_1}$ serves as a valid starting point. We began the proof by suggesting a closed structure for the blocks in the covariance matrix. We have shown that an inductive creation of the covariance matrix keeps the structure and has formulated a valid starting point. Thus we have a valid lifted representation together with lifted operations for constructing and storing the covariance matrix. □

The proof for Lemma 3 has also equipped us with an expression to calculate the lifted covariance matrix, stored in the $\boldsymbol{\rho}$ tensor containing $M \cdot M$ lifted vectors, where each vector $\boldsymbol{\rho}_{s,t}$ has a dimensionality of $2^{|\boldsymbol{L}_s \cap \boldsymbol{L}_t|}$. The lemma enables us to calculate the lifted joint distribution for any parameterized GBN and store it in a set of lifted variables. Remark: In the overlapping case, $\boldsymbol{\rho}$ is a tensor, which reduces to a matrix in the non-overlapping case.

Table 1 contains an overview of the ground and lifted way to store the joint distribution. The dimensionality of $\boldsymbol{\rho}$ is the worst case, only occurring if there are logvar sequence overlaps involving all logvars in $\mathcal{L}$.

We have now described an algorithm for constructing a lifted version of the covariance matrix of the joint distribution. The lifted mean vector and the lifted version of the covariance matrix together form the *lifted joint distribution*. For better reference of the covariance structure described by Eq. (47) in later calculations, we define terms to refer to the elements of the structure:

Definition 12 *(Kronecker component block, Kronecker permutation set).* We define a *Kronecker component block* as a matrix that follows the structure of Eq. (47). We call all possible Kronecker sequences occurring in the Kronecker component block the *Kronecker permutation set.*

Before moving on the defining matrix operations for the lifted joint distribution, we look at an example. Example 8 describes constructing the lifted covariance matrix in the overlapping scenario visualized in the orange, solid box of Fig. 2.

Example 8. We build upon the transition matrix $\boldsymbol{T}$ set up in Example 7. In the formulas, we assume that the index variables $\boldsymbol{q}$ have an implicit connection to the logvars they are representing. When writing the values of $\boldsymbol{q}_{s,t}$ for example, the link to $\boldsymbol{L}_s$ and $\boldsymbol{L}_t$ is lost. Therefore, we subscript the 0 and 1 values with the logvar they are connected to. Setting up the first element of the lifted covariance matrix is straight forward:

$$\boldsymbol{\rho}_{E(M),E(M)} = \begin{pmatrix} {}^{0_M}\rho_{1,1} \\ {}^{1_M}\rho_{1,1} \end{pmatrix} = \begin{pmatrix} \lambda_{E(M)} \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Next, we calculate our first off-diagonal entry based on Eq. (51):

$$\begin{aligned} \boldsymbol{\rho}_{E(M),S(P)} &= \begin{pmatrix} {}^{1_M 1_P}\rho_{1,2} \end{pmatrix} = \left(\sum_{\boldsymbol{q} \in \sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,s})} \rho^{\boldsymbol{q}}_{s,s} \zeta_{s,t} \prod_{L \in (\boldsymbol{L}_s \setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \right) \\ &= \left(({}^{0_M}\rho_{1,1} \cdot 0) + ({}^{1_M}\rho_{1,1} \cdot 0) \right) = \begin{pmatrix} 0 \end{pmatrix}. \end{aligned}$$

With $s = t = pos(S(P)) = 2$, the second on-diagonal block is built by

$$\begin{aligned} \boldsymbol{\rho}_{S(P),S(P)} &= \begin{pmatrix} {}^{0_P}\rho_{2,2} \\ {}^{1_P}\rho_{2,2} \end{pmatrix} = \begin{pmatrix} \sum_{n=1}^{t-1} \sum_{\boldsymbol{q} \in \sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho^{\boldsymbol{q}}_{s,n} \zeta_{n,t} \prod_{L \in (\boldsymbol{L}_n \setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \\ \sum_{n=1}^{t-1} \sum_{\boldsymbol{q} \in \sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho^{\boldsymbol{q}}_{s,n} \zeta_{n,t} \prod_{L \in (\boldsymbol{L}_n \setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \end{pmatrix} + \begin{pmatrix} \lambda_{S(P)} \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \sum_{\boldsymbol{q} \in \sigma_{0_P}(\{1_M 1_P\})} \rho^{\boldsymbol{q}}_{2,1} \zeta_{1,2} \prod_{L \in \{M\}} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \\ \sum_{\boldsymbol{q} \in \sigma_{1_P}(\{1_M 1_P\})} \rho^{\boldsymbol{q}}_{2,1} \zeta_{1,2} \prod_{L \in \{M\}} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \end{pmatrix} + \begin{pmatrix} \lambda_{S(P)} \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 + \lambda_{S(P)} \\ {}^{1_M 1_P}\rho_{1,2} \cdot \zeta_{1,2} \cdot |\boldsymbol{D}(M)|^{\pi_M(1_M 1_P)} \end{pmatrix} = \begin{pmatrix} \lambda_{S(P)} \\ 0 \cdot 0 \cdot 3 \end{pmatrix} = \begin{pmatrix} \lambda_{S(P)} \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}. \end{aligned}$$

This result is not surprising, because $\boldsymbol{\rho}_{1,1}$ has the same structure as $\boldsymbol{\rho}_{2,2}$ and in the topological ordering both could be exchanged, because both have no parents. The covariance between $E(M)$ and $I(M,P)$ is the first non-trivial example. Here, $s = pos(E(M) = 1$ and $t = pos(I(M,P)) = 3$:

$$\boldsymbol{\rho}_{1,3} = \begin{pmatrix} \rho_{1,3}^{0_M 1_P} \\ \rho_{1,3}^{1_M 1_P} \end{pmatrix} = \begin{pmatrix} \sum_{n=1}^{t-1} \sum\limits_{\boldsymbol{q}\in\sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho_{s,n}^{\boldsymbol{q}}\zeta_{n,t} \prod\limits_{L\in(\boldsymbol{L}_n\setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \\ \sum_{n=1}^{t-1} \sum\limits_{\boldsymbol{q}\in\sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho_{s,n}^{\boldsymbol{q}}\zeta_{n,t} \prod\limits_{L\in(\boldsymbol{L}_n\setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \end{pmatrix}$$

$$= \begin{pmatrix} \sum\limits_{\boldsymbol{q}\in\sigma_{0_M 1_P}(\{0_M,1_M\})} \rho_{1,1}^{\boldsymbol{q}}\zeta_{1,3} \prod\limits_{L\in\emptyset} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} + \sum\limits_{\boldsymbol{q}\in\sigma_{0_M 1_P}(\{1_M 1_P\})} \rho_{1,2}^{\boldsymbol{q}}\zeta_{2,3} \prod\limits_{L\in\emptyset} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \\ \sum\limits_{\boldsymbol{q}\in\sigma_{1_M 1_P}(\{0_M,1_M\})} \rho_{1,1}^{\boldsymbol{q}}\zeta_{1,3} \prod\limits_{L\in\emptyset} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} + \sum\limits_{\boldsymbol{q}\in\sigma_{1_M 1_P}(\{1_M 1_P\})} \rho_{1,2}^{\boldsymbol{q}}\zeta_{2,3} \prod\limits_{L\in\emptyset} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \end{pmatrix}$$

$$= \begin{pmatrix} \rho_{1,1}^{0_M}\zeta_{1,3} \\ \rho_{1,1}^{1_M}\zeta_{1,3} + \rho_{1,2}^{1_M 1_P}\zeta_{2,3} \end{pmatrix} = \begin{pmatrix} 1\cdot 2 \\ 0\cdot 2 + 0\cdot 5 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

For $s = pos(S(P)) = 2$ and $t = pos(I(M,P)) = 3$, the block is

$$\boldsymbol{\rho}_{2,3} = \begin{pmatrix} \rho_{2,3}^{1_M 0_P} \\ \rho_{2,3}^{1_M 1_P} \end{pmatrix} = \begin{pmatrix} \sum_{n=1}^{t-1} \sum\limits_{\boldsymbol{q}\in\sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho_{s,n}^{\boldsymbol{q}}\zeta_{n,t} \prod\limits_{L\in(\boldsymbol{L}_n\setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \\ \sum_{n=1}^{t-1} \sum\limits_{\boldsymbol{q}\in\sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho_{s,n}^{\boldsymbol{q}}\zeta_{n,t} \prod\limits_{L\in(\boldsymbol{L}_n\setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \end{pmatrix}$$

$$= \begin{pmatrix} \sum\limits_{\boldsymbol{q}\in\sigma_{1_M 0_P}(\{1_M 1_P\})} \rho_{2,1}^{\boldsymbol{q}}\zeta_{1,3} \prod\limits_{L\in\emptyset} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} + \sum\limits_{\boldsymbol{q}\in\sigma_{1_M 0_P}(\{0_P,1_P\})} \rho_{2,2}^{\boldsymbol{q}}\zeta_{2,3} \prod\limits_{L\in\emptyset} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \\ \sum\limits_{\boldsymbol{q}\in\sigma_{1_M 1_P}(\{1_M 1_P\})} \rho_{2,1}^{\boldsymbol{q}}\zeta_{1,3} \prod\limits_{L\in\emptyset} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} + \sum\limits_{\boldsymbol{q}\in\sigma_{1_M 1_P}(\{0_P,1_P\})} \rho_{2,2}^{\boldsymbol{q}}\zeta_{2,3} \prod\limits_{L\in\emptyset} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \end{pmatrix}$$

$$= \begin{pmatrix} \rho_{2,2}^{0_P}\zeta_{2,3} \\ \rho_{2,1}^{1_M 1_P}\zeta_{1,3} + \rho_{2,2}^{1_P}\zeta_{2,3} \end{pmatrix} = \begin{pmatrix} 2\cdot 5 \\ 0\cdot 2 + 0\cdot 5 \end{pmatrix} = \begin{pmatrix} 10 \\ 0 \end{pmatrix}.$$

For $s = pos(I(M,P)) = 3$ and $t = pos(I(M,P)) = 3$, we omit the product in the second step, because the set $\boldsymbol{L}_n \setminus \boldsymbol{L}_t$ is always empty, and get

$$\boldsymbol{\rho}_{3,3} = \begin{pmatrix} \rho_{3,3}^{0_M 0_P} \\ \rho_{3,3}^{0_M 1_P} \\ \rho_{3,3}^{1_M 0_P} \\ \rho_{3,3}^{1_M 1_P} \end{pmatrix} = \begin{pmatrix} \sum_{n=1}^{t-1} \sum\limits_{\boldsymbol{q}\in\sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho_{s,n}^{\boldsymbol{q}}\zeta_{n,t} \prod\limits_{L\in(\boldsymbol{L}_n\setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \\ \sum_{n=1}^{t-1} \sum\limits_{\boldsymbol{q}\in\sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho_{s,n}^{\boldsymbol{q}}\zeta_{n,t} \prod\limits_{L\in(\boldsymbol{L}_n\setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \\ \sum_{n=1}^{t-1} \sum\limits_{\boldsymbol{q}\in\sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho_{s,n}^{\boldsymbol{q}}\zeta_{n,t} \prod\limits_{L\in(\boldsymbol{L}_n\setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \\ \sum_{n=1}^{t-1} \sum\limits_{\boldsymbol{q}\in\sigma_{\boldsymbol{q}_{s,t}}(\boldsymbol{Q}_{s,n})} \rho_{s,n}^{\boldsymbol{q}}\zeta_{n,t} \prod\limits_{L\in(\boldsymbol{L}_n\setminus \boldsymbol{L}_t)} |\boldsymbol{D}(L)|^{\pi_L(\boldsymbol{q})} \end{pmatrix} + \begin{pmatrix} \lambda_{I(M,P)} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} \sum\limits_{\boldsymbol{q}\in\sigma_{0_M 0_P}(\{0_M 1_P,1_M 1_P\})} \rho_{3,1}^{\boldsymbol{q}}\zeta_{1,3} + \sum\limits_{\boldsymbol{q}\in\sigma_{0_M 0_P}(\{1_M 0_P,1_M 1_P\})} \rho_{3,2}^{\boldsymbol{q}}\zeta_{2,3} \\ \sum\limits_{\boldsymbol{q}\in\sigma_{0_M 1_P}(\{0_M 1_P,1_M 1_P\})} \rho_{3,1}^{\boldsymbol{q}}\zeta_{1,3} + \sum\limits_{\boldsymbol{q}\in\sigma_{0_M 1_P}(\{1_M 0_P,1_M 1_P\})} \rho_{3,2}^{\boldsymbol{q}}\zeta_{2,3} \\ \sum\limits_{\boldsymbol{q}\in\sigma_{1_M 0_P}(\{0_M 1_P,1_M 1_P\})} \rho_{3,1}^{\boldsymbol{q}}\zeta_{1,3} + \sum\limits_{\boldsymbol{q}\in\sigma_{1_M 0_P}(\{1_M 0_P,1_M 1_P\})} \rho_{3,2}^{\boldsymbol{q}}\zeta_{2,3} \\ \sum\limits_{\boldsymbol{q}\in\sigma_{1_M 1_P}(\{0_M 1_P,1_M 1_P\})} \rho_{3,1}^{\boldsymbol{q}}\zeta_{1,3} + \sum\limits_{\boldsymbol{q}\in\sigma_{1_M 1_P}(\{1_M 0_P,1_M 1_P\})} \rho_{3,2}^{\boldsymbol{q}}\zeta_{2,3} \end{pmatrix} + \begin{pmatrix} \lambda_{I(M,P)} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ \rho_{3,1}^{0_M 1_P}\zeta_{1,3} \\ \rho_{3,2}^{1_M 0_P}\zeta_{2,3} \\ \rho_{3,1}^{1_M 1_P}\zeta_{1,3} + \rho_{3,2}^{1_M 1_P}\zeta_{2,3} \end{pmatrix} + \begin{pmatrix} \lambda_{I(M,P)} \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 2\cdot 2 \\ 10\cdot 5 \\ 0\cdot 2 + 0\cdot 5 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 50 \\ 0 \end{pmatrix}.$$

Next, we present operations to work with the lifted joint to then develop a lifted query answering algorithm.

6. Lifted matrix operations for the lifted joint distribution

In this section, we develop a framework for working with the lifted joint distribution as a prerequisite for answering queries in the next section. For conditional query answering using Eq. (4) and Eq. (5), we need operations for addition and subtraction, for multiplication, and for inversion. We have already used some of these operations in the construction of the covariance matrix, but here we define them more formally. We apply these operations to blocks that follow the same structure as the blocks in the covariance matrix.

6.1. Addition and subtraction

Addition and subtraction of Kronecker component blocks is simple. The precondition is that the Kronecker component blocks share the same Kronecker permutation set. The factors belonging to the same Kronecker sequence in the Kronecker component blocks are added or subtracted to get the resulting Kronecker component block.

Example 9. Two Kronecker component blocks of the same Kronecker permutation set are added:

$$\begin{pmatrix} 5^{0_L} \\ 3^{1_L} \end{pmatrix} + \begin{pmatrix} 2^{0_L} \\ 2^{1_L} \end{pmatrix} = \begin{pmatrix} 7^{0_L} \\ 5^{1_L} \end{pmatrix}$$

6.2. Mutliplication

Multiplying two Kronecker component blocks $\boldsymbol{\Sigma}_{Y_s,Y_k}$ and $\boldsymbol{\Sigma}_{Y_k,Y_t}$ comes down to a repeated application of Eq. (49) because every summand of $\boldsymbol{\Sigma}_{Y_s,Y_k}$ is multiplied with every summand of $\boldsymbol{\Sigma}_{Y_k,Y_t}$, with each summand following the same structure as the $\boldsymbol{T}$ matrix, meaning we can calculate the values in the resulting Kronecker component block in a lifted way as follows:

$$\rho^{\boldsymbol{q}_{s,t}} = \sum_{\boldsymbol{q}_{s,k}=\mathbf{0}}^{\mathbf{1}} \sum_{\boldsymbol{q}_{k,t}=\mathbf{0}}^{\mathbf{1}} id(\boldsymbol{q}_{s,t}, or(\boldsymbol{q}_{s,k}, \boldsymbol{q}_{k,t}))\rho_{s,k}^{\boldsymbol{q}_{s,k}} \rho_{k,t}^{\boldsymbol{q}_{k,t}} \prod_{L \in \boldsymbol{L}_k} |\boldsymbol{D}(L)|^{lexp(L, \boldsymbol{q}^{(s,k)}, \boldsymbol{q}^{(k,t)})}, \tag{54}$$

with $or(\boldsymbol{q}_{s,k}, \boldsymbol{q}_{k,t})$ as the bitwise or-operation, a function id that returns 1 if the resulting logvar sequence of the or-operation is identical to the values in $\boldsymbol{q}_{(s,t)}$ at the logvar positions referenced in $\boldsymbol{q}_{(s,t)}$, i.e.,

$$id(\boldsymbol{q}_{s,t}, \boldsymbol{q}) = \begin{cases} 1 & \text{if } \pi_{\boldsymbol{q}_{s,t}}(\boldsymbol{q}) = \boldsymbol{q}_{s,t}, \\ 0 & \text{otherwise,} \end{cases} \tag{55}$$

and a function $lexp$

$$lexp(L, \boldsymbol{q}_{s,k}, \boldsymbol{q}_{k,t}) = \begin{cases} 1 & (L \in \boldsymbol{L}_k \setminus \boldsymbol{L}_s \vee \pi_L(\boldsymbol{q}_{s,k}) = \mathbf{1}) \wedge (L \in \boldsymbol{L}_k \setminus \boldsymbol{L}_t \vee \pi_L(\boldsymbol{q}_{k,t}) = \mathbf{1}), \\ 0 & \text{otherwise,} \end{cases} \tag{56}$$

which returns 1 if the referenced logvar L fulfills both parts of a conjunction. The first part asks that L occurs only in $\boldsymbol{L}_k$ and not $\boldsymbol{L}_s$ or that the value of L in $\boldsymbol{q}_{s,k}$ is 1, which can only happen if the first disjunct is false. The second part asks the same regarding t instead of s. The returned value as an exponent ensures that the factor of the domain size $|\boldsymbol{D}(L)|$ occurs whenever two all-ones matrices meet.

The main idea of Eq. (54) is to take into account all possible combinations of $\mathbf{J}$ and $\mathbf{I}$ matrices that occur in the summations of both blocks. Equation (54) shows that the Kronecker component block structure is closed under multiplication, which is necessary to combine it with other operations or to multiply several Kronecker component blocks without leaving the structure. Block matrix equation rules allow us to use Eq. (54) also for multiplying rows or matrices of structured blocks as long as the dimensions match.

Example 10. We take the resulting $\boldsymbol{\rho}_{2,3}$ and $\boldsymbol{\rho}_{3,3}$ Kronecker component blocks from Example 8:

$$\boldsymbol{\rho}_{res} = \begin{pmatrix} \boldsymbol{\rho}_{res}^{1_M 0_P} \\ \boldsymbol{\rho}_{res}^{1_M 1_P} \end{pmatrix}$$

We focus on calculating $\boldsymbol{\rho}_{res}^{1_M 0_P}$, the other entry is calculated analogously. In the iteration, we calculate the id operation for all combinations of elements of $\boldsymbol{\Phi}_{2,3}$ and $\boldsymbol{\Phi}_{3,3}$. For better readability, we calculate them separately:

$$\begin{pmatrix} id(1_M 0_P, or(1_M 0_P, 0_M 0_P)) \\ id(1_M 0_P, or(1_M 0_P, 0_M 1_P)) \\ id(1_M 0_P, or(1_M 0_P, 1_M 0_P)) \\ id(1_M 0_P, or(1_M 0_P, 1_M 1_P)) \\ id(1_M 0_P, or(1_M 1_P, 0_M 0_P)) \\ id(1_M 0_P, or(1_M 1_P, 0_M 1_P)) \\ id(1_M 0_P, or(1_M 1_P, 1_M 0_P)) \\ id(1_M 0_P, or(1_M 1_P, 1_M 1_P)) \end{pmatrix} = \begin{pmatrix} id(1_M 0_P, 1_M 0_P) \\ id(1_M 0_P, 1_M 1_P) \\ id(1_M 0_P, 1_M 0_P) \\ id(1_M 0_P, 1_M 1_P) \\ id(1_M 0_P, 1_M 1_P) \\ id(1_M 0_P, 1_M 1_P) \\ id(1_M 0_P, 1_M 1_P) \\ id(1_M 0_P, 1_M 1_P) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

We only write down the elements of the sum that have a result of the id function of 1. Then, we have four different calls of $lexp$ function, which we also write down separately for better readability:

$$\begin{pmatrix} lexp(M, 1_M 0_P, 0_M 0_P) \\ lexp(P, 1_M 0_P, 0_M 0_P) \\ lexp(M, 1_M 0_P, 1_M 0_P) \\ lexp(P, 1_M 0_P, 1_M 0_P) \end{pmatrix} = \begin{pmatrix} (True \vee True) \wedge (False \vee False) \\ (False \vee False) \wedge (False \vee False) \\ (True \vee True) \wedge (False \vee True) \\ (False \vee False) \wedge (False \vee False) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Then, the resulting ρ entry is calculated as follows:

$$\begin{aligned} \rho_{res}^{1_M 0_P} &= \sum_{\boldsymbol{q}_{2,3} \in \boldsymbol{\Phi}_{2,3}} \sum_{\boldsymbol{q}_{3,3} \in \boldsymbol{\Phi}_{3,3}} id(1_M 0_P, or(\boldsymbol{q}_{2,3}, \boldsymbol{q}_{3,3})) \rho_{2,3}^{\boldsymbol{q}_{2,3}} \rho_{3,3}^{\boldsymbol{q}_{3,3}} \prod_{L \in \{M,P\}} |\boldsymbol{D}(L)|^{lexp(L, \boldsymbol{q}^{(2,3)}, \boldsymbol{q}^{(3,3)})} \\ &= \rho_{2,3}^{1_M 0_P} \rho_{3,3}^{0_M 0_P} \prod_{L \in \{M,P\}} |\boldsymbol{D}(L)|^{lexp(L, 1_M 0_P, 0_M 0_P)} \\ &\quad + \rho_{2,3}^{1_M 0_P} \rho_{3,3}^{1_M 0_P} \prod_{L \in \{M,P\}} |\boldsymbol{D}(L)|^{lexp(L, 1_M 0_P, 1_M 0_P)} \\ &= \rho_{2,3}^{1_M 0_P} \rho_{3,3}^{0_M 0_P} \cdot 1 + \rho_{2,3}^{1_M 0_P} \rho_{3,3}^{1_M 0_P} |\boldsymbol{D}(M)| = 10 \cdot 3 \cdot 1 + 10 \cdot 50 \cdot 3 = 1530. \end{aligned}$$

The other entries of the resulting $\boldsymbol{\rho}$ vector are calculated analogously.

6.3. Lifted matrix inversion

In this section, we look at how to invert a matrix following the structure from Lemma 3. We start by understanding how to invert individual blocks of the matrix and then apply the block matrix inversion algorithm.

6.3.1. Inverting a diagonal Kronecker component block

The Kronecker component blocks on the diagonal follow a specific form researched by Searle and Henderson [78]. For this inversion, we use Lemma 4 by Searle and Henderson [78] for which they provide a detailed deduction and proof in their work.

Lemma 4. *Given*

$$\boldsymbol{V}_p = \sum_{\boldsymbol{q}=\boldsymbol{0}}^{\boldsymbol{1}} \theta_{\boldsymbol{q}} (\mathbf{J}_{n_p}^{q_p} \otimes \ldots \otimes \mathbf{J}_{n_p}^{q_1}), \tag{57}$$

of order $N_p = \prod_{r=1}^{p} n_r$ *where* q_i *refers to the entries in* $\boldsymbol{q}$ *and* n_i *to each dimension, the inverse is given by*

$$\boldsymbol{V}_p^{-1} = \sum_{\boldsymbol{q}=\boldsymbol{0}}^{\boldsymbol{1}} \kappa_{\boldsymbol{q}} (\mathbf{J}_{n_p}^{q_p} \otimes \ldots \otimes \mathbf{J}_{n_p}^{q_1}), \tag{58}$$

where

$$\boldsymbol{R}_p = \begin{bmatrix} 1 & 0 \\ 1 & n_p \end{bmatrix} \otimes \ldots \otimes \begin{bmatrix} 1 & 0 \\ 1 & n_1 \end{bmatrix} \tag{59}$$

and

$$\boldsymbol{\kappa} = \boldsymbol{R}^{-1} \frac{1}{\boldsymbol{R}\boldsymbol{\theta}}. \tag{60}$$

There are two main benefits of Lemma 4. First, it works fully in a lifted way because it does not involve any matrix operations dependent on the cardinality of the all-ones and identity matrices. Second, it stays within the Kronecker component block structure, making it possible to use the result in all upcoming calculations analogously to working with other blocks in the covariance matrix.

Interestingly, the approach used in the non-overlapping scenario [6] is a special case of Lemma 4, where the Kronecker sequences consist of only one term. Lemma 4 simplifies into Lemma 5 [78].

Algorithm 2 Lifted recursive block matrix inversion.

function LIFTEDINVERSION(ρ, τ) ▷ lifted representation as an input
  if $\lambda.size = 1$ **then**
    $\rho_{inv} \leftarrow \text{INV}(\rho, \tau)$ ▷ based on Lemma 4
    return ρ_{inv}
  $\rho_{\tilde{A}} \leftarrow \rho_{1,1}$
  $\rho_{\tilde{B}}, \rho_{\tilde{C}} \leftarrow \rho_{1,2:K}, \rho_{2:K,1}$
  $\rho_{\tilde{D}} \leftarrow \rho_{2:K,2:K}$
  $\rho_{\tilde{A}^{-1}} \leftarrow \text{LIFTEDINVERSION}(\rho_{\tilde{A}}, \tau)$ ▷ recursive call
  $\rho_{\tilde{F}} \leftarrow \rho_{\tilde{D}} - \text{MULTI}(\rho_{\tilde{C}}, \rho_{\tilde{A}^{-1}}, \rho_{\tilde{B}}, \tau)$ ▷ based on Eq. (54)
  $\rho_{\tilde{F}^{-1}} \leftarrow \text{LIFTEDINVERSION}(\rho_{\tilde{F}}, \tau)$ ▷ recursive call
  $\rho_O, \rho_P, \rho_Q, \rho_R \leftarrow \text{MULTI}(\rho_{\tilde{A}}, \rho_{\tilde{B}}\rho_{\tilde{C}}, \rho_{\tilde{F}})$ ▷ based on Eqs. (54) and (64)
  $\rho_{inv} \leftarrow \text{STACK}(\rho_O, \rho_P, \rho_Q, \rho_R)$ ▷ based on Eq. (64)
  return ρ_{inv}, τ ▷ lifted representation as an output

Lemma 5. *Let $\boldsymbol{L}$ be an $G \times G$ matrix of the form $\boldsymbol{L} = a\mathbf{J} + b\mathbf{I}$. Then, the inverse of L can be calculated analytically by $\boldsymbol{L}^{-1} = x\mathbf{J} + y\mathbf{I}$, where*

$$x = -\frac{a}{b(aG+b)} \text{ and } y = \frac{1}{b}. \tag{61}$$

Proof. To prove this lemma, we follow the definition of an inverse $\boldsymbol{L}\boldsymbol{L}^{-1} = \mathbf{I}$. Solving the equation $\boldsymbol{L}\boldsymbol{L}^{-}1 = (a\mathbf{J} + b\mathbf{I})(x\mathbf{J} + y\mathbf{I}) = \mathbf{I}$ is equivalent to solving the linear equation system (LES)

$$(a+b)(x+y) + (G-1)ax = 1, \tag{62}$$

$$(a+b)x + a(x+y) + (G-2)ax = 0. \tag{63}$$

Solving the LES for x and y results in Equation (61). □

Now that we can invert individual blocks on the diagonal and multiply them with other blocks from the covariance matrix without changing the structure, we can apply the block matrix inversion algorithm.

6.3.2. Block matrix inversion

To break down the inversion, we use the block matrix inversion formula [79] with $\tilde{\boldsymbol{F}} = \tilde{\boldsymbol{D}} - \tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}^{-1}\tilde{\boldsymbol{B}}$:

$$\begin{bmatrix} \tilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}} \\ \tilde{\boldsymbol{C}} & \tilde{\boldsymbol{D}} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{P} \\ \boldsymbol{Q} & \boldsymbol{R} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{A}}^{-1} + \tilde{\boldsymbol{A}}^{-1}\tilde{\boldsymbol{B}}\tilde{\boldsymbol{F}}^{-1}\tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}^{-1} & -\tilde{\boldsymbol{A}}^{-1}\tilde{\boldsymbol{B}}\tilde{\boldsymbol{F}}^{-1} \\ -\tilde{\boldsymbol{F}}^{-1}\tilde{\boldsymbol{C}}\tilde{\boldsymbol{A}}^{-1} & \tilde{\boldsymbol{F}}^{-1} \end{bmatrix}. \tag{64}$$

To invert a full matrix $\boldsymbol{Z}$ that follows the structure of our covariance matrix, we use the lifted inversion of individual blocks together with lifted matrix multiplication recursively. The steps within the recursive approach are as follows. Pseudocode can be found in Algorithm 2.

Step 1: The input for the function is the lifted representation $\boldsymbol{\rho}$ and $\boldsymbol{\tau}$ of the matrix $\boldsymbol{Z}$. If the matrix $\boldsymbol{\rho}$ has only one element, $\boldsymbol{Z}$ would consist of only one block and we can use Lemma 4 to calculate the lifted representation of $\boldsymbol{Z}^{-1}$. The resulting $\boldsymbol{\rho}_{res}$ is returned and the function call terminates. If $\boldsymbol{\rho}$ has more than one element the algorithm continues.

Step 2: The grounded matrix $\boldsymbol{Z}$ would be split into four blocks based on Eq. (64). These four blocks are constructed of the $K \times K$ blocks forming the matrix $\boldsymbol{Z}$ (where K is decreased by one in each recursion step). In our fully lifted algorithm, we analogously split the tensor $\boldsymbol{\rho}$:
- $\tilde{\boldsymbol{A}} = \boldsymbol{B}_{1,1}$, $\rho_{\tilde{\boldsymbol{A}}} = \rho_{1,1}$,
- $\tilde{\boldsymbol{B}} = \boldsymbol{B}_{1,2:K}$, $\boldsymbol{\rho}_{\tilde{\boldsymbol{B}}} = \boldsymbol{\rho}_{1,2:K}$
- $\tilde{\boldsymbol{C}} = \boldsymbol{B}_{1,1}$, $\boldsymbol{\rho}_{\tilde{\boldsymbol{C}}} = \boldsymbol{\rho}_{2:K,1}$
- $\tilde{\boldsymbol{D}} = \boldsymbol{B}_{2:K,2:K}$, $\boldsymbol{\rho}_{\tilde{\boldsymbol{D}}} = \boldsymbol{\rho}_{2:K,2:K}$

Step 3: The inversion function is called for the lifted representation of $\tilde{\boldsymbol{A}}$ and Step 1 will directly return the lifted inverse. The lifted version $\boldsymbol{\rho}_{\tilde{\boldsymbol{F}}}$ of matrix $\tilde{\boldsymbol{F}}$ can be calculated using the lifted multiplication rules from Eq. (54). The inversion algorithm is recursively called for $\boldsymbol{\rho}_{\tilde{\boldsymbol{F}}}$.

Step 4: Once the recursive call returns the lifted inverse for $\boldsymbol{\rho}_{\tilde{\boldsymbol{F}}}$, $\boldsymbol{\lambda}_{\tilde{\boldsymbol{F}}}$ and $\boldsymbol{\tau}_{\tilde{\boldsymbol{F}}}$, the lifted representations for blocks $\boldsymbol{O}$, $\boldsymbol{P}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be calculated using Eq. (54). Based on Eq. (64), the four lifted representations of $\boldsymbol{O}$, $\boldsymbol{P}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are combined into $\boldsymbol{\rho}_{\boldsymbol{Z}^{-1}}$.

Having now the ability to work with the lifted version of the covariance matrix, we look into using this lifted joint representation and the lifted operations to answer queries.

7. Lifted query answering

This section covers query answering using the lifted version of the joint distribution. As described in the preliminaries, in query answering, we need to calculate conditional probability distributions over query variables $\boldsymbol{Q} \subset \mathcal{X}$ given evidence $\boldsymbol{E} = \boldsymbol{e}$ with $\boldsymbol{E} \subset \mathcal{X}$. The main calculations are introduced in Eq. (4) and Eq. (5), which we repeat here for better readability:

$$\boldsymbol{\mu}^* = \boldsymbol{\mu}_{\boldsymbol{Q}} + \boldsymbol{\Sigma}_{\boldsymbol{QE}}\boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}(\boldsymbol{e} - \boldsymbol{\mu}_{\boldsymbol{E}}), \tag{65}$$

$$\boldsymbol{\Sigma}^* = \boldsymbol{\Sigma}_{\boldsymbol{QQ}} - \boldsymbol{\Sigma}_{\boldsymbol{QE}}\boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{EQ}}. \tag{66}$$

We define a few helper functions:

Definition 13 *(const, perm, gr).* Given a set of grounded randvars $\boldsymbol{X}$, a PRV Y, and a sequence of constants $\boldsymbol{l}$, we define the following three helper functions:

$$const_{\boldsymbol{L}}(\boldsymbol{X}) = \{\sigma_{\boldsymbol{L}}(\boldsymbol{l}) \mid X(\boldsymbol{l}) \in \boldsymbol{X}\} \quad \text{(constants of } \boldsymbol{L} \text{ occurring in } \boldsymbol{X}\text{)}, \tag{67}$$

$$perm(\boldsymbol{X}) = \{\boldsymbol{l} \mid \boldsymbol{l} \in \times_{L \in lv(lif(\boldsymbol{X}))} const_{\{L\}}(\boldsymbol{X})\} \quad \text{(cross product of } const_{\{L\}}(\boldsymbol{X}) \text{ over } lv(lif(\boldsymbol{X}))\text{)}, \tag{68}$$

$$gr(Y, \{\boldsymbol{l}_i\}_i) = \{X(\boldsymbol{l}') \mid X(\boldsymbol{l}') \in \boldsymbol{gr}(Y) \wedge \pi_{\boldsymbol{L}_Y}(\boldsymbol{l}) = \boldsymbol{l}' \wedge \boldsymbol{l} \in \{\boldsymbol{l}_i\}_i\} \quad (Y \text{ grounded with constants of each } \boldsymbol{l}_i\text{}). \tag{69}$$

7.1. Lifted answering of a marginal query

A marginal query $P(\boldsymbol{Q})$ is a query without evidence. Obtaining a marginal distribution of a multivariate normal distribution is trivial, because all matrix calculations in Eq. (65) and Eq. (66) can be omitted when having an empty matrix $\boldsymbol{\Sigma}_{\boldsymbol{EE}}$. One can simply select the means $\boldsymbol{\mu}_{\boldsymbol{Q}}$ and covariance sub-matrix $\boldsymbol{\Sigma}_{\boldsymbol{QQ}}$ that are corresponding to the queried randvars and insert them into the probability distribution

$$P(\boldsymbol{Q}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{Q}}; \boldsymbol{\Sigma}_{\boldsymbol{QQ}}). \tag{70}$$

For the lifted case, we select the means by $\boldsymbol{\mu}_{\boldsymbol{Q}} = \boldsymbol{\eta}_{lif(\boldsymbol{Q})}$, ground the covariance matrix $\boldsymbol{\Sigma}$ with Eq. (47), and then select the elements with $\boldsymbol{\Sigma}_{\boldsymbol{QQ}}$.

7.2. Lifted answering of a conditional query

When answering a conditional query based on Eq. (65) and Eq. (66), we need to apply the following operations:

1. lifted addition when adding the lifted version of $\boldsymbol{\Sigma}_{\boldsymbol{QQ}}$ to $\boldsymbol{\Sigma}_{\boldsymbol{QE}}\boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{EQ}}$,
2. lifted multiplication in $\boldsymbol{\Sigma}_{\boldsymbol{QE}}\boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{EQ}}$ and in $\boldsymbol{\Sigma}_{\boldsymbol{QE}}\boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}$,
3. lifted inversion in $\boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}$, and
4. lifted handling of the ground evidence values $\boldsymbol{e} - \boldsymbol{\mu}_{\boldsymbol{E}}$ and the combination with the lifted result of $\boldsymbol{\Sigma}_{\boldsymbol{QE}}\boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}$.

The covariance matrix $\boldsymbol{\Sigma}$ follows a structure that allows addition, multiplication, and inversion. However, this structure is not necessarily given for $\boldsymbol{\Sigma}_{\boldsymbol{QQ}}, \boldsymbol{\Sigma}_{\boldsymbol{QE}}, \boldsymbol{\Sigma}_{\boldsymbol{EQ}}$, and $\boldsymbol{\Sigma}_{\boldsymbol{EE}}$. For inversion and matrix multiplications, all blocks in these matrices need to follow the Kronecker block structure with matching dimensions, which is given if all ground level observations $\boldsymbol{E}$ and query variables $\boldsymbol{Q}$ have the same set of instantiated logvars. If so, we call such a query liftable. More formally, liftability is defined as:

Definition 14 *(Liftable query).* Given a set of PRVs $\mathcal{Y}$ and a corresponding set of logvars $\mathcal{L}$, a *query* $P(\boldsymbol{Q} | \boldsymbol{E} = \boldsymbol{e})$, with $\boldsymbol{E} \subset \boldsymbol{gr}(\mathcal{Y})$ and $\boldsymbol{Q} \subset \boldsymbol{gr}(\mathcal{Y})$, is *liftable* if it fulfills

$$(\forall Y \in \boldsymbol{lif}(\boldsymbol{E}) : \boldsymbol{gr}(Y, perm(\boldsymbol{E} \cup \boldsymbol{Q})) \in \boldsymbol{E}) \wedge (\forall Y \in \boldsymbol{lif}(\boldsymbol{Q}) : \boldsymbol{gr}(Y, perm(\boldsymbol{E} \cup \boldsymbol{Q})) \in \boldsymbol{E}) \tag{71}$$

Definition 14 ensures that matrices $\boldsymbol{\Sigma}_{\boldsymbol{QQ}}, \boldsymbol{\Sigma}_{\boldsymbol{QE}}, \boldsymbol{\Sigma}_{\boldsymbol{EQ}}$, and $\boldsymbol{\Sigma}_{\boldsymbol{EE}}$ of a liftable query follow the Kronecker block structure needed to perform matrix multiplications and inversions in a lifted way. The restriction excludes all queries that do not follow the Kronecker block structure from being calculated in a lifted way. One example for a non-liftable query would be a query having ground randvars of the same PRV in both evidence set and query set. In the non-overlapping logvar case, Eq. (71) is always true as long as randvars from the same PRV are not present in both query and evidence set. The matrix operations enable lifted calculations of Operations 1-3 in the list above, resulting in the conditional covariance $\boldsymbol{\Sigma}^*$. The last operation (4 in the list above) means calculating the conditional mean $\boldsymbol{\mu}^*$, which involves handling evidence. The easiest way to handle evidence is to do the calculations on ground level, meaning grounding $\boldsymbol{\mu}_{\boldsymbol{Q}}$ and the result of $\boldsymbol{\Sigma}_{\boldsymbol{QE}}\boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}$ to do a grounded matrix multiplication with the ground level vector $\boldsymbol{e} - \boldsymbol{\mu}_{\boldsymbol{E}}$. Even with this grounding, performing the other steps in a lifted way already brings major time savings because more complex matrix operations are still avoided, but we are not satisfied yet. Therefore, we look into cases next where we can group the ground level evidence to further reduce the time complexity. Section 8 contains the detailed complexity discussion for all steps and approaches discussed here.

7.3. Grouping evidence

In general, the difficulty of handling evidence in a lifted way is that evidence itself occurs on a ground level, i.e., evidence contains individual randvars. In the continuous setting, it is very unlikely that the same evidence value will occur for two different randvars. If we assume perfect measure accuracy and a fully continuous setting, the chance for measuring the same value twice is zero (or infinitely close to zero). In the non-overlapping case, different evidence values are not a big deal because every observed randvar of a PRV has the same effect on all randvars belonging to another PRV. Thus, we can group the evidence within each observed PRV by summing up all ground values for the PRV in $\boldsymbol{e} - \boldsymbol{\mu_E}$ and multiplying it with the corresponding term from the lifted result of $\boldsymbol{\Sigma_{QE}\Sigma_{EE}^{-1}}$. The resulting vector along the K PRVs for which we have evidence is given by

$$\begin{bmatrix} \sum_{E_h \in \boldsymbol{gr}(Y_1^E)} (e_h - \mu_h) \\ \vdots \\ \sum_{E_h \in \boldsymbol{gr}(Y_K^E)} (e_h - \mu_h) \end{bmatrix}. \tag{72}$$

In the case of overlapping logvar sequences, summing up the evidence for the whole PRV is not working because every overlap introduces independencies, i.e., not all randvars of one PRV have the same effect on all randvars of another PRV. Next, we look at how we can still lift part of the calculations involved in the overlapping case.

7.3.1. Two-PRV case

We begin with a scenario of only two PRVs before generalizing to multiple PRVs. We have one PRV $Y_Q = lif(\boldsymbol{Q})$ whose instances are (partially) queried and one PRV $Y_E = lif(\boldsymbol{E})$ whose instances are (partially) observed in the query $P(\boldsymbol{Q}|\boldsymbol{E} = \boldsymbol{e})$. Evidence randvars only influence query randvars that share the same constants. In a liftable query based on Definition 14, the number of evidence randvars is equal for each queried variable but the actual observed evidence values can be different. In the ground case, the operation is a multiplication of a Kronecker component block $\boldsymbol{\Sigma_{QE}\Sigma_{EE}^{-1}}$ and the evidence vector $(\boldsymbol{e} - \boldsymbol{\mu_E})$. Every row of the Kronecker component block is multiplied with the same vector $(\boldsymbol{e} - \boldsymbol{\mu_E})$. Non-overlapping logvar sequences between the observed Y_E and the queried PRV Y_Q result in duplicates in the Kronecker component block, in a way that logvars only present in Y_E lead to duplicate columns and logvars only present in Y_Q lead to duplicate rows. Duplicates in the rows will lead to the same conditional mean value for the corresponding ground query randvars and duplicates in the columns will lead to the same influence of evidence randvars onto ground randvars. To combine the randvars with the same influence and result, we group along the overlapping logvars. We denote the overlaps in the logvar sequences as $\boldsymbol{L}_O = \boldsymbol{L}_{lif(\boldsymbol{Q})} \cap \boldsymbol{L}_{lif(\boldsymbol{E})}$ and group the evidence along the constants $\boldsymbol{l}_O$ of the overlapping logvars, with $\boldsymbol{l}_O \in \boldsymbol{D}(\boldsymbol{L}_O)$, and add up the evidence values:

$$sumev_{\boldsymbol{E}}(Y_E, \boldsymbol{l}_O) = \sum e - \mu_E \mid E = e \wedge E \in \boldsymbol{gr}(Y_E, \{\boldsymbol{l}_O\}) \tag{73}$$

The conditional mean vector is grouped along the same set of instance sequences

$$\mu_{Y_Q} = \begin{pmatrix} \mu_{Y_Q, \boldsymbol{l}_O^1} \\ \vdots \\ \mu_{Y_Q, \boldsymbol{l}_O^G} \end{pmatrix} = \begin{pmatrix} \eta_{Y_Q} \\ \vdots \\ \eta_{Y_Q} \end{pmatrix} + \begin{pmatrix} \nu_{Y_Q, \boldsymbol{l}_O^1} \\ \vdots \\ \nu_{Y_Q, \boldsymbol{l}_O^G} \end{pmatrix}, \tag{74}$$

where G is the number of groups and given by $G = |const_{\boldsymbol{L}_O}(\boldsymbol{Q})|$. The entries $\nu_{Y_Q, \boldsymbol{l}_O^g}$, with $g = 1, ..., G$ are given by

$$\nu_{Y_Q, \boldsymbol{l}_O^g} = \nu_{Y_Q, \boldsymbol{l}_O^g, Y_E}, \tag{75}$$

with

$$\nu_{Y_Q, \boldsymbol{l}_O^g, Y_E} = \sum_{\boldsymbol{l}_O \in inst_{\boldsymbol{L}_O}(\boldsymbol{E})} \sum_{\boldsymbol{q} \in \boldsymbol{Q}_{\boldsymbol{L}_Q, \boldsymbol{L}_E}} sumev_{\boldsymbol{E}}(Y_E, \boldsymbol{l}_O^g) \rho^{\boldsymbol{q}}_{Y_Q, Y_E} filter(\boldsymbol{l}_O^g, \boldsymbol{l}_O, \boldsymbol{q}), \tag{76}$$

and

$$filter(\boldsymbol{l}_O^g, \boldsymbol{l}_O, \boldsymbol{q}) = \begin{cases} 1 & \pi_{\sigma_{q=0}(\boldsymbol{q})}(\boldsymbol{l}_O^g) = \pi_{\sigma_{q=0}(\boldsymbol{q})}(\boldsymbol{l}_O), \\ 0 & \text{otherwise}, \end{cases} \tag{77}$$

where q in the selection is a placeholder for all positions in $\boldsymbol{q}$. The filter function makes sure that the independencies between randvars in $\boldsymbol{Q}$ and $\boldsymbol{E}$ are taken into account. In a full overlap, the sum simplifies to Eq. (72).

7.3.2. Multi-PRV case

Having solved the case for one evidence and one query PRV, we now relax further and allow for multiple PRVs. The liftable query rule from Definition 14 still applies, but even with this rule there might be different overlapping sets for each evidence PRV to the queried PRVs. The sequence of all overlapping logvars between any PRV in $\boldsymbol{Q}$ and any PRV in $\boldsymbol{E}$ is still given by $\boldsymbol{L}_O = \boldsymbol{L}_{lif(\boldsymbol{Q})} \cap \boldsymbol{L}_{lif(\boldsymbol{E})}$. For each query PRV Y_Q, the groups for the final mean $\mu_{Y_Q, \boldsymbol{l}_O^g}$ and for the result of the Kronecker component block multiplication $\nu_{Y_Q, \boldsymbol{l}_O^g}$ are given by $inst_{\boldsymbol{L}_O}(\boldsymbol{Q})$, such that $\boldsymbol{l}_O^g \in inst_{\boldsymbol{L}_O}(\boldsymbol{Q})$. If we have more than one evidence PRV, the equality of $\nu_{Y_Q, \boldsymbol{l}_O^g} = \nu_{Y_Q, \boldsymbol{l}_O^g, Y_E}$

in Eq. (75) does not hold anymore, because all evidence randvars might influence the mean values. Equation (76) still works for all individual evidence PRVs and query PRVs but can contain different groupings that need to be combined in a way that the values in $v_{Y_Q,\boldsymbol{l}_O^g,Y_E}$ are sorted into the correct group of $v_{Y_Q,\boldsymbol{l}_O^g}$. The grouping is done by

$$v_{Y_Q,\boldsymbol{l}_O} = \sum_{Y_E \in \boldsymbol{lif}(\boldsymbol{E})} v_{Y_Q,\boldsymbol{l}'_O,Y_E} match(\boldsymbol{l}'_O, \boldsymbol{l}_O), \tag{78}$$

with

$$match(\boldsymbol{l}'_O, \boldsymbol{l}_O) = \begin{cases} 1 & \pi_{\boldsymbol{L}'_O}(\boldsymbol{l}_O) = \boldsymbol{l}'_O, \\ 0 & \text{otherwise}, \end{cases} \tag{79}$$

where $\boldsymbol{L}'_O$ is referring to the logvar sequence corresponding to the constants in $\boldsymbol{l}'_O$. Values calculated in Eq. (78) can then be put into Eq. (74) for the final query answer.

7.4. Example queries and calculations

In this section, we cover a few example queries to illustrate the liftability characteristic and the calculations performed in query answering.

Example 11. Given the lifted joint distribution in Appendix A.3, we define the following queries, where we omit set parentheses and value assignments for better readability.

1. $P(H(P1) \mid E(M1), E(M2), S(P1))$
2. $P(H(P1), H(P2) \mid S(P1), S(P2), I(M1, P1), I(M1, P2), I(M2, P1), I(M2, P2))$
3. $P(U(C1), U(C2) \mid H(P1), H(P2))$
4. $P(H(P1) \mid I(M1, P1), S(P2))$
5. $P(H(P1) \mid H(P2))$

The entries of the evidence vector $\boldsymbol{e}$ for every query are the natural numbers beginning with one.

Query 1: Query 1 is a liftable query and we have only one constant defining the groups for the conditional mean. The query answer is given by:

$$\boldsymbol{\rho}^*_{H(P)} = \begin{pmatrix} {}^{0_P}\rho_{H(P)} \\ {}^{1_P}\rho_{H(P)} \end{pmatrix} = \begin{pmatrix} 1 \\ 6642 \end{pmatrix}; \eta^*_{H(P)} = 48; \Sigma^*_{H(P1)} = 6643; \mu^*_{H(P1)} = 48$$

Query 2: Query 2 is a liftable query and we have two constants defining the groups for the conditional mean. The query answer is given by:

$$\boldsymbol{\rho}^*_{H(P)} = \begin{pmatrix} {}^{0_P}\rho_{H(P)} \\ {}^{1_P}\rho_{H(P)} \end{pmatrix} = \begin{pmatrix} 49 \\ 64 \end{pmatrix}; \begin{pmatrix} \eta^*_{H(P),P1} \\ \eta^*_{H(P),P2} \end{pmatrix} = \begin{pmatrix} 50 \\ 70 \end{pmatrix};$$

$$\boldsymbol{\Sigma}^* = \begin{bmatrix} 113 & 64 \\ 64 & 113 \end{bmatrix}; \begin{pmatrix} \mu^*_{H(P1)} \\ \mu^*_{H(P2)} \end{pmatrix} = \begin{pmatrix} 50 \\ 70 \end{pmatrix}$$

Query 3: Query 3 is a liftable query. We have only one single group because there is no overlap between query and evidence. The query answer is given by:

$$\boldsymbol{\rho}^*_{U(C)} = \begin{pmatrix} {}^{0_P}\rho_{U(C)} \\ {}^{1_P}\rho_{U(C)} \end{pmatrix} = \begin{pmatrix} 3 \\ 152430 \end{pmatrix}; \eta^*_{U(C)} = -2048;$$

$$\boldsymbol{\Sigma}^* = \begin{bmatrix} 152434 & 152430 \\ 152430 & 152434 \end{bmatrix}; \begin{pmatrix} \mu^*_{U(C1)} \\ \mu^*_{U(C2)} \end{pmatrix} = \begin{pmatrix} -2048 \\ -2048 \end{pmatrix}$$

Query 4: Query 4 is a non-liftable query because $P1$ and $P2$ occur in the query but we only have a measurement for $I(M1, P1)$, which contradicts Definition 14.

Query 5: Query 5 is a non-liftable query because randvars of the same PRV occur in the evidence and the query set, which contradicts Definition 14.

We have lifted approaches for constructing the joint distribution and lifted query answering algorithms for working with the lifted joint. We have also shown in a running example how all operations can be used and how queries can be answered in a lifted way. Next, we evaluate the complexity gains by a theoretical complexity analysis.

8. Complexity analysis

This section covers the run-time and space complexity of creating the joint distribution and the run-time complexity of answering conditional probability queries. Let us specify the parameters used for this complexity analysis. We have N randvars in the GBN combined into M PRVs. The terms N_E and N_Q denote the number of randvars in the evidence and the query set respectively, whereas the terms M_E and M_Q denote the number of PRVs referenced in the evidence and query set respectively. The term Λ denotes the number of logvars and S the longest logvar sequence. In general, we focus on the upper bound of the run-time complexity. There are matrix inversion and multiplication algorithms that have a run-time complexity of less than $O(N^3)$, but for simplicity and without changing the overall argumentation, we take $O(N^3)$ as an upper bound [80,81].

8.1. Complexity for constructing the joint distribution

The only time-consuming part, when creating the lifted joint, is to generate the joint covariance matrix, since the mean values of the nodes just need to be stored into a vector (for the lifted and grounded approach).

8.1.1. Ground complexity

Constructing the ground version of the joint probability distribution calculates the $N \times N$-dimensional covariance matrix by iterating in two loops over the randvars. In each step, the covariance between two randvars X_i and X_j, with $j > i$, is calculated by taking into account all randvars V_k that occur earlier than X_j, i.e., $k < j$, in the topological ordering, resulting in an upper bound of $O(N^3)$. The space complexity of storing the covariance matrix $\boldsymbol{\Sigma}$ and the mean vector $\boldsymbol{\mu}$ is $O(N^2 + N) = O(N^2)$. Remark: We have a lower bound of $\Omega(N^2)$, if the network architecture limits the number of parents a randvar has to a constant value independent of N.

8.1.2. Lifted complexity

The lifted approach presented in Section 5 is fully independent on the number of randvars N, independent of whether the network contains overlaps or not. To construct the $M \cdot M$ $\boldsymbol{\rho}$ vectors, we iterate two times over all M PRVs. To calculate an individual $\boldsymbol{\rho}_{s,t}$ vector, we need to iterate over the PRVs Y_u prior in the topological ordering to Y_t, i.e., $u < t$, and for each of these PRVs Y_u, we iterate over at most 2^S possible Kronecker sequences. Combining these iterations results in a overall run-time complexity of $O(M^3 2^S)$. In a parameterized GBN where we typically have $M \ll N$ and $2^S \ll N$, the change in complexity class results in a significant speed up. The space complexity to store all values in $\boldsymbol{\rho}$ is $O(M^2 2^S)$. In the non-overlapping scenario, 2^S reduces to a constant because even if a PRV has a logvar sequence with more than one logvar, the logvars can be combined into a single logvar without influencing the model. When combining logvars, the domain size increases for the combined logvar but this has no effect on the construction run-time as visible in the complexity classes.

8.2. Complexity for conditional query answering

For query answering, we need to look at the four operations listed in Section 7.2.

8.2.1. Ground complexity

Matrix addition has a complexity of $O(N_Q^2)$ because every element in the matrix needs to be visited at least once. The calculation of the evidence vector $\boldsymbol{e} - \boldsymbol{\mu}_{\boldsymbol{E}}$ has a complexity of $O(N_E)$. The inversion of evidence covariance matrix has a complexity of $O(N_E^3)$. The involved matrix multiplications have a complexity of $O(N_Q^2 N_E)$ and $O(N_Q N_E^2)$, respectively. Combining all operations results in a complexity for query answering of $O(N_E^3 + N_Q N_E^2 + N_Q^2 N_E)$. Depending on the partitioning of evidence and query set, either the term N_E^3 or the term $N_Q^2 N_E$ is dominating the complexity. The matrix addition and the calculation of the evidence vector are out-weighted by the other terms.

8.2.2. Lifted complexity

Multiplying two Kronecker component blocks has a complexity of $O(2^{2S})$ because in the worst case we need to iterate fully through all permutations of possible Kronecker sequences. Thus, the matrix multiplications (Items 2 and 3 in the list in Section 7.2) have complexity of $O(M_E^2 M_Q 2^{2S})$ and $O(M_E M_Q^2 2^{2S})$.

The inversion of one individual Kronecker component block from the diagonal of the covariance matrix involves the multiplication of two at most $2^S \times 2^S$ matrices resulting in a complexity of $O(2^{3S})$. In the block matrix inversion we have at most M_E recursive function calls and Kronecker component block multiplications of at most M_E blocks, resulting in a combined inversion complexity of $O(M_E 2^{3S} + M_E^2 2^S)$.

To calculate the mean vector, we still have ground operations. Grounding the result of $\boldsymbol{\Sigma}_{\boldsymbol{QE}} \boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}$ takes $O(N_Q N_E)$ time and multiplying it with the ground level evidence vector $\boldsymbol{e} - \boldsymbol{\mu}_{\boldsymbol{E}}$ is in $O(N_Q N_E)$ as well. This is still a linear relationship to both N_Q and N_E. Overall, this results in a complexity of $O(M_E^2 M_Q 2^{2S} + M_E M_Q^2 2^{2S} + M_E 2^{3S} + N_Q N_E)$.

With the grouping of the evidence along the overlapping logvar instances, we reduce N_Q to G_Q where $G_Q = |\boldsymbol{D}(\boldsymbol{L}_Q \cap \boldsymbol{L}_E)|$ refers to the number of instances that occur both in $\boldsymbol{Q}$ and $\boldsymbol{E}$. The biggest time-savings occur if there is a relatively even split between queried variables and evidence variables, when they have no or little overlap. When there is a full overlap between evidence PRVs

Table 2
Experiment setups.

Experiment	PRVs	Domain sizes	Query
Overlap	$E(M), S(P),$ $I(M,P), H(P)$	$\tau_M, \tau_P = 2^1, ..., 2^K$	$\boldsymbol{Q} = \boldsymbol{gr}(H(P)),$ $\boldsymbol{E} = \boldsymbol{gr}(S(P))$
Mixed	$E(M), S(P), I(M,P),$ $H(P), W(D)$	$\tau_M, \tau_D = 2^1, ..., 2^K,$ $\tau_P = 2$	$\boldsymbol{Q} = \boldsymbol{gr}(W(D)),$ $\boldsymbol{E} = \boldsymbol{gr}(E(M))$
No Overlap	$H(P), T(N),$ $W(D), U(C)$	$\tau_P, \tau_T, \tau_N = 2^1, ..., 2^K,$ $\tau_C = 2$	$\boldsymbol{Q} = \boldsymbol{gr}(U(C)),$ $\boldsymbol{E} = \boldsymbol{gr}(H(P), T(N))$
Large	21 PRVs including 5 logvars	Three domains increasing, two domains constant	19 evidence PRVs, 1 query PRV

and query PRVs, then $G_Q = N_Q$ and the grouping will not lead to efficiency gains. In the non-overlapping case, there is only one group, i.e., $G_Q = 1$, which is consistent with the linear dependency on N_E reported in [6].

Theorem 1. *The complexity of query answering as described in Section 7 including evidence grouping is given by*

$$O(M_E^2 M_Q 2^{2S} + M_E M_Q^2 2^{2S} + M_E 2^{3S} + G_Q N_E). \tag{80}$$

9. Empirical results

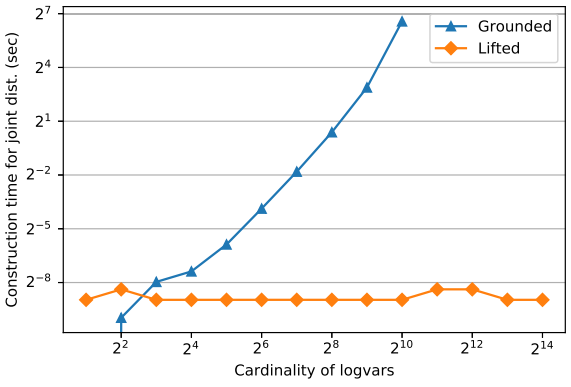
We show in this empirical study that the complexity gains translate into practice in an implementation in Python code. We support the theoretical complexity analysis in the experiments. First, we evaluate the lifted construction of the joint distribution described in Section 5. Second, we evaluate the conditional query answering. We use three different sub-graphs of our running example along with three different queries focusing on different preconditions for the query answering algorithm. Additionally, we use a fourth experiment with 21 PRVs and 5 logvars to test our approach in a more realistically sized domain. The first experiment (overlap) is using the sub-graph in the orange, solid box, focusing on the full overlap between logvar sequences in query and evidence set. The second experiment (mixed) is using the orange, solid box with the added workload of doctors (PRV $W(D)$), focusing on a query where grouping plays a big role because of balanced query and evidence set. The third example (no overlap) uses the sub-graph in the blue-box, focusing on a case where there is no overlap between any evidence and query randvars (also not along the path) but where the query-set is held constant while the evidence set is increasing. The fourth example uses a graph with a mixed overlap structure and multiple combinations of overlaps between query and evidence set. In all four experiments, we increase certain domain sizes to see the influence on the run-time in both the ground and the lifted approach. Table 2 contains the PRVs, domain sizes, and queries used in the experiments. Fig. 4 shows the results, which we discuss next.

9.1. Experimental constructions

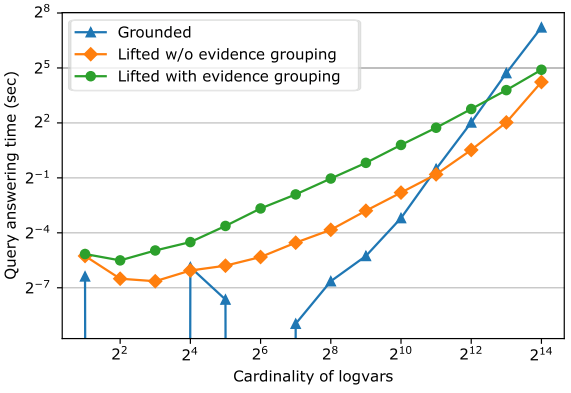
We run the lifted and grounded construction algorithms for the experiments in Table 2. As described in the theoretical analysis, the run-time should be independent of the logvar domain size (cardinality), which is validated in all experiments as can be seen in Figs. 4a to 4d. Overall, the run-time in the construction of the overlap experiment is lowest, because the number of PRVs and the number of logvars involved in the PRVs is lowest. The slight ups and downs in the lifted scenarios are due to the very low run-time (milliseconds) and comparably high effect of slight shifts in background processes.

9.2. Experimental query answering

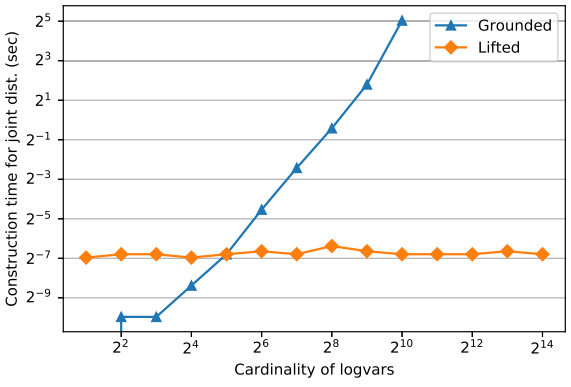
For query answering, we also use the experiments from Table 2. In all cases, we observe both lifted variants to be outperforming the grounded algorithm from a certain logvar domain size onwards (see Figs. 4d to 4g). In the non-overlapping case (Fig. 4f), where we have a constant number of queried randvars, both lifted algorithms outperform the ground algorithm significantly and show only a linear with respect to the number of evidence randvars increase as expected. When comparing the lines in Fig. 4f, we see no speed-ups between the lifted query answering using evidence grouping and the lifted query answering without evidence grouping because the linear factor of N_Q is constant and small. In the overlap experiment in Fig. 4d, where we also have an overlap between the logvars in $\boldsymbol{Q}$ and $\boldsymbol{E}$, we see that the lifting algorithm with grouping is worse than the algorithm without grouping. Because of the full overlap between $\boldsymbol{Q}$ and $\boldsymbol{E}$ all groups will have exactly one evidence entry resulting in no efficiency gains. With higher domain sizes, the difference between grouping and non-grouping gets smaller because the constant overhead of the handling the groups has less relative effect. In the mixed scenario in Fig. 4e, we have high speed-ups for the lifted algorithm with grouping compared to the lifted algorithm without grouping. This speed-up up occurs because we have no overlaps between $\boldsymbol{Q}$ and $\boldsymbol{E}$, resulting in one big group for which the evidence can be summed up, resulting in the elimination of the large N_Q factor. The results of the large experiment in Fig. 4g show that the speed-ups occur also in bigger and more realistic domains. Here, the break-even is even happening in lower logvar cardinality because due to the increased number of PRVs and logvars the network is growing faster when the cardinality increases.



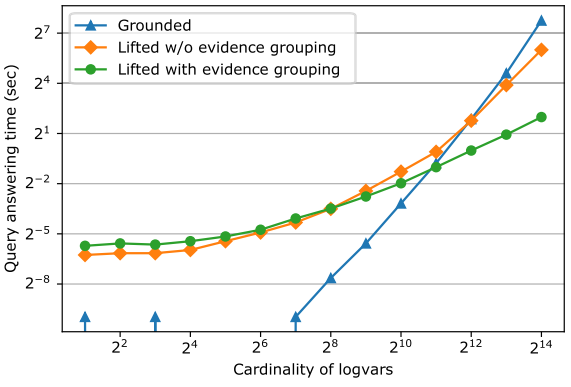
(a) Overlap construction



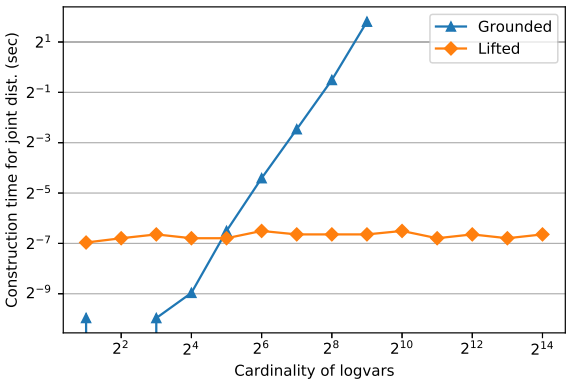
(d) Overlap query answering



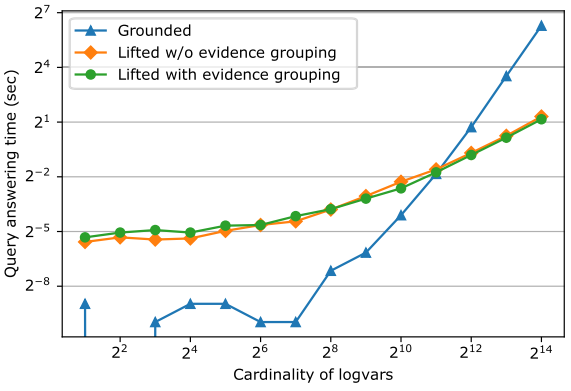
(b) Mixed construction



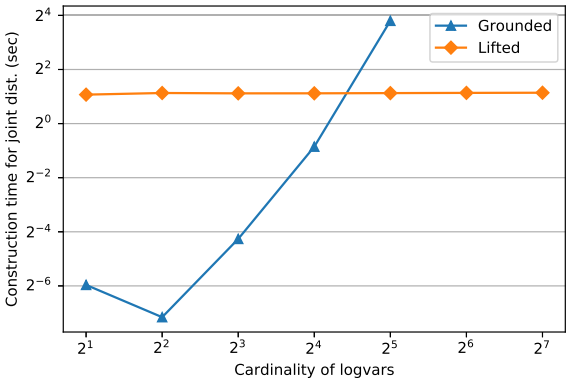
(e) Mixed query answering



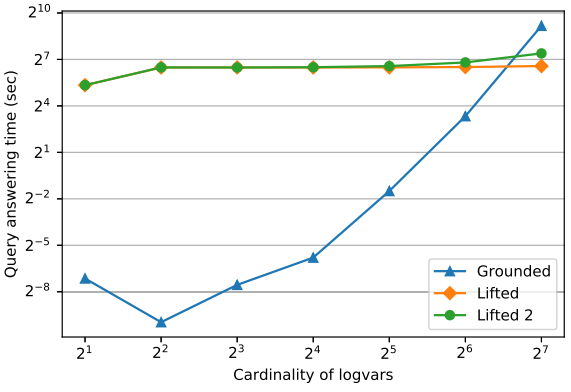
(c) No overlap construction



(f) No overlap query answering



(d) 21 PRVs, 5 logvars construction



(g) 21 PRVs, 5 logvars query answering

Fig. 4. Evaluation of experiments.

10. Conclusion and future work

We have presented an extended view on lifting GBNs that connected existing formalizations, showcased operations and applications using a running example, and introduced a new evidence handling scheme. Specifically, the article includes algorithms for constructing a lifted joint distribution for parameterized GBNs with and without overlapping logvars, connecting a non-matrix notation with a matrix notation, both approaches providing their own access to the concept of lifting GBNs. For the lifted joint in matrix notation, the article presents operations that work in a fully lifted way including addition, multiplication, and inversion. The operations provide the basis for the lifted query answering algorithms presented, including a new evidence handling approach to lift even more calculations. The new approach for evidence handling groups evidence with the same linear effects in cases of partial overlaps between the logvar sequences in the query set and the evidence set. To ensure lifted calculations, the article also introduces the notion of a liftable query that prescribes under which conditions queries can be answered in the current formalism without grounding the full lifted joint. We have shown with a theoretical complexity analysis and an experimental evaluation that we can achieve significant performance improvements compared to the ground level implementations of the algorithms.

For lifting GBNs, there are three main open issues that could trigger further research. First, we have placed some restrictions on queries for liftability. Adding mechanisms to allow for a broader class of liftable queries through combination of lifted and ground treatment would broaden the possible areas of applications. Second, for lifting in general, we assume indistinguishable randvars. An interesting question is how to treat not indistinguishable but very similar randvars. The challenge would be in understanding under which circumstances approximations can fulfill certain error bounds and thus be a feasible alternative for exact query answering. Ideas exist for understanding error bounds in discrete environments [40,82], which might provide inspiration. Third, current approaches are either lifting discrete or continuous PGMs. Understanding how to work with and subsequently, lift hybrid algorithms, e.g., the hybrid version of the junction tree algorithm [83], could be an interesting path forward.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The research has been supported by a doctoral scholarship from the “Studienstiftung des deutschen Volkes”.

Appendix A. Supplementary calculations

A.1. Ground level covariance construction

We use Eq. (8) to inductively calculate the ground covariance matrix. We start with writing out the whole sum.

$$
\begin{aligned}
\Sigma_{H(Patient1),H(Patient1)} &= 0 + \sigma^2_{H(Patient1)} = 1,\\
\Sigma_{H(Patient1),H(Patient2)} &= 0,\\
\Sigma_{H(Patient2),H(Patient2)} &= 0 + \sigma^2_{H(Patient2)} = 1,\\
\Sigma_{H(Patient1),T(Nurse1)} &= 0,\\
\Sigma_{H(Patient2),T(Nurse1)} &= 0,\\
\Sigma_{H(Nurse1),T(Nurse1)} &= 0 + \sigma^2_{H(Nurse1)} = 2,\\
\Sigma_{H(Patient1),T(Nurse2)} &= 0,\\
\Sigma_{H(Patient2),T(Nurse2)} &= 0,\\
\Sigma_{H(Nurse1),T(Nurse2)} &= 0,\\
\Sigma_{H(Nurse2),T(Nurse2)} &= 0 + \sigma^2_{H(Nurse2)} = 2,\\
\Sigma_{H(Patient1),T(Nurse3)} &= 0,\\
\Sigma_{H(Patient2),T(Nurse3)} &= 0,\\
\Sigma_{H(Nurse1),T(Nurse3)} &= 0,
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{H(Nurse2),T(Nurse3)} &= 0,\\
\Sigma_{H(Nurse3),T(Nurse3)} &= 0 + \sigma^2_{H(Nurse3)} = 2,\\
\Sigma_{H(Patient1),W(Doctor1)} &= \Sigma_{H(Patient1),H(Patient1)}\beta_{H(Patient1),W(Doctor1)}\\
&\quad + \Sigma_{H(Patient1),H(Patient2)}\beta_{H(Patient2),W(Doctor1)}\\
&= 1\cdot(-2) + 0\cdot(-2) = -2\\
\Sigma_{H(Patient2),W(Doctor1)} &= \Sigma_{H(Patient2),H(Patient1)}\beta_{H(Patient1),W(Doctor1)}\\
&\quad + \Sigma_{H(Patient2),H(Patient2)}\beta_{H(Patient2),W(Doctor1)}\\
&= 0\cdot(-2) + 1\cdot(-2) = -2\\
\Sigma_{T(Nurse1),W(Doctor1)} &= \Sigma_{T(Nurse1),H(Patient1)}\beta_{H(Patient1),W(Doctor1)}\\
&\quad + \Sigma_{T(Nurse1),H(Patient2)}\beta_{H(Patient2),W(Doctor1)}\\
&= 0\cdot(-2) + 0\cdot(-2) = 0\\
\Sigma_{T(Nurse2),W(Doctor1)} &= \Sigma_{T(Nurse2),H(Patient1)}\beta_{H(Patient1),W(Doctor1)}\\
&\quad + \Sigma_{T(Nurse2),H(Patient2)}\beta_{H(Patient2),W(Doctor1)}\\
&= 0\cdot(-2) + 0\cdot(-2) = 0\\
\Sigma_{T(Nurse3),W(Doctor1)} &= \Sigma_{T(Nurse3),H(Patient1)}\beta_{H(Patient1),W(Doctor1)}\\
&\quad + \Sigma_{T(Nurse3),H(Patient2)}\beta_{H(Patient2),W(Doctor1)}\\
&= 0\cdot(-2) + 0\cdot(-2) = 0\\
\Sigma_{W(Doctor1),W(Doctor1)} &= \Sigma_{W(Doctor1),H(Patient1)}\beta_{H(Patient1),W(Doctor1)}\\
&\quad + \Sigma_{W(Doctor1),H(Patient2)}\beta_{H(Patient2),W(Doctor1)}\\
&\quad + \sigma^2_{W(Doctor1)}\\
&= -2\cdot(-2) + -2\cdot(-2) + 3 = 11\\
\Sigma_{H(Patient1),W(Doctor2)} &= \Sigma_{H(Patient1),H(Patient1)}\beta_{H(Patient1),W(Doctor2)}\\
&\quad + \Sigma_{H(Patient1),H(Patient2)}\beta_{H(Patient2),W(Doctor2)}\\
&= 1\cdot(-2) + 0\cdot(-2) = -2\\
\Sigma_{H(Patient2),W(Doctor2)} &= \Sigma_{H(Patient2),H(Patient1)}\beta_{H(Patient1),W(Doctor2)}\\
&\quad + \Sigma_{H(Patient2),H(Patient2)}\beta_{H(Patient2),W(Doctor2)}\\
&= 0\cdot(-2) + 1\cdot(-2) = -2\\
\Sigma_{T(Nurse1),W(Doctor2)} &= \Sigma_{T(Nurse1),H(Patient1)}\beta_{H(Patient1),W(Doctor2)}\\
&\quad + \Sigma_{T(Nurse1),H(Patient2)}\beta_{H(Patient2),W(Doctor2)}\\
&= 0\cdot(-2) + 0\cdot(-2) = 0\\
\Sigma_{T(Nurse2),W(Doctor2)} &= \Sigma_{T(Nurse2),H(Patient1)}\beta_{H(Patient1),W(Doctor2)}\\
&\quad + \Sigma_{T(Nurse2),H(Patient2)}\beta_{H(Patient2),W(Doctor2)}\\
&= 0\cdot(-2) + 0\cdot(-2) = 0\\
\Sigma_{T(Nurse3),W(Doctor2)} &= \Sigma_{T(Nurse3),H(Patient1)}\beta_{H(Patient1),W(Doctor2)}\\
&\quad + \Sigma_{T(Nurse3),H(Patient2)}\beta_{H(Patient2),W(Doctor2)}\\
&= 0\cdot(-2) + 0\cdot(-2) = 0\\
\Sigma_{W(Doctor1),W(Doctor2)} &= \Sigma_{W(Doctor1),H(Patient1)}\beta_{H(Patient1),W(Doctor2)}\\
&\quad + \Sigma_{W(Doctor1),H(Patient2)}\beta_{H(Patient2),W(Doctor2)}\\
&= -2\cdot(-2) + -2\cdot(-2) = 8\\
\Sigma_{W(Doctor2),W(Doctor2)} &= \Sigma_{W(Doctor2),H(Patient1)}\beta_{H(Patient1),W(Doctor2)}
\end{aligned}
$$

$$
\begin{aligned}
&+ \Sigma_{W(Doctor2),H(Patient2)} \beta_{H(Patient2),W(Doctor2)} \\
&+ \sigma^2_{W(Doctor2)} \\
&= -2 \cdot (-2) + -2 \cdot (-2) + 3 = 11
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{H(Patient1),U(Coffee1)} &= \Sigma_{H(Patient1),T(Nurse1)} \beta_{T(Nurse1),U(Coffee1)} \\
&+ \Sigma_{H(Patient1),T(Nurse2)} \beta_{T(Nurse2),U(Coffee1)} \\
&+ \Sigma_{H(Patient1),T(Nurse3)} \beta_{T(Nurse3),U(Coffee1)} \\
&+ \Sigma_{H(Patient1),W(Doctor1)} \beta_{W(Doctor1),U(Coffee1)} \\
&+ \Sigma_{H(Patient1),W(Doctor2)} \beta_{W(Doctor2),U(Coffee1)} \\
&= 0 \cdot 2 + 0 \cdot 2 + 0 \cdot 2 + (-2) \cdot 3 + (-2) \cdot 3 = -12
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{H(Patient2),U(Coffee1)} &= \Sigma_{H(Patient2),T(Nurse1)} \beta_{T(Nurse1),U(Coffee1)} \\
&+ \Sigma_{H(Patient2),T(Nurse2)} \beta_{T(Nurse2),U(Coffee1)} \\
&+ \Sigma_{H(Patient2),T(Nurse3)} \beta_{T(Nurse3),U(Coffee1)} \\
&+ \Sigma_{H(Patient2),W(Doctor1)} \beta_{W(Doctor1),U(Coffee1)} \\
&+ \Sigma_{H(Patient2),W(Doctor2)} \beta_{W(Doctor2),U(Coffee1)} \\
&= 0 \cdot 2 + 0 \cdot 2 + 0 \cdot 2 + (-2) \cdot 3 + (-2) \cdot 3 = -12
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{T(Nurse1),U(Coffee1)} &= \Sigma_{T(Nurse1),T(Nurse1)} \beta_{T(Nurse1),U(Coffee1)} \\
&+ \Sigma_{T(Nurse1),T(Nurse2)} \beta_{T(Nurse2),U(Coffee1)} \\
&+ \Sigma_{T(Nurse1),T(Nurse3)} \beta_{T(Nurse3),U(Coffee1)} \\
&+ \Sigma_{T(Nurse1),W(Doctor1)} \beta_{W(Doctor1),U(Coffee1)} \\
&+ \Sigma_{T(Nurse1),W(Doctor2)} \beta_{W(Doctor2),U(Coffee1)} \\
&= 2 \cdot 2 + 0 \cdot 2 + 0 \cdot 2 + 0 \cdot 3 + 0 \cdot 3 = 4
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{T(Nurse2),U(Coffee1)} &= \Sigma_{T(Nurse2),T(Nurse1)} \beta_{T(Nurse1),U(Coffee1)} \\
&+ \Sigma_{T(Nurse2),T(Nurse2)} \beta_{T(Nurse2),U(Coffee1)} \\
&+ \Sigma_{T(Nurse2),T(Nurse3)} \beta_{T(Nurse3),U(Coffee1)} \\
&+ \Sigma_{T(Nurse2),W(Doctor1)} \beta_{W(Doctor1),U(Coffee1)} \\
&+ \Sigma_{T(Nurse2),W(Doctor2)} \beta_{W(Doctor2),U(Coffee1)} \\
&= 0 \cdot 2 + 2 \cdot 2 + 0 \cdot 2 + 0 \cdot 3 + 0 \cdot 3 = 4
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{T(Nurse3),U(Coffee1)} &= \Sigma_{T(Nurse3),T(Nurse1)} \beta_{T(Nurse1),U(Coffee1)} \\
&+ \Sigma_{T(Nurse3),T(Nurse2)} \beta_{T(Nurse2),U(Coffee1)} \\
&+ \Sigma_{T(Nurse3),T(Nurse3)} \beta_{T(Nurse3),U(Coffee1)} \\
&+ \Sigma_{T(Nurse3),W(Doctor1)} \beta_{W(Doctor1),U(Coffee1)} \\
&+ \Sigma_{T(Nurse3),W(Doctor2)} \beta_{W(Doctor2),U(Coffee1)} \\
&= 0 \cdot 2 + 0 \cdot 2 + 2 \cdot 2 + 0 \cdot 3 + 0 \cdot 3 = 4
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{W(Doctor1),U(Coffee1)} &= \Sigma_{W(Doctor1),T(Nurse1)} \beta_{T(Nurse1),U(Coffee1)} \\
&+ \Sigma_{W(Doctor1),T(Nurse2)} \beta_{T(Nurse2),U(Coffee1)} \\
&+ \Sigma_{W(Doctor1),T(Nurse3)} \beta_{T(Nurse3),U(Coffee1)} \\
&+ \Sigma_{W(Doctor1),W(Doctor1)} \beta_{W(Doctor1),U(Coffee1)} \\
&+ \Sigma_{W(Doctor1),W(Doctor2)} \beta_{W(Doctor2),U(Coffee1)} \\
&= 0 \cdot 2 + 0 \cdot 2 + 0 \cdot 2 + 11 \cdot 3 + 8 \cdot 3 = 57
\end{aligned}
$$

$$
\Sigma_{W(Doctor2),U(Coffee1)} = \Sigma_{W(Doctor2),T(Nurse1)} \beta_{T(Nurse1),U(Coffee1)}
$$

$$
\begin{aligned}
&+ \Sigma_{W(Doctor2),T(Nurse2)} \beta_{T(Nurse2),U(Coffee1)} \\
&+ \Sigma_{W(Doctor2),T(Nurse3)} \beta_{T(Nurse3),U(Coffee1)} \\
&+ \Sigma_{W(Doctor2),W(Doctor1)} \beta_{W(Doctor1),U(Coffee1)} \\
&+ \Sigma_{W(Doctor2),W(Doctor2)} \beta_{W(Doctor2),U(Coffee1)} \\
&= 0 \cdot 2 + 0 \cdot 2 + 0 \cdot 2 + 8 \cdot 3 + 11 \cdot 3 = 57
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{U(Coffee1),U(Coffee1)} &= \Sigma_{U(Coffee1),T(Nurse1)} \beta_{T(Nurse1),U(Coffee1)} \\
&+ \Sigma_{U(Coffee1),T(Nurse2)} \beta_{T(Nurse2),U(Coffee1)} \\
&+ \Sigma_{U(Coffee1),T(Nurse3)} \beta_{T(Nurse3),U(Coffee1)} \\
&+ \Sigma_{U(Coffee1),W(Doctor1)} \beta_{W(Doctor1),U(Coffee1)} \\
&+ \Sigma_{U(Coffee1),W(Doctor2)} \beta_{W(Doctor2),U(Coffee1)} \\
&+ \sigma^2_{U(Coffee1)} \\
&= 4 \cdot 2 + 4 \cdot 2 + 4 \cdot 2 + 57 \cdot 3 + 57 \cdot 3 + 4 = 370
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{H(Patient1),U(Coffee2)} &= \Sigma_{H(Patient1),T(Nurse1)} \beta_{T(Nurse1),U(Coffee2)} \\
&+ \Sigma_{H(Patient1),T(Nurse2)} \beta_{T(Nurse2),U(Coffee2)} \\
&+ \Sigma_{H(Patient1),T(Nurse3)} \beta_{T(Nurse3),U(Coffee2)} \\
&+ \Sigma_{H(Patient1),W(Doctor1)} \beta_{W(Doctor1),U(Coffee2)} \\
&+ \Sigma_{H(Patient1),W(Doctor2)} \beta_{W(Doctor2),U(Coffee2)} \\
&= 0 \cdot 2 + 0 \cdot 2 + 0 \cdot 2 + (-2) \cdot 3 + (-2) \cdot 3 = -12
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{H(Patient2),U(Coffee2)} &= \Sigma_{H(Patient2),T(Nurse1)} \beta_{T(Nurse1),U(Coffee2)} \\
&+ \Sigma_{H(Patient2),T(Nurse2)} \beta_{T(Nurse2),U(Coffee2)} \\
&+ \Sigma_{H(Patient2),T(Nurse3)} \beta_{T(Nurse3),U(Coffee2)} \\
&+ \Sigma_{H(Patient2),W(Doctor1)} \beta_{W(Doctor1),U(Coffee2)} \\
&+ \Sigma_{H(Patient2),W(Doctor2)} \beta_{W(Doctor2),U(Coffee2)} \\
&= 0 \cdot 2 + 0 \cdot 2 + 0 \cdot 2 + (-2) \cdot 3 + (-2) \cdot 3 = -12
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{T(Nurse1),U(Coffee2)} &= \Sigma_{T(Nurse1),T(Nurse1)} \beta_{T(Nurse1),U(Coffee2)} \\
&+ \Sigma_{T(Nurse1),T(Nurse2)} \beta_{T(Nurse2),U(Coffee2)} \\
&+ \Sigma_{T(Nurse1),T(Nurse3)} \beta_{T(Nurse3),U(Coffee2)} \\
&+ \Sigma_{T(Nurse1),W(Doctor1)} \beta_{W(Doctor1),U(Coffee2)} \\
&+ \Sigma_{T(Nurse1),W(Doctor2)} \beta_{W(Doctor2),U(Coffee2)} \\
&= 2 \cdot 2 + 0 \cdot 2 + 0 \cdot 2 + 0 \cdot 3 + 0 \cdot 3 = 4
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{T(Nurse2),U(Coffee2)} &= \Sigma_{T(Nurse2),T(Nurse1)} \beta_{T(Nurse1),U(Coffee2)} \\
&+ \Sigma_{T(Nurse2),T(Nurse2)} \beta_{T(Nurse2),U(Coffee2)} \\
&+ \Sigma_{T(Nurse2),T(Nurse3)} \beta_{T(Nurse3),U(Coffee2)} \\
&+ \Sigma_{T(Nurse2),W(Doctor1)} \beta_{W(Doctor1),U(Coffee2)} \\
&+ \Sigma_{T(Nurse2),W(Doctor2)} \beta_{W(Doctor2),U(Coffee2)} \\
&= 0 \cdot 2 + 2 \cdot 2 + 0 \cdot 2 + 0 \cdot 3 + 0 \cdot 3 = 4
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{T(Nurse3),U(Coffee2)} &= \Sigma_{T(Nurse3),T(Nurse1)} \beta_{T(Nurse1),U(Coffee2)} \\
&+ \Sigma_{T(Nurse3),T(Nurse2)} \beta_{T(Nurse2),U(Coffee2)} \\
&+ \Sigma_{T(Nurse3),T(Nurse3)} \beta_{T(Nurse3),U(Coffee2)} \\
&+ \Sigma_{T(Nurse3),W(Doctor1)} \beta_{W(Doctor1),U(Coffee2)}
\end{aligned}
$$

$$
\begin{aligned}
&+ \Sigma_{T(Nurse3),W(Doctor2)} \beta_{W(Doctor2),U(Coffee2)} \\
&= 0 \cdot 2 + 0 \cdot 2 + 2 \cdot 2 + 0 \cdot 3 + 0 \cdot 3 = 4
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{W(Doctor1),U(Coffee2)} &= \Sigma_{W(Doctor1),T(Nurse1)} \beta_{T(Nurse1),U(Coffee2)} \\
&+ \Sigma_{W(Doctor1),T(Nurse2)} \beta_{T(Nurse2),U(Coffee2)} \\
&+ \Sigma_{W(Doctor1),T(Nurse3)} \beta_{T(Nurse3),U(Coffee2)} \\
&+ \Sigma_{W(Doctor1),W(Doctor1)} \beta_{W(Doctor1),U(Coffee2)} \\
&+ \Sigma_{W(Doctor1),W(Doctor2)} \beta_{W(Doctor2),U(Coffee2)} \\
&= 0 \cdot 2 + 0 \cdot 2 + 0 \cdot 2 + 11 \cdot 3 + 8 \cdot 3 = 57
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{W(Doctor2),U(Coffee2)} &= \Sigma_{W(Doctor2),T(Nurse1)} \beta_{T(Nurse1),U(Coffee2)} \\
&+ \Sigma_{W(Doctor2),T(Nurse2)} \beta_{T(Nurse2),U(Coffee2)} \\
&+ \Sigma_{W(Doctor2),T(Nurse3)} \beta_{T(Nurse3),U(Coffee2)} \\
&+ \Sigma_{W(Doctor2),W(Doctor1)} \beta_{W(Doctor1),U(Coffee2)} \\
&+ \Sigma_{W(Doctor2),W(Doctor2)} \beta_{W(Doctor2),U(Coffee2)} \\
&= 0 \cdot 2 + 0 \cdot 2 + 0 \cdot 2 + 8 \cdot 3 + 11 \cdot 3 = 57
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{U(Coffee1),U(Coffee2)} &= \Sigma_{U(Coffee1),T(Nurse1)} \beta_{T(Nurse1),U(Coffee2)} \\
&+ \Sigma_{U(Coffee1),T(Nurse2)} \beta_{T(Nurse2),U(Coffee2)} \\
&+ \Sigma_{U(Coffee1),T(Nurse3)} \beta_{T(Nurse3),U(Coffee2)} \\
&+ \Sigma_{U(Coffee1),W(Doctor1)} \beta_{W(Doctor1),U(Coffee2)} \\
&+ \Sigma_{U(Coffee1),W(Doctor2)} \beta_{W(Doctor2),U(Coffee2)} \\
&= 4 \cdot 2 + 4 \cdot 2 + 4 \cdot 2 + 57 \cdot 3 + 57 \cdot 3 + 4 = 366
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_{U(Coffee2),U(Coffee2)} &= \Sigma_{U(Coffee1),T(Nurse1)} \beta_{T(Nurse1),U(Coffee2)} \\
&+ \Sigma_{U(Coffee1),T(Nurse2)} \beta_{T(Nurse2),U(Coffee2)} \\
&+ \Sigma_{U(Coffee1),T(Nurse3)} \beta_{T(Nurse3),U(Coffee2)} \\
&+ \Sigma_{U(Coffee1),W(Doctor1)} \beta_{W(Doctor1),U(Coffee2)} \\
&+ \Sigma_{U(Coffee1),W(Doctor2)} \beta_{W(Doctor2),U(Coffee2)} \\
&+ \sigma^2_{U(Coffee2)} \\
&= 4 \cdot 2 + 4 \cdot 2 + 4 \cdot 2 + 57 \cdot 3 + 57 \cdot 3 + 4 = 370
\end{aligned}
$$

Calculations for $U(Coffee3)$ are omitted. The resulting covariance matrix is

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & -2 & -2 & -12 & -12 & -12 \\
0 & 1 & 0 & 0 & 0 & -2 & -2 & -12 & -12 & -12 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 4 & 4 & 4 \\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 4 & 4 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 4 & 4 & 4 \\
-2 & -2 & 0 & 0 & 0 & 11 & 8 & 57 & 57 & 57 \\
-2 & -2 & 0 & 0 & 0 & 8 & 11 & 57 & 57 & 57 \\
-12 & -12 & 4 & 4 & 4 & 57 & 57 & 370 & 366 & 366 \\
-12 & -12 & 4 & 4 & 4 & 57 & 57 & 366 & 370 & 366 \\
-12 & -12 & 4 & 4 & 4 & 57 & 57 & 366 & 366 & 370
\end{bmatrix}.
$$

A.2. Lifted covariance construction - matrix notation

We use Eq. (38) to construct the covariance matrix. Factors infront of all-ones matrices and identity matrices can be stored lifted using ρ and λ.

$$
\begin{aligned}
\boldsymbol{\Sigma}_{\mathbf{gr}(H(P)),\mathbf{gr}(H(P))} &= \lambda_{H(P)}\mathbf{I}_{\tau_P\times\tau_P} = 1\mathbf{I}_{\tau_P\times\tau_P}\\
\boldsymbol{\Sigma}_{\mathbf{gr}(H(P)),\mathbf{gr}(T(N))} &= \boldsymbol{\Sigma}_{\mathbf{gr}(H(P)),\mathbf{gr}(H(P))}\boldsymbol{T}_{\mathbf{gr}(H(P)),\mathbf{gr}(T(N))} = \mathbf{I}_{\tau_P\times\tau_P}\mathbf{0}_{\tau_P\times\tau_N} = \mathbf{0}_{\tau_P\times\tau_N}\\
\boldsymbol{\Sigma}_{\mathbf{gr}(T(N)),\mathbf{gr}(T(N))} &= \boldsymbol{\Sigma}_{\mathbf{gr}(T(N)),\mathbf{gr}(H(P))}\boldsymbol{T}_{\mathbf{gr}(H(P)),\mathbf{gr}(T(N))} + \lambda_{T(N)}\mathbf{I}_{\tau_N\times\tau_N}\\
&= \mathbf{0}_{\tau_N\times\tau_P}\mathbf{0}_{\tau_N\times\tau_P} + \lambda_{T(N)}\mathbf{I}_{\tau_N\times\tau_N} = 0\mathbf{J}_{\tau_N\times\tau_N} + 2\mathbf{I}_{\tau_N\times\tau_N}\\
\boldsymbol{\Sigma}_{\mathbf{gr}(H(P)),\mathbf{gr}(W(D))} &= \boldsymbol{\Sigma}_{\mathbf{gr}(H(P)),\mathbf{gr}(H(P))}\boldsymbol{T}_{\mathbf{gr}(H(P)),\mathbf{gr}(W(D))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(H(P)),\mathbf{gr}(T(N))}\boldsymbol{T}_{\mathbf{gr}(T(N)),\mathbf{gr}(W(D))}\\
&= 1\mathbf{I}_{\tau_P\times\tau_P}(-2)\mathbf{J}_{\tau_P\times\tau_D} + \mathbf{0}_{\tau_P\times\tau_N}\mathbf{0}_{\tau_N\times\tau_D} = (-2)\mathbf{J}_{\tau_P\times\tau_D}\\
\boldsymbol{\Sigma}_{\mathbf{gr}(T(N)),\mathbf{gr}(W(D))} &= \boldsymbol{\Sigma}_{\mathbf{gr}(T(N)),\mathbf{gr}(H(P))}\boldsymbol{T}_{\mathbf{gr}(H(P)),\mathbf{gr}(W(D))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(T(N)),\mathbf{gr}(T(N))}\boldsymbol{T}_{\mathbf{gr}(T(N)),\mathbf{gr}(W(D))}\\
&= \mathbf{0}_{\tau_N\times\tau_P}(-2)\mathbf{J}_{\tau_P\times\tau_D} + 2\mathbf{I}_{\tau_N\times\tau_N}\mathbf{0}_{\tau_N\times\tau_D} = 0\mathbf{J}_{\tau_N\times\tau_D}\\
\boldsymbol{\Sigma}_{\mathbf{gr}(W(D)),\mathbf{gr}(W(D))} &= \boldsymbol{\Sigma}_{\mathbf{gr}(W(D)),\mathbf{gr}(H(P))}\boldsymbol{T}_{\mathbf{gr}(H(P)),\mathbf{gr}(W(D))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(W(D)),\mathbf{gr}(T(N))}\boldsymbol{T}_{\mathbf{gr}(T(N)),\mathbf{gr}(W(D))}\\
&\quad + \lambda_{W(D)}\mathbf{I}_{\tau_D\times\tau_D}\\
&= (-2)\mathbf{J}_{\tau_D\times\tau_P}(-2)\mathbf{J}_{\tau_P\times\tau_D} + \mathbf{0}_{\tau_D\times\tau_N}\mathbf{0}_{\tau_N\times\tau_D} + \lambda_{W(D)}\mathbf{I}_{\tau_D\times\tau_D}\\
&= (-2)\cdot(-2)\cdot\tau_P\mathbf{J}_{\tau_D\times\tau_D} + \lambda_{W(D)}\mathbf{I}_{\tau_D\times\tau_D} = 8\mathbf{J}_{\tau_D\times\tau_D} + 3\mathbf{I}_{\tau_D\times\tau_D}\\
\boldsymbol{\Sigma}_{\mathbf{gr}(H(P)),\mathbf{gr}(U(C))} &= \boldsymbol{\Sigma}_{\mathbf{gr}(H(P)),\mathbf{gr}(H(P))}\boldsymbol{T}_{\mathbf{gr}(H(P)),\mathbf{gr}(U(C))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(H(P)),\mathbf{gr}(T(N))}\boldsymbol{T}_{\mathbf{gr}(T(N)),\mathbf{gr}(U(C))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(H(P)),\mathbf{gr}(W(D))}\boldsymbol{T}_{\mathbf{gr}(W(D)),\mathbf{gr}(U(C))}\\
&= 1\mathbf{I}_{\tau_P\times\tau_P}0\mathbf{J}_{\tau_P\times\tau_C} + 0\mathbf{J}_{\tau_P\times\tau_N}2\mathbf{J}_{\tau_N\times\tau_C} + -2\mathbf{J}_{\tau_P\times\tau_D}3\mathbf{J}_{\tau_D\times\tau_C}\\
&= -2\cdot 3\cdot\tau_D\mathbf{J}_{\tau_P\times\tau_C} = -12\mathbf{J}_{\tau_P\times\tau_C}\\
\boldsymbol{\Sigma}_{\mathbf{gr}(T(N)),\mathbf{gr}(U(C))} &= \boldsymbol{\Sigma}_{\mathbf{gr}(T(N)),\mathbf{gr}(H(P))}\boldsymbol{T}_{\mathbf{gr}(H(P)),\mathbf{gr}(U(C))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(T(N)),\mathbf{gr}(T(N))}\boldsymbol{T}_{\mathbf{gr}(T(N)),\mathbf{gr}(U(C))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(T(N)),\mathbf{gr}(W(D))}\boldsymbol{T}_{\mathbf{gr}(W(D)),\mathbf{gr}(U(C))}\\
&= 0\mathbf{J}_{\tau_N\times\tau_P}0\mathbf{J}_{\tau_P\times\tau_C} + 2\mathbf{I}_{\tau_N\times\tau_N}2\mathbf{J}_{\tau_N\times\tau_C} + 0\mathbf{J}_{\tau_N\times\tau_D}3\mathbf{J}_{\tau_D\times\tau_C}\\
&= 2\cdot 2\mathbf{J}_{\tau_N\times\tau_C} = 4\mathbf{J}_{\tau_N\times\tau_C}\\
\boldsymbol{\Sigma}_{\mathbf{gr}(W(D)),\mathbf{gr}(U(C))} &= \boldsymbol{\Sigma}_{\mathbf{gr}(W(D)),\mathbf{gr}(H(P))}\boldsymbol{T}_{\mathbf{gr}(H(P)),\mathbf{gr}(U(C))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(W(D)),\mathbf{gr}(T(N))}\boldsymbol{T}_{\mathbf{gr}(T(N)),\mathbf{gr}(U(C))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(W(D)),\mathbf{gr}(W(D))}\boldsymbol{T}_{\mathbf{gr}(W(D)),\mathbf{gr}(U(C))}\\
&= -2\mathbf{J}_{\tau_D\times\tau_P}0\mathbf{J}_{\tau_P\times\tau_C} + 0\mathbf{J}_{\tau_D\times\tau_N}2\mathbf{J}_{\tau_N\times\tau_C}\\
&\quad + (8\mathbf{J}_{\tau_D\times\tau_D} + 3\mathbf{I}_{\tau_D\times\tau_D})3\mathbf{J}_{\tau_D\times\tau_C}\\
&= (8\tau_D + 3)3\cdot 2\mathbf{J}_{\tau_D\times\tau_C} = 57\mathbf{J}_{\tau_D\times\tau_C}\\
\boldsymbol{\Sigma}_{\mathbf{gr}(U(C)),\mathbf{gr}(U(C))} &= \boldsymbol{\Sigma}_{\mathbf{gr}(U(C)),\mathbf{gr}(H(P))}\boldsymbol{T}_{\mathbf{gr}(H(P)),\mathbf{gr}(U(C))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(U(C)),\mathbf{gr}(T(N))}\boldsymbol{T}_{\mathbf{gr}(T(N)),\mathbf{gr}(U(C))}\\
&\quad + \boldsymbol{\Sigma}_{\mathbf{gr}(U(C)),\mathbf{gr}(W(D))}\boldsymbol{T}_{\mathbf{gr}(W(D)),\mathbf{gr}(U(C))}\\
&\quad + \lambda_{U(C)}\mathbf{I}_{\tau_C,\tau_C}\\
&= -12\mathbf{J}_{\tau_C\times\tau_P}0\mathbf{J}_{\tau_P\times\tau_C} + 4\mathbf{J}_{\tau_C\times\tau_N}2\mathbf{J}_{\tau_N\times\tau_C}\\
&\quad + (57\mathbf{J}_{\tau_C\times\tau_D})3\mathbf{J}_{\tau_D\times\tau_C} + \lambda_{U(C)}\mathbf{I}_{\tau_C,\tau_C}\\
&= (4\cdot 2\tau_N)\mathbf{J}_{\tau_C\times\tau_C} + (57\cdot 3\tau_D)\mathbf{J}_{\tau_C\times\tau_C} = 366\mathbf{J}_{\tau_C\times\tau_C} + 4\mathbf{I}_{\tau_C,\tau_C}
\end{aligned}
$$

A.3. Full lifted joint covariance

Here, we list all values of the lifted representation of the covariance matrix for the full example visualized in Fig. 2.

$$\boldsymbol{\rho}_{1,1} = \begin{pmatrix} \rho_{1,1}^{0_M} \\ \rho_{1,1}^{1_M} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{1,2} = \begin{pmatrix} \rho_{1,2}^{1_M 1_P} \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{2,2} = \begin{pmatrix} \rho_{2,2}^{0_P} \\ \rho_{2,2}^{1_P} \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{1,3} = \begin{pmatrix} \rho_{1,3}^{0_M 1_P} \\ \rho_{1,3}^{1_M 1_P} \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{2,3} = \begin{pmatrix} \rho_{2,3}^{1_M 0_P} \\ \rho_{2,3}^{1_M 1_P} \end{pmatrix} = \begin{pmatrix} 10 \\ 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{3,3} = \begin{pmatrix} \rho_{3,3}^{0_M 0_P} \\ \rho_{3,3}^{0_M 1_P} \\ \rho_{3,3}^{1_M 0_P} \\ \rho_{3,3}^{1_M 1_P} \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 50 \\ 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{1,4} = \begin{pmatrix} \rho_{1,4}^{1_M 1_P} \end{pmatrix} = \begin{pmatrix} 8 \end{pmatrix}$$

$$\boldsymbol{\rho}_{2,4} = \begin{pmatrix} \rho_{2,4}^{0_P} \\ \rho_{2,4}^{1_P} \end{pmatrix} = \begin{pmatrix} 114 \\ 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{3,4} = \begin{pmatrix} \rho_{3,4}^{1_M 0_P} \\ \rho_{3,4}^{1_M 1_P} \end{pmatrix} = \begin{pmatrix} 582 \\ 16 \end{pmatrix}$$

$$\boldsymbol{\rho}_{4,4} = \begin{pmatrix} \rho_{4,4}^{0_P} \\ \rho_{4,4}^{1_P} \end{pmatrix} = \begin{pmatrix} 6642 \\ 192 \end{pmatrix}$$

$$\boldsymbol{\rho}_{1,5} = \begin{pmatrix} \rho_{1,5}^{1_M 1_T} \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{2,5} = \begin{pmatrix} \rho_{2,5}^{1_P 1_T} \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{3,5} = \begin{pmatrix} \rho_{3,5}^{1_M 1_P 1_T} \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{4,5} = \begin{pmatrix} \rho_{4,5}^{1_P 1_T} \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{5,5} = \begin{pmatrix} \rho_{5,5}^{0_T} \\ \rho_{5,5}^{1_T} \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{1,6} = \begin{pmatrix} \rho_{1,6}^{1_M 1_D} \end{pmatrix} = \begin{pmatrix} -32 \end{pmatrix}$$

$$\boldsymbol{\rho}_{2,6} = \begin{pmatrix} \rho_{2,6}^{1_P 1_D} \end{pmatrix} = \begin{pmatrix} -228 \end{pmatrix}$$

$$\boldsymbol{\rho}_{3,6} = \begin{pmatrix} \rho_{3,6}^{1_M 1_P 1_D} \end{pmatrix} = \begin{pmatrix} -1228 \end{pmatrix}$$

$$\boldsymbol{\rho}_{4,6} = \begin{pmatrix} \rho_{4,6}^{1_P 1_D} \end{pmatrix} = \begin{pmatrix} -14054 \end{pmatrix}$$

$$\boldsymbol{\rho}_{5,6} = \begin{pmatrix} \rho_{5,6}^{1_T 1_D} \end{pmatrix} = \begin{pmatrix} 0 \end{pmatrix}$$

$$\boldsymbol{\rho}_{6,6} = \begin{pmatrix} \rho_{6,6}^{0_D} \\ \rho_{6,6}^{1_D} \end{pmatrix} = \begin{pmatrix} 3 \\ 56216 \end{pmatrix}$$

$$\boldsymbol{\rho}_{1,7} = \begin{pmatrix} \rho_{1,7}^{1_M 1_C} \end{pmatrix} = \begin{pmatrix} -192 \end{pmatrix}$$

$$\boldsymbol{\rho}_{2,7} = \begin{pmatrix} \rho_{2,7}^{1_P 1_C} \end{pmatrix} = \begin{pmatrix} -1368 \end{pmatrix}$$

$$\boldsymbol{\rho}_{3,7} = \left(\rho_{3,7}^{1_M 1_P 1_C} \right) = \left(-7368 \right)$$
$$\boldsymbol{\rho}_{4,7} = \left(\rho_{4,7}^{1_P 1_C} \right) = \left(-84324 \right)$$
$$\boldsymbol{\rho}_{5,7} = \left(\rho_{5,7}^{1_T 1_C} \right) = \left(4 \right)$$
$$\boldsymbol{\rho}_{6,7} = \left(\rho_{6,7}^{1_T 1_D} \right) = \left(337305 \right)$$
$$\boldsymbol{\rho}_{7,7} = \begin{pmatrix} \rho_{7,7}^{0_C} \\ \rho_{7,7}^{1_C} \end{pmatrix} = \begin{pmatrix} 4 \\ 2023854 \end{pmatrix}$$

References

[1] P. Larrañaga, S. Moral, Probabilistic graphical models in artificial intelligence, Appl. Soft Comput. 11 (2) (2011) 1511–1528, https://doi.org/10.1016/j.asoc.2008.01.003, the Impact of Soft Computing for the Progress of Artificial Intelligence, https://www.sciencedirect.com/science/article/pii/S1568494608000094.
[2] D. Poole, First-order probabilistic inference, in: IJCAI, vol. 3, 2003, pp. 985–991.
[3] A. Kimmig, L. Mihalkova, L. Getoor, Lifted graphical models: a survey, Mach. Learn. 99 (1) (2015), https://doi.org/10.1007/s10994-014-5443-2.
[4] V. Sharma, N.A. Sheikh, H. Mittal, V. Gogate, P. Singla, Lifted marginal MAP inference, arXiv preprint, arXiv:1807.00589, 2018.
[5] S. Holtzen, T. Millstein, G.V. den Broeck, Generating and sampling orbits for lifted probabilistic inference, arXiv preprint, arXiv:1903.04672, 2019.
[6] M. Hartwig, R. Möller, Lifted query answering in Gaussian Bayesian networks, in: Proceedings of Machine Learning Research, 2020, pp. 233–244.
[7] M. Hartwig, T. Braun, R. Möller, Handling overlaps when lifting Gaussian Bayesian networks, in: IJCAI, 2021, pp. 4980–4986.
[8] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.
[9] R.D. Shachter, C.R. Kenley, Gaussian influence diagrams, Manag. Sci. 35 (5) (1989) 527–550, https://doi.org/10.1287/mnsc.35.5.527.
[10] M. Richardson, P. Domingos, Markov logic networks, Mach. Learn. 62 (1–2) (2006) 107–136.
[11] M. Jaeger, Relational Bayesian networks, in: UAI-97 Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1997, pp. 191–199.
[12] L. De Raedt, A. Kimmig, H. Toivonen, ProbLog: a probabilistic prolog and its application in link discovery, in: IJCAI-07 Proceedings of 20th International Joint Conference on Artificial Intelligence, IJCAI Organization, 2007, pp. 2062–2467.
[13] R. de Salvo Braz, E. Amir, D. Roth, Lifted first-order probabilistic inference, in: IJCAI-05 Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI Organization, 2005, pp. 1319–1325.
[14] R. de Salvo Braz, E. Amir, D. Roth, MPE and partial inversion in lifted probabilistic variable elimination, in: AAAI-06 Proceedings of the 21st AAAI Conference on Artificial Intelligence, AAAI Press, 2006, pp. 1123–1130.
[15] B. Milch, L.S. Zettelmoyer, K. Kersting, M. Haimes, L.P. Kaelbling, Lifted probabilistic inference with counting formulas, in: AAAI-08 Proceedings of the 23rd AAAI Conference on Artificial Intelligence, AAAI Press, 2008, pp. 1062–1068.
[16] N. Taghipour, J. Davis, H. Blockeel, Generalised counting for lifted variable elimination, in: ILP-13 Proceedings of the International Conference on Inductive Logic Programming, Springer, 2013, pp. 107–122.
[17] U. Apsel, R.I. Brafman, Extended lifted inference with joint formulas, in: UAI-11 Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2011, pp. 74–83.
[18] N. Taghipour, D. Fierens, J. Davis, H. Blockeel, Lifted variable elimination: decoupling the operators from the constraint language, J. Artif. Intell. Res. 47 (1) (2013) 393–439.
[19] T. Braun, R. Möller, Parameterised queries and lifted query answering, in: IJCAI-18 Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI Organization, 2018, pp. 4980–4986.
[20] S. Holtzen, T. Millstein, G. Van den Broeck, Generating and sampling orbits for lifted probabilistic inference, in: UAI-19 Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2019, pp. 1–10.
[21] T. Braun, R. Möller, Lifted junction tree algorithm, in: Proceedings of KI 2016: Advances in Artificial Intelligence, Springer, 2016, pp. 30–42.
[22] T. Braun, R. Möller, Preventing groundings and handling evidence in the lifted junction tree algorithm, in: Proceedings of KI 2017: Advances in Artificial Intelligence, Springer, 2017, pp. 85–98.
[23] M. Hoffmann, T. Braun, R. Möller, Lifted division for lifted Hugin belief propagation, in: AISTATS-22 Proceedings of the 25th International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 3216–3224.
[24] S.L. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, J. R. Stat. Soc. B 50 (1988) 157–224.
[25] G.R. Shafer, P.P. Shenoy, Probability propagation, Ann. Math. Artif. Intell. 2 (1) (1990) 327–351.
[26] F.V. Jensen, S.L. Lauritzen, K.G. Olesen, Bayesian updating in recursive graphical models by local computations, Comput. Stat. Q. 4 (1990) 269–282.
[27] G. Van den Broeck, N. Taghipour, W. Meert, J. Davis, L. De Raedt, Lifted probabilistic inference by first-order knowledge compilation, in: IJCAI-11 Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI Organization, 2011, pp. 2178–2185.
[28] G. Van den Broeck, J. Davis, Conditioning in first-order knowledge compilation and lifted probabilistic inference, in: AAAI-12 Proceedings of the 26th AAAI Conference on Artificial Intelligence, AAAI Press, 2012, pp. 1961–1967.
[29] A. Darwiche, P. Marquis, A knowledge compilation map, J. Artif. Intell. Res. 17 (1) (2002) 229–264.
[30] A. Jha, V. Gogate, A. Meliou, D. Suciu, Lifted inference seen from the other side: the tractable features, in: NIPS-10 Advances in Neural Information Processing Systems 23, Curran Associates, Inc., 2010, pp. 973–981.
[31] V. Gogate, P. Domingos, Probabilistic theorem proving, in: UAI-11 Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2011, pp. 256–265.
[32] R. de Salvo Braz, S. Natarajan, H. Bui, J. Shavlik, S. Russell, Anytime lifted belief propagation, in: 6th International Workshop on Statistical Relational Learning, 2009, pp. 1–3.
[33] P. Singla, A. Nath, P. Domingos, Approximate lifted belief propagation, in: Proceedings of the 6th AAAI Conference on Statistical Relational Artificial Intelligence, AAAI Press, 2010, pp. 92–97.
[34] B. Ahmadi, K. Kersting, M. Mladenov, S. Natarajan, Exploiting symmetries for scaling loopy belief propagation and relational training, Mach. Learn. 92 (1) (2013) 91–132.
[35] P. Singla, A. Nath, P. Domingos, Approximate lifting techniques for belief propagation, in: AAAI-14 Proceedings of the 28th AAAI Conference on Artificial Intelligence, AAAI Press, 2014, pp. 2497–2504.
[36] V. Gogate, A. Jha, D. Venugopal, Advances in lifted importance sampling, in: AAAI-12 Proceedings of the 26th AAAI Conference on Artificial Intelligence, AAAI Press, 2012, pp. 1910–1916.

[37] D. Venugopal, V. Gogate, On lifting the Gibbs sampling algorithm, in: NIPS-12 Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1655–1663.
[38] D. Venugopal, S. Sarkhel, K. Cherry, Non-parametric domain approximation for scalable Gibbs sampling in MLNs, in: UAI-16 Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2016, pp. 745–754.
[39] M. Niepert, Markov chains on orbits of permutation groups, in: UAI-12 Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2012, pp. 624–633.
[40] G. Van den Broeck, M. Niepert, Lifted probabilistic inference for asymmetric graphical models, in: AAAI-15 Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI Press, 2015, pp. 3599–3605.
[41] P. Sen, A. Deshpande, L. Getoor, Bisimulation-based approximate lifted inference, in: UAI-09 Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2009, pp. 496–505.
[42] J. Choi, E. Amir, Lifted relational variational inference, in: UAI-12 Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2012, pp. 196–206.
[43] H.H. Bui, T.N. Huynh, D. Sontag, Lifted tree-reweighted variational inference, in: UAI-14 Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2014, pp. 92–101.
[44] U. Apsel, R.I. Brafman, Exploiting uniform assignments in first-order MPE, in: UAI-12 Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2012, pp. 74–83.
[45] T. Braun, R. Möller, Lifted most probable explanation, in: Proceedings of the International Conference on Conceptual Structures, Springer, 2018, pp. 39–54.
[46] T. Braun, Rescued from a Sea of Queries: Exact Inference in Probabilistic Relational Models, Ph.D. thesis, University of Lübeck, 2020.
[47] S. Sarkhel, D. Venugopal, P. Singla, V. Gogate, Lifted MAP inference for Markov logic networks, in: AISTATS-14 Proceedings of the 17th International Conference on Artificial Intelligence and Statistics, AAAI Press, 2014, pp. 859–867.
[48] S. Sarkhel, P. Singla, V. Gogate, Fast lifted MAP inference via partitioning, in: Advances in Neural Information Processing Systems 28 (NeurIPS 2015), Curran Associates, Inc., 2015, pp. 1–9.
[49] M. Mladenov, K. Kersting, A. Globerson, Efficient lifting of MAP LP relaxations using k-locality, in: AISTATS-14 Proceedings of the 17th International Conference on Artificial Intelligence and Statistics, PMLR, 2014, pp. 623–632.
[50] V. Sharma, N.A. Sheikh, H. Mittal, V. Gogate, P. Singla, Lifted marginal MAP inference, in: UAI-18 Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2018, pp. 917–926.
[51] M. Gehrke, T. Braun, R. Möller, Lifted dynamic junction tree algorithm, in: International Conference on Conceptual Structures, Springer, 2018, pp. 55–69.
[52] M. Gehrke, T. Braun, R. Möller, Relational forward backward algorithm for multiple queries, in: FLAIRS-19 Proceedings of the 32nd International Florida Artificial Intelligence Research Society Conference, AAAI Press, 2019.
[53] K.P. Murphy, Dynamic Bayesian Networks: Representation, Inference and Learning, Ph.D. thesis, University of California, Berkeley, 2002.
[54] J. Vlasselaer, W. Meert, G. Van den Broeck, L. De Raedt, Exploiting local and repeated structure in dynamic Bayesian networks, Artif. Intell. 232 (2016) 43–53.
[55] M. Gehrke, T. Braun, R. Möller, Lifted temporal most probable explanation, in: Proceedings of the International Conference on Conceptual Structures 2019, Springer, 2019.
[56] M. Niepert, G. Van den Broeck, Tractability through exchangeability: a new perspective on efficient probabilistic inference, in: AAAI-14 Proceedings of the 28th AAAI Conference on Artificial Intelligence, AAAI Press, 2014, pp. 2467–2475.
[57] N. Taghipour, D. Fierens, G. Van den Broeck, J. Davis, H. Blockeel, Completeness results for lifted variable elimination, in: AISTATS-13 Proceedings of the 16th International Conference on Artificial Intelligence and Statistics, AAAI Press, 2013, pp. 572–580.
[58] G. Van den Broeck, On the completeness of first-order knowledge compilation for lifted probabilistic inference, in: NIPS-11 Advances in Neural Information Processing Systems 24, Curran Associates, Inc., 2011, pp. 1386–1394.
[59] Y. Wang, T. van Bremen, Y. Wang, O. Kuželka, Domain-lifted sampling for universal two-variable logic and extensions, in: AAAI-22 Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI Press, 2022, pp. 10070–10079.
[60] S.M. Kazemi, A. Kimmig, G. Van den Broeck, D. Poole, Domain recursion for lifted inference with existential quantifiers, in: 7th International Workshop on Statistical Relational AI at the 33rd Conference on Uncertainty in Artificial Intelligence, 2017, pp. 1–6.
[61] M. Gehrke, Taming Exact Inference in Temporal Probabilistic Relational Models, Ph.D. thesis, University of Lübeck, 2021.
[62] M. Gehrke, T. Braun, R. Möller, A. Waschkau, C. Strumann, J. Steinhäuser, Lifted maximum expected utility, in: Artificial Intelligence in Health, Springer, 2019, pp. 131–141.
[63] M. Gehrke, T. Braun, R. Möller, Lifted temporal maximum expected utility, in: Proceedings of the 32nd Canadian Conference on Artificial Intelligence, Canadian AI 2019, Springer, 2019.
[64] C. Boutilier, R. Reiter, B. Price, Symbolic dynamic programming for first-order MDPs, in: IJCAI-01 Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI Organization, 2001, pp. 690–697.
[65] J. McCarthy, Situations, Actions, and Causal Laws, Tech. rep., Standford University, 1963.
[66] S. Sanner, K. Kersting, Symbolic dynamic programming for first-order POMDPs, in: AAAI-10 Proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI Press, 2010, pp. 1140–1146.
[67] T. Braun, M. Gehrke, F. Lau, R. Möller, Lifting in multi-agent systems under uncertainty, in: UAI-22 Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2022, pp. 1–8.
[68] J. Choi, E. Amir, D.J. Hill, Lifted inference for relational continuous models, in: UAI-10 Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2010, pp. 13–18.
[69] S.H. Bach, M. Broecheler, B. Huang, L. Getoor, Hinge-loss Markov random fields and probabilistic soft logic, J. Mach. Learn. Res. 18 (2017) 1–67.
[70] G. Van den Broeck, A. Darwiche, On the complexity and approximation of binary evidence in lifted inference, in: NIPS-13 Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 2868–2876.
[71] H.H. Bui, T.N. Huyhn, R. de Salvo Braz, Exact lifted inference with distinct soft evidence on every object, in: AAAI-12 Proceedings of the 26th AAAI Conference on Artificial Intelligence, AAAI Press, 2012, pp. 1875–1881.
[72] M. Gehrke, T. Braun, R. Möller, Uncertain evidence for probabilistic relational models, in: Proceedings of the 32nd Canadian Conference on Artificial Intelligence, Canadian AI 2019, Springer, 2019.
[73] B. Ahmadi, K. Kersting, S. Sanner, Multi-evidence lifted message passing with application to PageRank and the Kalman filter, in: IJCAI-11 Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI Organization, 2011, pp. 1152–1158.
[74] D. Venugopal, V. Gogate, Evidence-based clustering for scalable inference in Markov logic, in: ECML PKDD 2014: Machine Learning and Knowledge Discovery in Databases, Springer, 2014, pp. 258–273.
[75] M. Gehrke, T. Braun, R. Möller, Taming reasoning in temporal probabilistic relational models, in: ECAI-20 Proceedings of the 24th European Conference on Artificial Intelligence, 2020.
[76] N. Timm, Applied Multivariate Analysis, Springer Texts in Statistics, Springer, New York, 2007, https://books.google.de/books?id=vtiyg6fnnskC.
[77] B.J. Broxson, The Kronecker product, Master's thesis, University of North Florida, 2006.
[78] S.R. Searle, H.Y. Henderson, Dispersion matrices for variance components models, J. Am. Stat. Assoc. 74 (366a) (1979) 465–470.
[79] D.S. Bernstein, Matrix Mathematics: Theory, Facts, and Formulas, Princeton University Press, 2009.

[80] A.M. Davie, A.J. Stothers, Improved bound for complexity of matrix multiplication, Proc. R. Soc. Edinb., Sect. A, Math. 143 (2) (2013) 351–369.
[81] F. Le Gall, Powers of tensors and fast matrix multiplication, in: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation, 2014, pp. 296–303.
[82] M. Gehrke, R. Möller, T. Braun, Taming reasoning in temporal probabilistic relational models, arXiv preprint, arXiv:1911.07040, 2019.
[83] S.L. Lauritzen, F. Jensen, Stable local computation with conditional Gaussian distributions, Stat. Comput. 11 (2) (2001) 191–203, https://doi.org/10.1023/A:1008935617754.