UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME

Statistical Relational AI

Exploiting Symmetries

Tanya Braun, University of Münster
Marcel Gehrke, University of Lübeck
Marco Wilhelm, TU Dortmund University

Marcel Gehrke

Agenda

1. Introduction [Tanya]
2. **Exploiting Symmetries in Probabilistic Graphical Models [Marcel]**
    - Identifying Most Likely Sources of Events
    - Introducing Time
    - Approximating Symmetries over Time to Keep Reasoning Polynomial
3. Exploiting Symmetries in Conditional Knowledge Bases [Marco]
4. Summary [Tanya]

Who did it? – Identifying the Most Likely Sources of Events

Using Symmetries in Inference for a new Query Type

UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME
Marcel Gehrke

Who did it? – Identifying the Most Likely Sources of Events

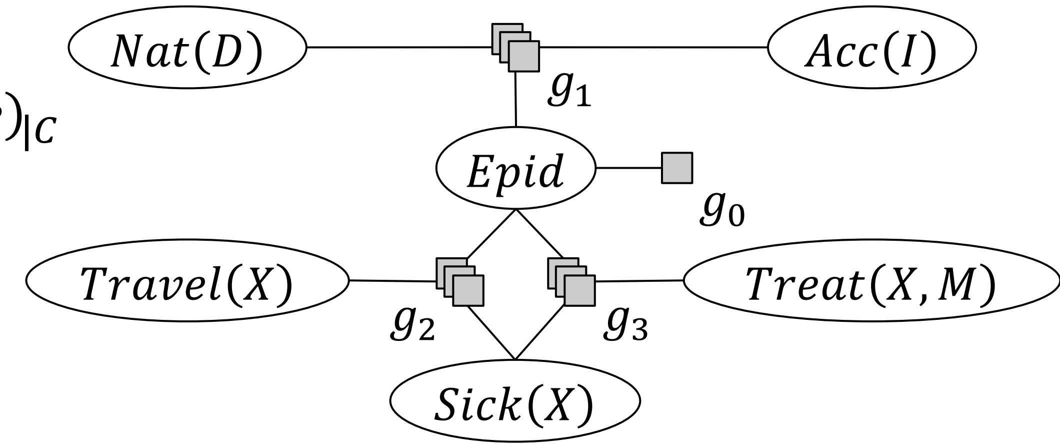
- Under-specified evidence, i.e., set of observations, without a known source: $R(X) = r$
  - $Sick(X') = true, |dom(X')| = 1000$
  - NB: Not possible in propositional setting as an observation can only belong to a specific random variable without additional information about relations or types
- Optimisation problem for a single logical variable in the evidence:
  Given evidence $\boldsymbol{e}$ with known source, find a domain C for X such that the probability of the evidence without a source is maximal under the domain, written as

$$\arg\max_{C} P(R(X) = r|\boldsymbol{e})_{|C}$$

  - Use C as source for evidence
  - Example from above: If no further evidence and only one group of indistinguishable instances represented by X in model, then any 1000 instances represented by X will do

Marcel Gehrke, Ralf Möller, and Tanya Braun. Who did it? Identifying the Most Likely Origins of Events. In *PGM-22 Proceedings of the 11th* International Conference on Probabilistic Graphical Models, 2022.

Who did it? – Combinatorial Problem

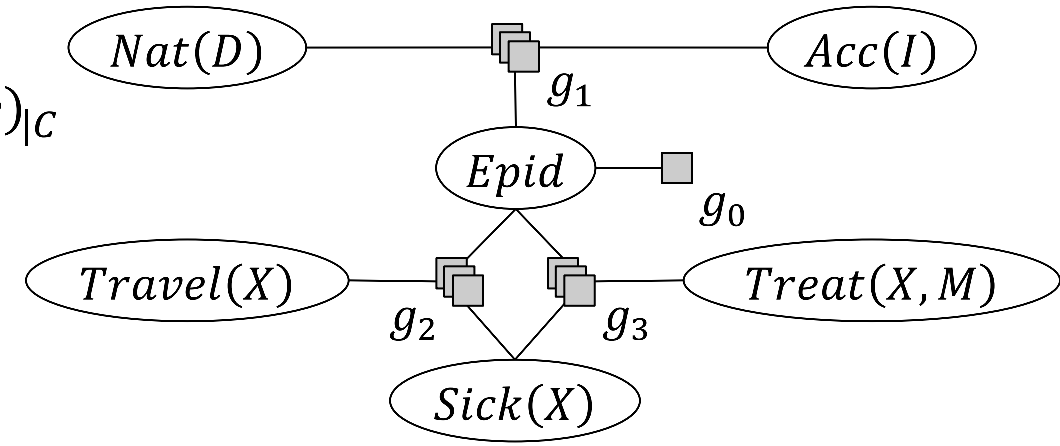


- Optimisation problem:

$$\arg\max_{C} P(R(X) = r|\boldsymbol{e})_{|C}$$

- What if we have under-specified evidence for two different PRVs
  - $Travel(X) = true, |dom(X)| = 10$
  - $Sick(Y) = true, |dom(Y)| = 10$
    - The same 10 constants exhibit both observations at the same time (overlapping) and
    - that 10 constants exhibit $DoR(X) = true$ and 10 other constants exhibit $Rep(X) = true$ (disjoint)
  - Assuming we could identify the corresponding propositional random variables and $|\mathcal{D}(X)| = 100$ in our model
    - $\binom{100}{10} \approx 3 \cdot 10^{26}$ possibilities

Who did it? – Adding Constraint Sets

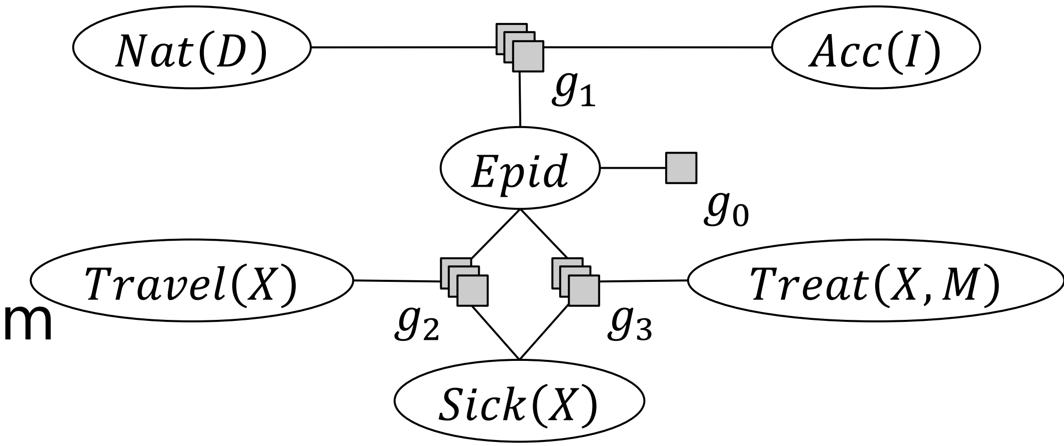


- Optimisation problem:

$$\arg\max_{C} P(R(X) = r \mid \boldsymbol{e})_{|C}$$

- However, it gets complicated once you have more sets of unknown sources or sets of known sources to consider as well
  - $Travel(X) = true, |dom(X)| = 1000$
  - $Treat(Y, m1) = true, |dom(Y)| = 500$
  - $Sick(Z) = true, dom(Z) = \{x_1, \dots, x_{100}\}$
  - Various domain assignments possible from full overlap to complete disjoint sets
    - $dom(Z) = dom(Y') = dom(X') = \{x_1, \dots, x_{100}\}$, $dom(Y'') = dom(X'') = \{x_{101}, \dots, x_{500}\}$, $dom(X''') = \{x_{501}, \dots, x_{1000}\}$
    - …
    - $dom(Z) = \{x_1, \dots, x_{100}\}$ $dom(Y) = \{x_{101}, \dots, x_{600}\}$ $dom(X) =$ $\{x_{601}, \dots, x_{1600}\}$

Who did it? – Adding even more Constraint Sets



- Number of (distinguishable) events (from known and unknown sources) influence the number possibilities
  - # of balls and urns of combinatorial problem depend on # of distinguishable events
  - Remains a combinatorial problem even with indistinguishability
- In practice number of different events limited
  - Groups of indistinguishable individuals assumed to behave approximately the same
  - No contradicting events for one constant
  - Computable in many practical cases

Interim Summary

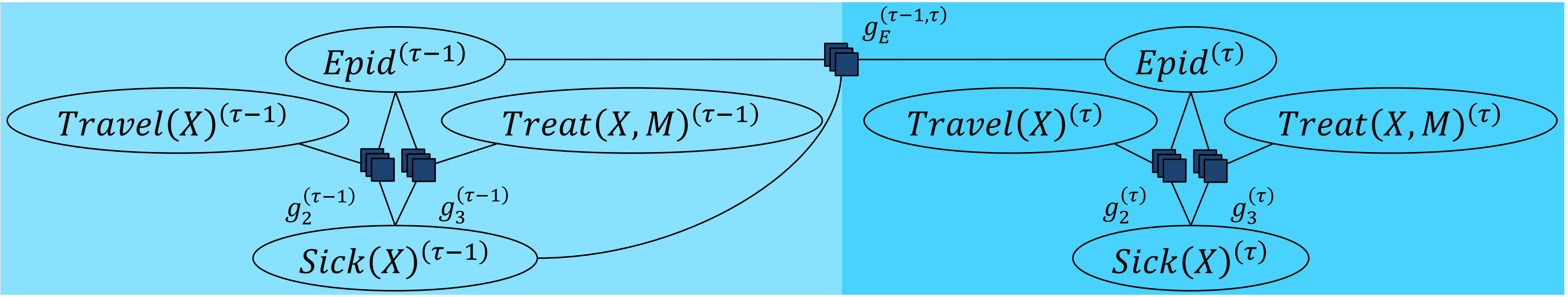
- New type of query based on logical variables
- Identifying the most likely sources of events is a combinatorial problem
  - Having indistinguishability allows us to tackle the combinatorial problem (fewer urns and balls to consider)
  - With a propositional representation not applicable
  - Without indistinguishability already for small domain sizes too many possibilities

Temporal Lifted Inference

Using Symmetries in Temporal Inference

Dynamic Parfactor-based Model

- Marginal distribution query: $P(A_\pi^i \mid E_{0:t})$ w.r.t. the model:
  - Hindsight: $\pi < t$ (Was there an epidemic $t - \pi$ days ago?)
  - Filtering: $\pi = t$ (Is there currently an epidemic?)
  - Prediction: $\pi > t$ (Is there an epidemic in $\pi - t$ days?)
- MPE, MAP on temporal sequence



2-slice Model & 1.5-slice Model

Partial specification of a sequential model (without $P(\boldsymbol{R}^{(0)})$)

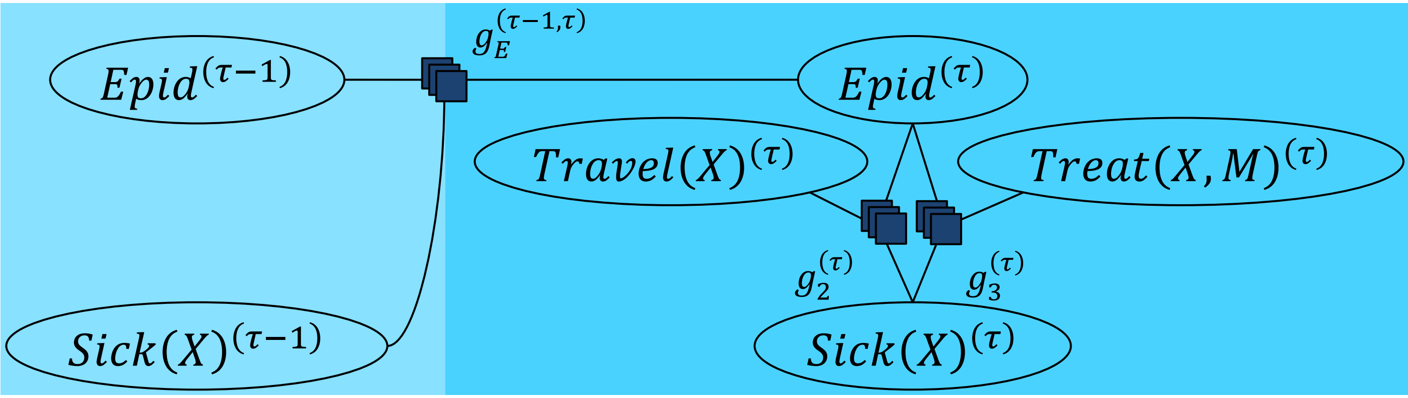
- 2-slice model (in effect only a 1.5-slice model)
  - Specification of a transition model $P(\boldsymbol{R}^{(\tau)} | \boldsymbol{R}^{(\tau-1)})$ based on a factorisation
    - Template model that gets instantiated by replacing τ with a (time) step t
    - Sufficient: Intra-slice description for τ, Inter-slice description from $\tau - 1$ to τ
  - For parfactor-based models: $G^{\rightarrow} = \left\{g_i^{(\tau)}\right\}_{i=1}^{n} \cup \left\{g_j^{(\tau-1,\tau)}\right\}_{j=1}^{m}$ with
    - Intra-slice parfactors

$$g_i^{(\tau)} = \phi_i^{(\tau)}\left(A_1^{(\tau)}, \dots, A_{k_i}^{(\tau)}\right)_{|C_i^{(\tau)}}$$

    - Inter-slice parfactors

$$g_j^{(\tau-1,\tau)} = \phi_j^{(\tau-1,\tau)}\left(A_1^{(\nu)}, \dots, A_{k_j}^{(\nu)}\right)_{|C_j^{(\tau-1,\tau)}}$$

      - $\nu \in \{\tau, \tau - 1\}$

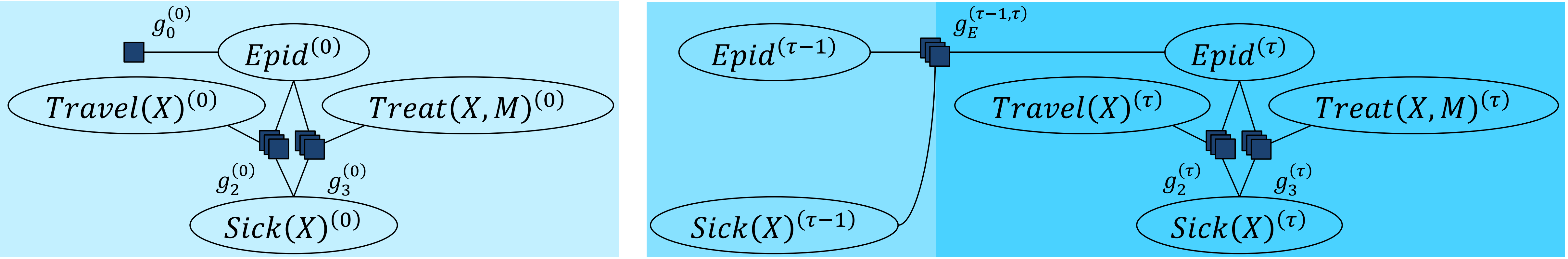


Dynamic Parfactor-based Model

- Dynamic parfactor-based model $(G^0, G^{\rightarrow})$ with
  - $G^0 = \left\{g_i^{(0)}\right\}_{i=1}^{n_0}, g_i^{(0)} = \phi_i^{(0)}\left(\mathcal{A}_1^{(0)}, \dots, \mathcal{A}_{k_i}^{(0)}\right)$
  - $G^{\rightarrow}$ a 2-slice model, i.e., $G^{\rightarrow} = \left\{g_i^{(\tau)}\right\}_{i=1}^{n} \cup \left\{g_j^{(\tau-1,\tau)}\right\}_{j=1}^{m}$
  - Example: $G^0 = \left\{g_2^{(0)}, g_3^{(0)}\right\} \cup \left\{g_0^{(0)}\right\}$, $G^{\rightarrow} = \left\{g_2^{(\tau)}, g_3^{(\tau)}\right\} \cup \left\{g_E^{(\tau-1,\tau)}\right\}$
    - No assumptions about observability

G^0 often consists of intra-slice parfactors of $G^{\rightarrow}$ with $\tau = 0$, possibly extended by pseudo prior distributions, i.e.,

$$\left\{g_i^{(0)}\right\}_{i=1}^{n_0} = \left\{g_i^{(\tau)|\tau=0}\right\}_{i=1}^{n} \cup \left\{g_i^{(0)}\right\}_{i=1}^{n'}$$



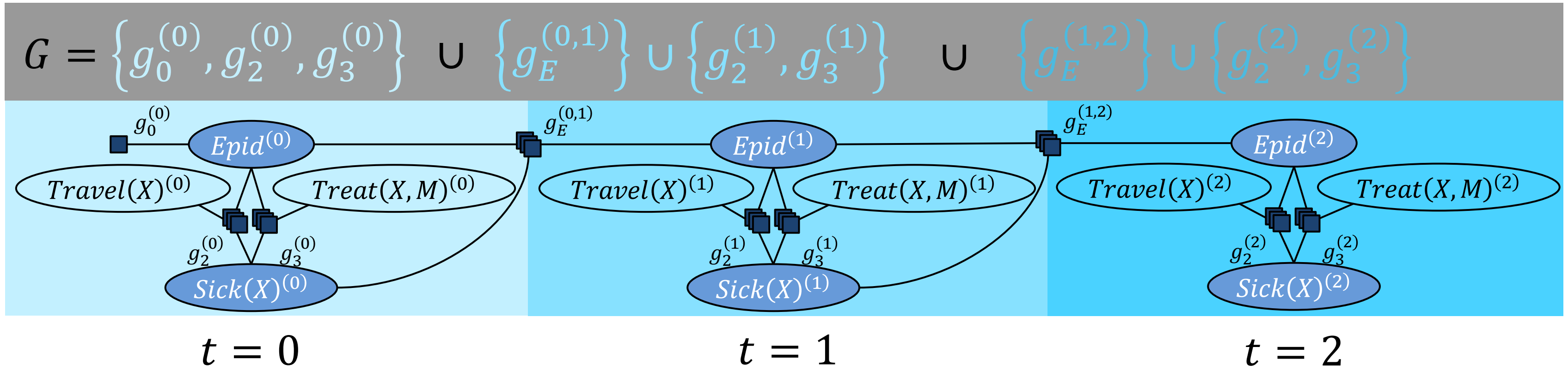
Sequential Models

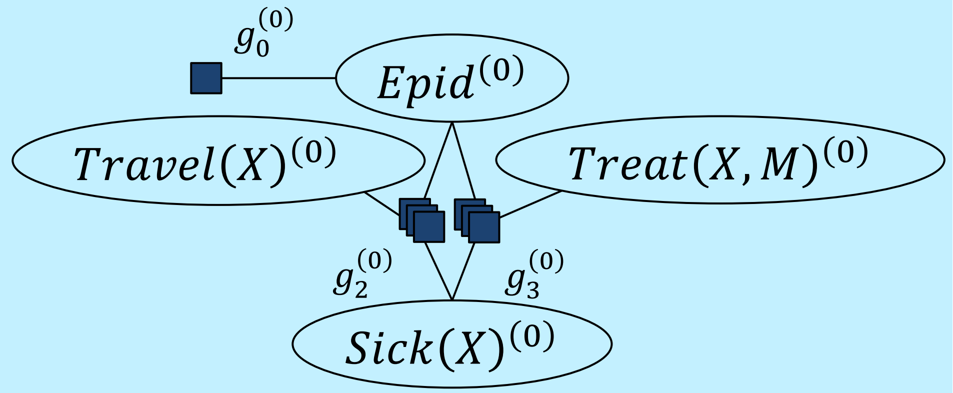
- Assumptions: Markov-1, stationary process

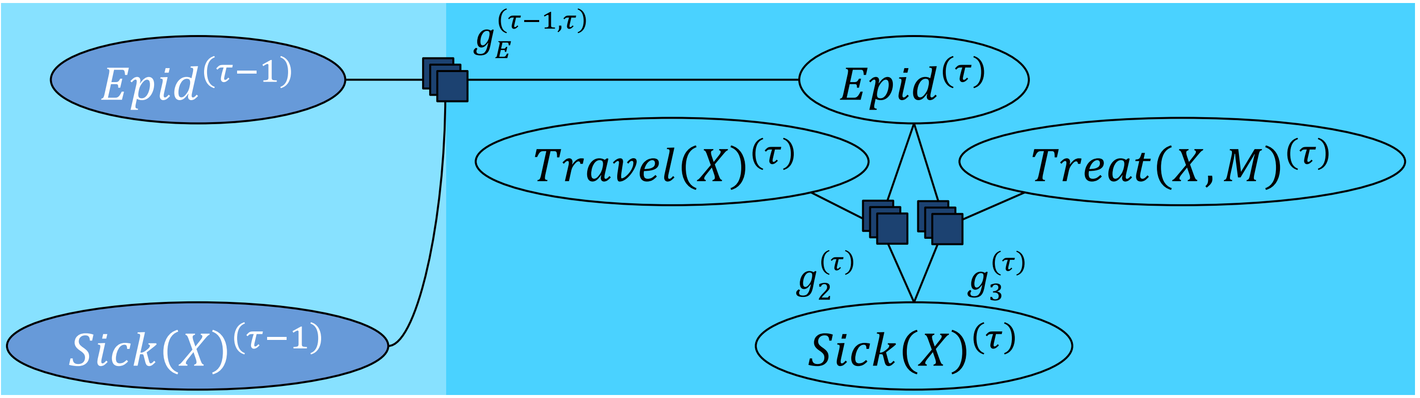
- Typical definition: Sequential model is a tuple $(M^0, M^\rightarrow)$, where
  - M^0 describes the behaviour at step 0 (only intra-slice behaviour as no previous step)
  - $M^\rightarrow$ is a 2-slice model describing intra- and inter-slice behaviour
  - Existing model types: dynamic BNs, HMMs, dynamic parfactor models, dynamic MLNs, …
- Frequent connection between $M^0, M^\rightarrow$
  - In factor-based models: Intra-slice behaviour in $M^\rightarrow$ for step τ equal to behaviour in M^0, i.e., intra-slice parfactors in $M^\rightarrow$ with $\tau = 0$
  - M^0 may also contain prior information encoded in distributions
    - In BN-based models: prior distributions in M^0

Independences in Dynamic Models

- *Observation*:
  All paths from one step to the next go through the PRVs $A^{(\tau-1)}$ of the inter-slice parfactors $g^{(\tau-1,\tau)}$ in $M^{\rightarrow}$

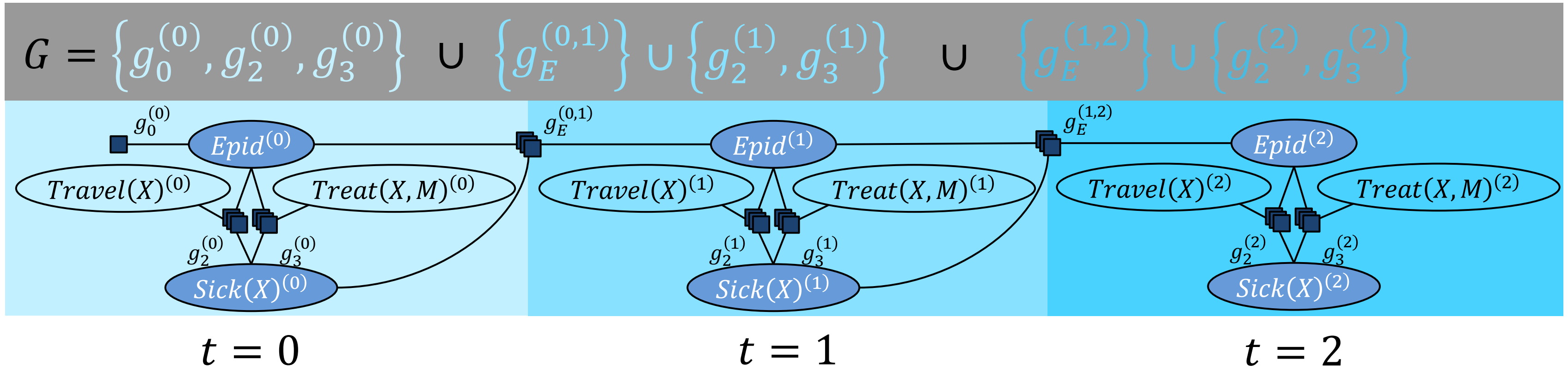


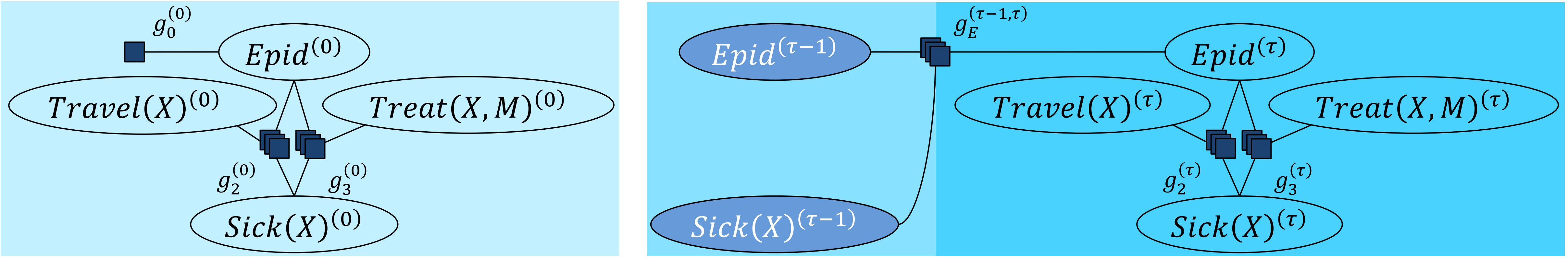




Independences in Dynamic Models

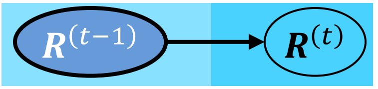
- Fact: Conditioning on these PRVs makes a current step independent of the previous steps
  - Collect information on these PRVs as a message before progressing in time

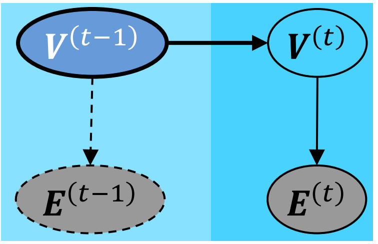


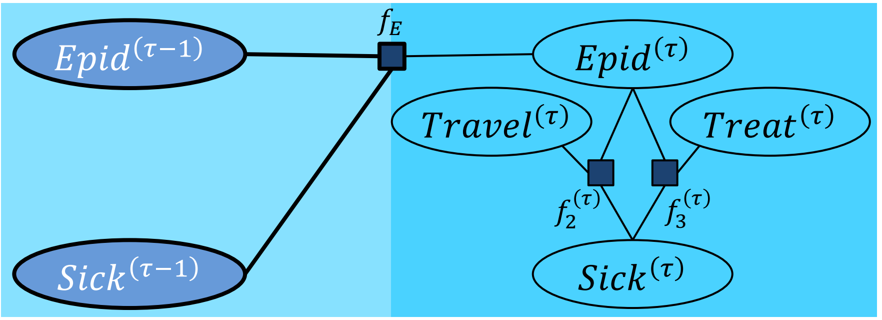


Interfaces

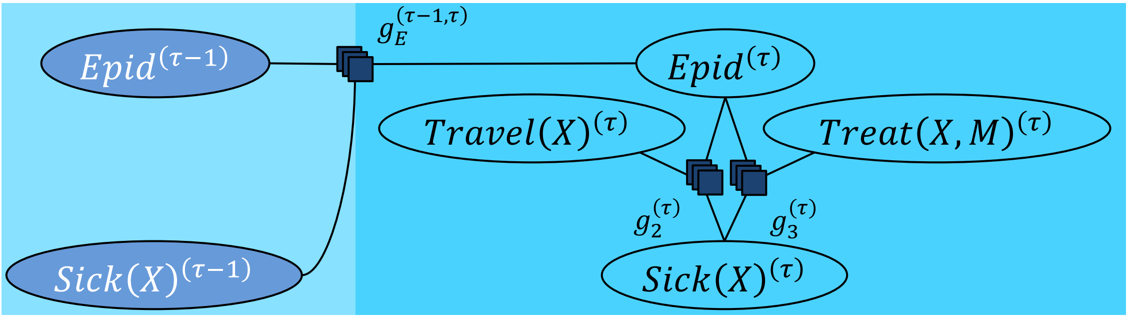
- Forward interface $\boldsymbol{I}^{(\tau-1)}$:
  Separating subset between $M^{(\tau-1)}$ and $M^{(\tau)}$
  - Trivial separators, usually not minimal
    - Maximal interface: $\boldsymbol{R}$
    - If $\boldsymbol{R} = \boldsymbol{V} \cup \boldsymbol{E}$: maximal interface $\boldsymbol{V}$
  - Set of PRVs with index $\tau - 1$ in $M^{\rightarrow}$
    - $\boldsymbol{I}^{(\tau-1)} = \{A^{(\tau-1)} | A^{(\tau-1)} \in \mathrm{rv}(M^{\rightarrow})\}$
      - Can only appear in inter-slice parfactors or MLN formulas







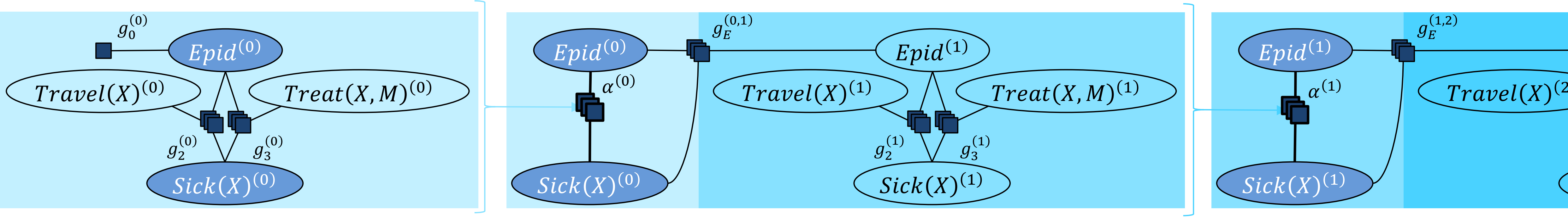
$\boldsymbol{I}$ also separates backwards from $M^{(\tau)}$ to $M^{(\tau-1)}$;
there is also an explicit *backward interface*,
which is determined in direction of τ to $\tau - 1$
[More details in Kevin Murphy's dissertation, 2002]



Interfaces To Move Through Time

- When inference for step t done
  - Query for $\boldsymbol{I}^{(t)}$ in current model $M^{(t)}$
    - Comparable to messages in FO jtrees: Calculate message $\alpha^{(t)}$ over separator $\boldsymbol{I}^{(t)}$
  - ...model $M^{(t+1)}$
    - ...$^{1} \cup \alpha^{(t)}$
    - ...connection to $M^{(0:t)}$, as all information of $M^{(0:t)}$ encoded in $\alpha^{(t)}$
- Use FO jtrees and LJT for moving through time

Helper structure and lifted inference algorithm to answer multiple queries

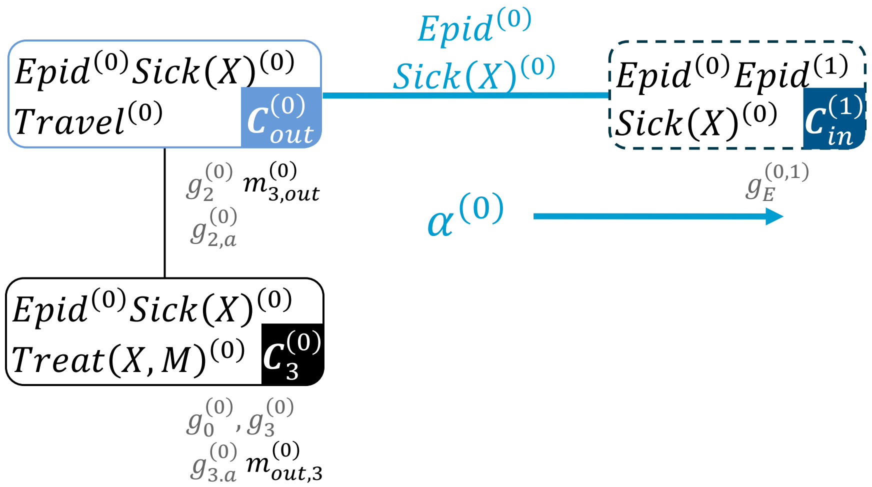


Progressing in Time: Example

- $t = 0$
  - Current FO jtree
  - Enter evidence $\boldsymbol{e}^{(0)} = \{sick(alice)^{(0)}\}$
  - Send intra-slice messages: $m^{(0)}_{3,out}$, $m^{(0)}_{out,3}$
  - Answer queries $P\left(R^{(0)}_i \middle| \boldsymbol{e}^{(0)}\right)$
- Inter-slice message: $\alpha^{(0)}$
  - Eliminate all non-separator PRVs, here $Travel(X)^{(0)}$, from the local model $G^{(0)}_{out} = \{g^{(0)}_2, g^{(0)}_{2,a}\}$ and $m^{(0)}_{3,out}$
  - Send result as message $\alpha^{(0)}$ to $\boldsymbol{C}^{(1)}_{in}$

$\alpha^{(0)}$ contains all information from $G^{(0)}$ including $\boldsymbol{e}^{(0)}$, which makes slice 0 and 1 independent.

Next: Instantiate an FO Jtree for $\tau = 1$ and add $\alpha^{(0)}$ to the local model of $\boldsymbol{C}^{(1)}_{in}$.

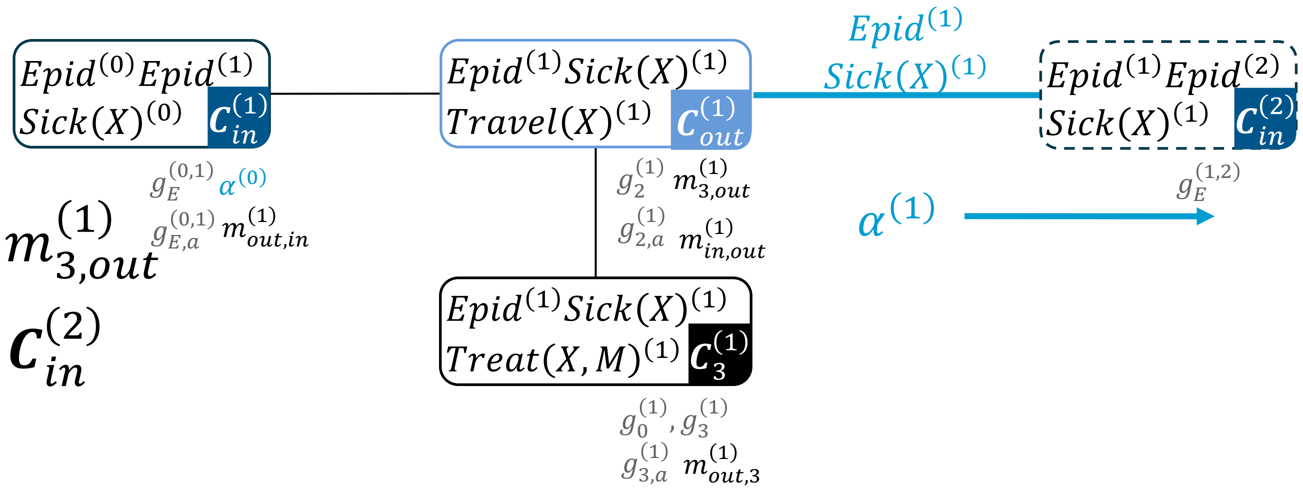


Progressing in Time: Example

- $t = 1$
  - Current FO jtree with $\alpha^{(0)}$
  - Enter evidence $\boldsymbol{e}^{(1)} = \{sick(alice)^{(1)}\}$
  - Send intra-slice messages: $m_{3,out}^{(1)}, m_{out,3}^{(1)}, m_{in,out}^{(1)}, m_{out,in}^{(1)}$
  - Answer queries $P\left(R_i^{(1)} \middle| \boldsymbol{e}^{(0:1)}\right)$
- Inter-slice message: $\alpha^{(1)}$
  - Eliminate $Travel(X)^{(1)}$ from $G_{out}^{(1)} = \{g_2^{(1)}, g_{2,a}^{(1)}\}$, $m_{in,out}^{(1)}$, and $m_{3,out}^{(1)}$
  - Send result as message $\alpha^{(1)}$ to $\boldsymbol{C}_{in}^{(2)}$

The information in $\alpha^{(0)}$ is distributed during the intra-slice message passing and therefore also included in $\alpha^{(1)}$. $\alpha^{(1)}$ now contains all information from $G^{(0:1)}$ including $\boldsymbol{e}^{(0:1)}$, which makes slice 1 and 2 independent.

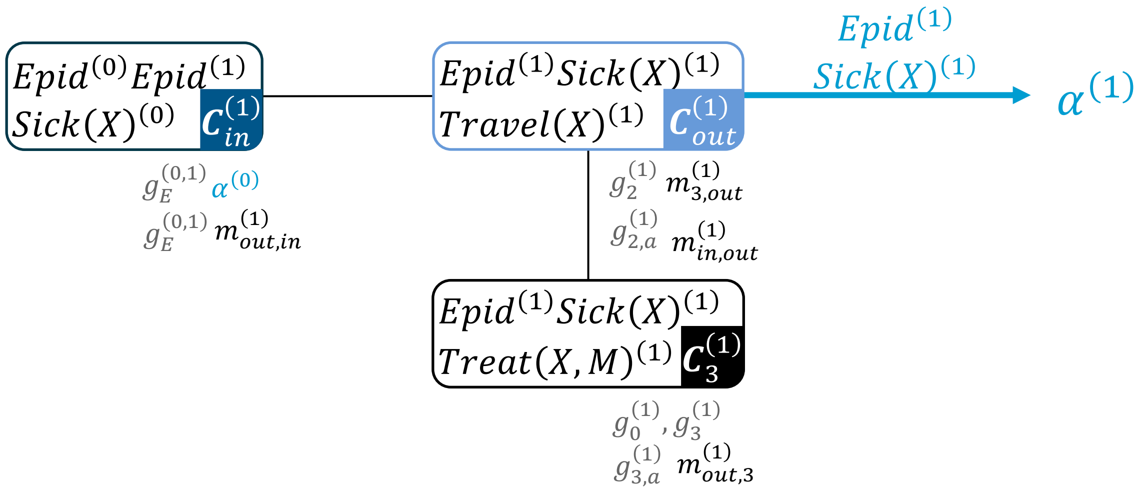
Next: Instantiate an FO Jtree for $\tau = 2$ and add $\alpha^{(1)}$ to the local model of $\boldsymbol{C}_{in}^{(2)}$.



Filtering Queries

- Queries one can answer in this way: *Filtering* queries
  - $P\left(R^{(t)} \middle| \boldsymbol{e}^{(0:t)}\right)$
  - Queries for random variables / grounded instances of PRVs / parameterised queries of the current slice t given all evidence $\boldsymbol{e}^{(0:t)}$ up until t
  - Advantages
    - Only a current FO jtree necessary
    - One additional message ($\alpha^{(t)}$) to progress in time

- What about *prediction* and *hindsight* queries?
  - $P\left(R^{(\pi)} \middle| \boldsymbol{e}^{(0:t)}\right)$, $\pi \neq t$



Complexity

- LJT complexity for message passing and query answering

$$O_{MP} = O\left(n_J \cdot \log_2(n) \cdot r^{w_g} \cdot n^{r_\# w_\#}\right)$$

$$O_{QA} = O\left(\log_2(n) \cdot r^{w_g} \cdot n^{r_\# w_\#}\right)$$

Lifted calculations Largest parfactor size

  - n_T, n_J number of nodes in FO dtree/jtree
  - n largest domain size
  - r largest range size, $r_\#$ largest range size of a PRV in a CRV
  - w_g largest ground width
  - $w_\#$ largest counting width
- LDJT: Moving forward in time
  - Uses message passing within each time step → O_{MP}
  - Uses LJT query answering for inter-slice message → O_{QA}

Complexity

- Given a maximum number T occurring over all M queries ($M = \sum_{t=0}^{T} m_t$, $m = \frac{1}{T}\sum_{t=0}^{T} m_t$)
  - Moving forward: $T \cdot \left(O_{MP} + O_{QA}\right)$
  - *Best* worst case at a time step $\tau = \tau' \in \{0, \dots, T\}$
    - Filtering query for τ', each with O_{QA}
  - Overall,

$$T \cdot \left(O_{MP} + O_{QA}\right) + M \cdot O_{QA} = \left(T \cdot n_J + T + M\right) \cdot O\left(\log_2(n) \cdot r^{w_g} \cdot n^{r_\# w_\#}\right)$$

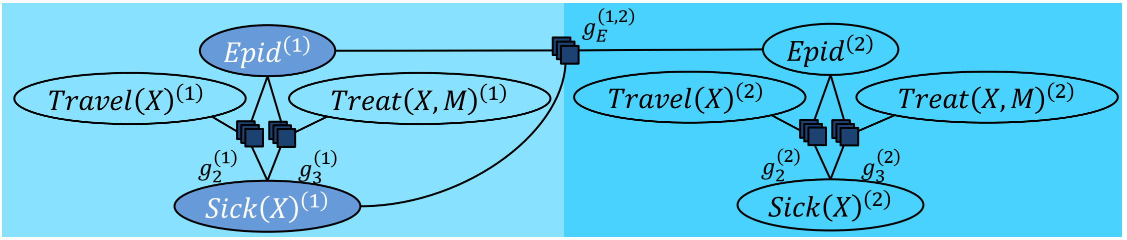
$$= O\left(\left(T \cdot n_J + M\right) \cdot \log_2(n) \cdot r^{w_g} \cdot n^{r_\# w_\#}\right)$$

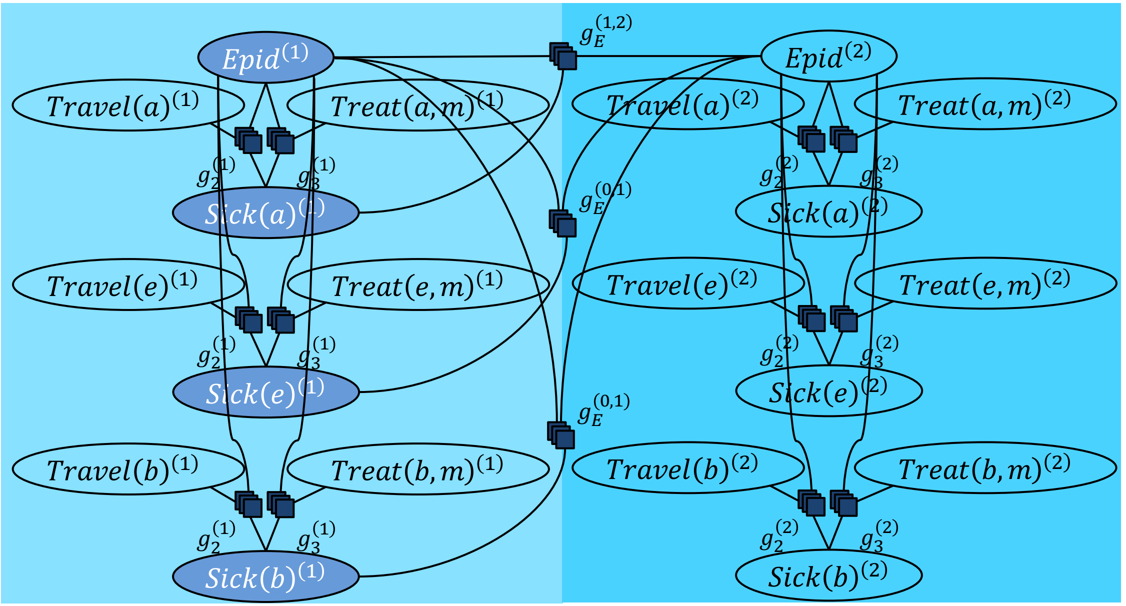
$$= O\left(\left(T \cdot n_J + T \cdot m\right) \cdot \log_2(n) \cdot r^{w_g} \cdot n^{r_\# w_\#}\right)$$

Compare LJT complexity
$O\left(\left(n_J + m\right) \cdot \log_2(n) \cdot r^{w_g} \cdot n^{r_\# w_\#}\right)$

Comparison to Ground Inference

- Earning of LVE vs. VE
  - $n_{gr(T)} \gg n_T$ (with large domains)
  - $w \gg (w_g + w_\#)$ (with count conversions)
    - Without count conversions, $w = w_g$, $w_\# = 0$
- In sequential case,
  - Earnings because of lifting the interface
    - Even without count conversions, $w \gg w_g$
  - Example: Grounding with
    - $\mathrm{dom}(X) = \{a, e, b\}$
    - $\mathrm{dom}(M) = \{m\}$





Interim Summary

- Interfaces to separate past from present and present from future
  - Inter-slice messages
    - Forward message to propagate all information up to current slice to next slice
    - Backward message to propagate all information down to current slice to previous slice
- LDJT algorithm
  - LJT algorithm for intra-slice inference
  - Inter-slice messages to move in time
  - Reduced complexity in terms of lifted interface

Temporal Lifted Inference

Keeping Symmetries in Temporal Inference

UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME
Marcel Gehrke

The Problem with Evidence

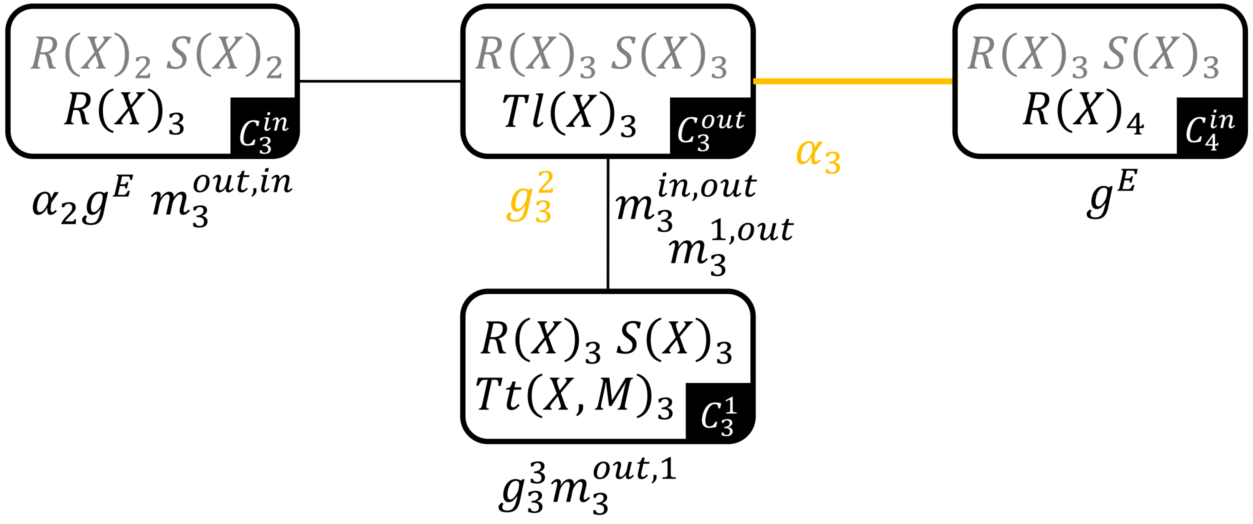
- Evidence can ground a model over time
- Non-symmetric evidence
  - Observe evidence for some instances in one time step
  - Observe evidence for a subset of these instances in another time step
  - Splits a logical variable slowly over time
- Vanilla junction trees for each time step
  - Without any splits
- Forward message carries over splits, leading to slowly grounding a model over time

Marcel Gehrke, Ralf Möller, and Tanya Braun. Lifted Taming Reasoning in Temporal Probabilistic Relational Models. In *ECAI-20 Proceedings of the 24th European Conference on Artificial Intelligence*, 2020.

Evidence over Time

- Slight variation of example
  - Replace $Epid_t$ with $R(X)_t$
- Evidence: $Tl_3(x_1) = true$
  - Split g_3^2 into
    - $g_3^{2=1}$ for x_1 and
    - $g_3^{2\neq 1}$ for $X \neq x_1$

- α_3 consists of
  - $m_3^{in,out}$
  - $m_3^{1,out}$
  - $g_3^{2=1}$ and $g_3^{2\neq 1}$ with $Tl_3(X)$ eliminated

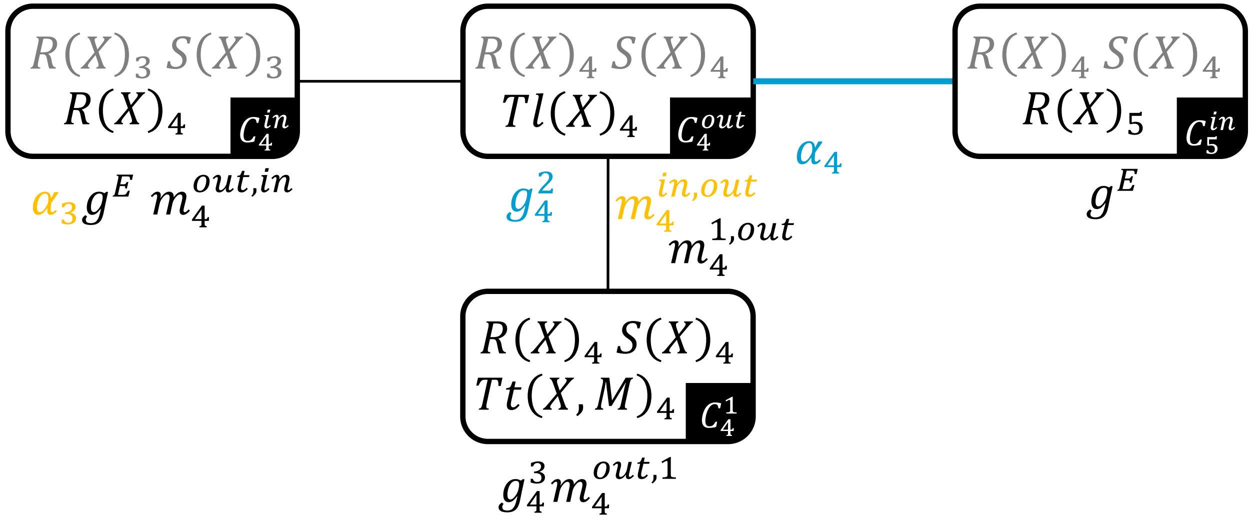


Evidence over Time

- Next step $\tau = 4$
- Evidence: $Tl_4(x_2) = true$
  - Split g_4^2 into
    - $g_4^{2=2}$ for x_2 and
    - $g_4^{2\neq 2}$ for $X \neq x_2$

X is slowly grounded

- α_3 contains
  - $g_3^{2=1}$ and $g_3^{2\neq 1}$ with $Tl_3(X)$ eliminated
  - For $m_4^{in,out}$, X is split w.r.t. x_1
- In α_4, X is split w.r.t. x_1 and x_2



Undoing Splits

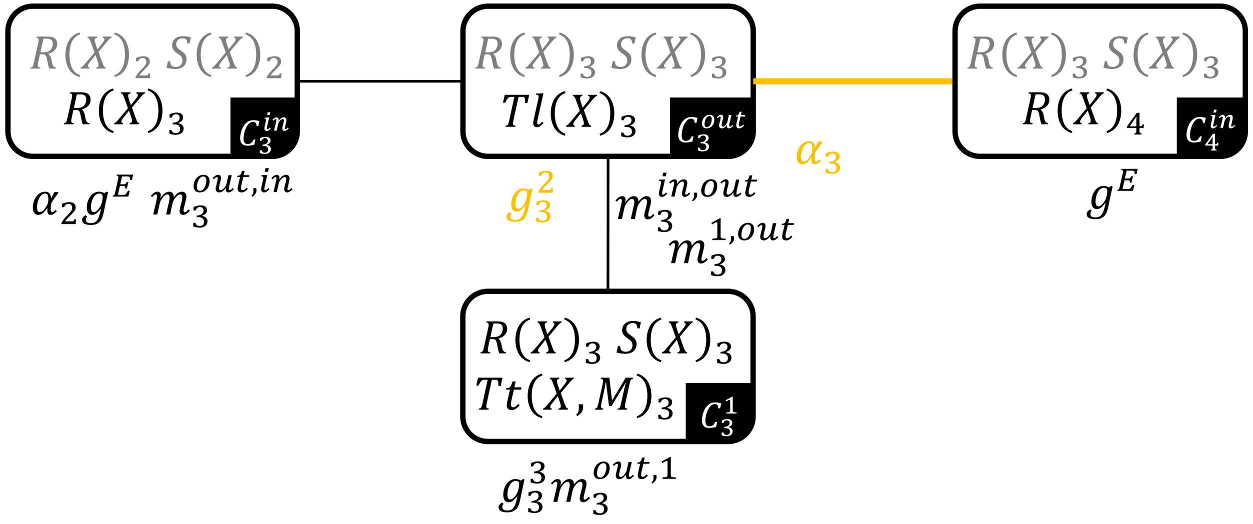
- Need to undo splits to

  keep reasoning polynomial w.r.t. domain sizes

1. Where can splits be undone efficiently?
2. How to undo splits?
3. Is it reasonable to undo splits?
   - Effect of slight differences in evidence?
   - Impact of evidence vs. temporal behaviour of model?

1. Where Can Splits Be Undone Efficiently?

- Evidence causes splits in a logical variable in the same way in all factors in a model
- LDJT always instantiates a vanilla junction tree
- Forward message carries over splits



2. How to Undo Splits?

- The colour passing algorithm can efficiently identify exact symmetries
- But
  - Evidence causes differences in distributions

- Need to *find* approximate symmetries to undo splits caused by evidence
- Need a way to *merge* factors

Parag Singla and Pedro Domingos: Lifted First-order Belief Propagation. In *AAAI-08 Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, 2008.
Kristian Kersting, Babak Ahmadi, and Sriraam Natarajan. Counting Belief Propagation. In *UAI-09 Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009.
Babak Ahmadi, Kristian Kersting, Martin Mladenov, and Sriraam Natarajan. Exploiting Symmetries for Scaling Loopy Belief Propagation and Relational Training. In *Machine Learning*, 92(1):91-132, 2013.

Comparing Parfactors: Approaches

- Comparing all marginals is expensive
- Comparing the joint distribution over the complete interface is expensive
- Comparing marginals of a *subset* of PRVs can determine non-similar factors similar
  - E.g.,
    - $\mathrm{dom}(X) = \{x_1\}$

$R(X)$	$S(X)$	g
$false$	$false$	0
$false$	$true$	7
$true$	$false$	4
$true$	$true$	1

$R(X)$	$S(X)$	g
$false$	$false$	2
$false$	$true$	4
$true$	$false$	2
$true$	$true$	4

    - $P(S(x_1) = true)$: $\frac{2}{3}$ $\frac{2}{3}$
    - $P(R(x_1) = true)$: $\frac{5}{12}$ $\frac{1}{2}$

Comparing Parfactors

- Potentials determine distributions
- Similar ratios in potentials lead to similar marginals and similar factors
  - E.g.,
    - $\mathrm{dom}(X) = \{x_1\}$

$R(X)$	$S(X)$	g
$false$	$false$	4
$false$	$true$	3
$true$	$false$	2
$true$	$true$	1

$R(X)$	$S(X)$	g
$false$	$false$	3.9
$false$	$true$	3.1
$true$	$false$	2.1
$true$	$true$	0.9

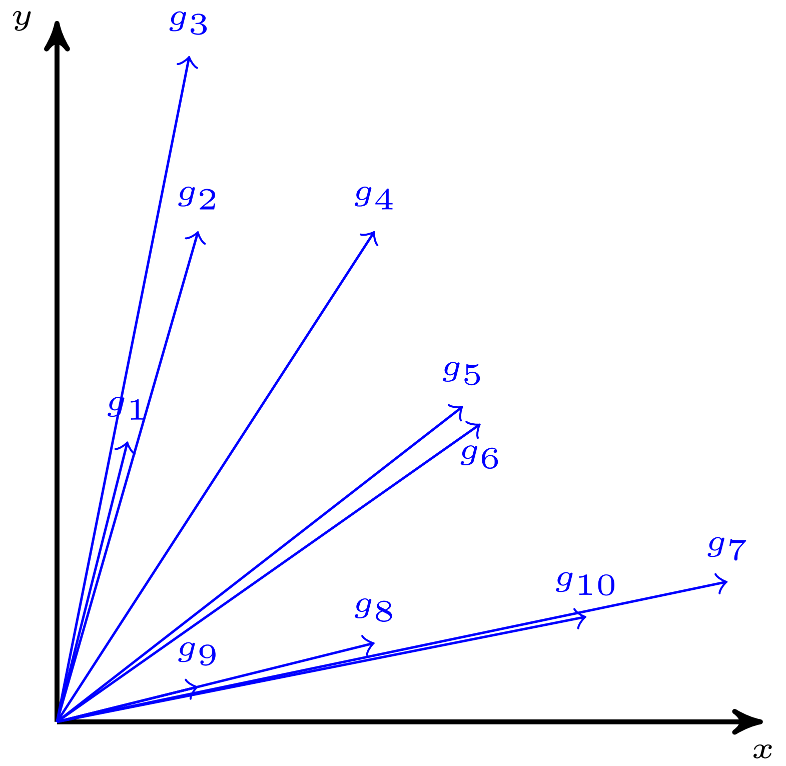
    - $P(S(x_1) = true)$: $\frac{4}{10}$ $\frac{4}{10}$
    - $P(R(x_1) = true)$: $\frac{3}{10}$ $\frac{3}{10}$
    - $P(S(x_1) = true, R(x_1) = true)$: $\frac{1}{10}$ $\frac{0.9}{10}$

Identifying Similar Groups

- Groups are equal if they have the same full joint distribution
- Full joint distribution computationally hard to get

→ Use parfactors as vector

→ If vectors of two groups point in same direction, they have a similar full joint distribution



Find Approximate Symmetries

- Cosine similarity for similarity of vectors

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- E.g.,

$Tl(X)$	$S(X)$	g
$false$	$false$	0
$false$	$true$	7
$true$	$false$	4
$true$	$true$	1

$Tl(X)$	$S(X)$	g
$false$	$false$	2
$false$	$true$	4
$true$	$false$	2
$true$	$true$	4

  – $\cos(\theta) = \frac{0 \cdot 2 + 7 \cdot 4 + 4 \cdot 2 + 1 \cdot 4}{\sqrt{0+49+16+1} \cdot \sqrt{4+16+4+16}} \sim 0.7785$

Find Approximate Symmetries

- Cosine similarity for similarity of vectors

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- E.g.,

$Tl(X)$	$S(X)$	g
$false$	$false$	4
$false$	$true$	3
$true$	$false$	2
$true$	$true$	1

$Tl(X)$	$S(X)$	g
$false$	$false$	3.9
$false$	$true$	3.1
$true$	$false$	2.1
$true$	$true$	0.9

  – $\cos(\theta) = \frac{4 \cdot 3.9 + 3 \cdot 3.1 + 2 \cdot 2.1 + 1 \cdot 0.9}{\sqrt{16+9+4+1} \cdot \sqrt{15.21+9.61+4.41+0.81}} \sim 0.9993$

Find Approximate Symmetries

- Cosine similarity for similarity of vectors

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- E.g.,

$Tl(X)$	$S(X)$	g
$false$	$false$	4
$false$	$true$	3
$true$	$false$	2
$true$	$true$	1

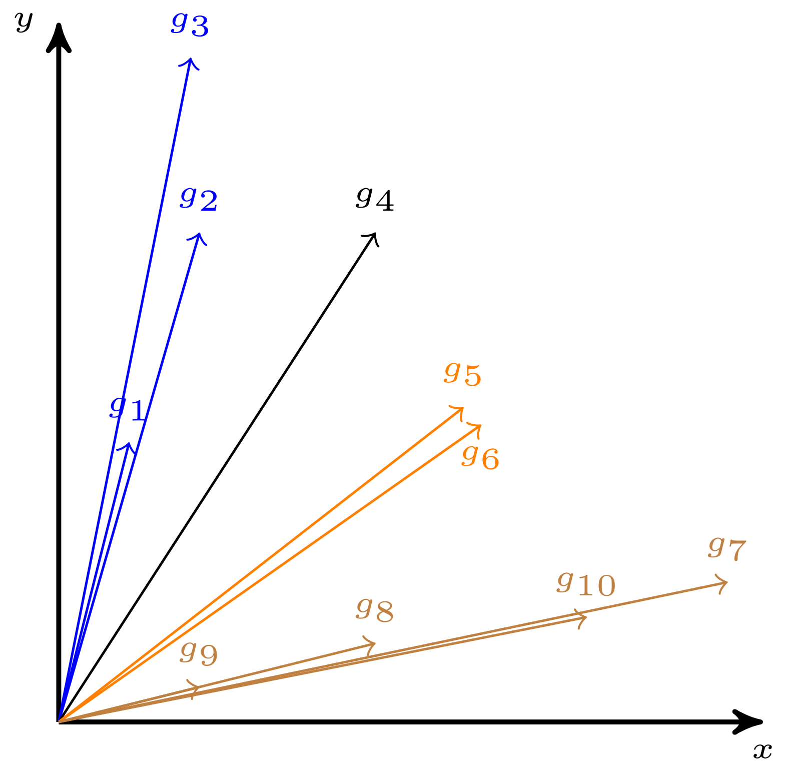
$Tl(X)$	$S(X)$	g
$false$	$false$	8
$false$	$true$	6
$true$	$false$	4
$true$	$true$	2

  - $\cos(\theta) = \frac{4 \cdot 8 + 3 \cdot 6 + 2 \cdot 4 + 1 \cdot 3}{\sqrt{16+9+4+1} \cdot \sqrt{64+36+16+4}} = 1$

Use $1 - \cos(\theta)$ as distance function in clustering algorithm

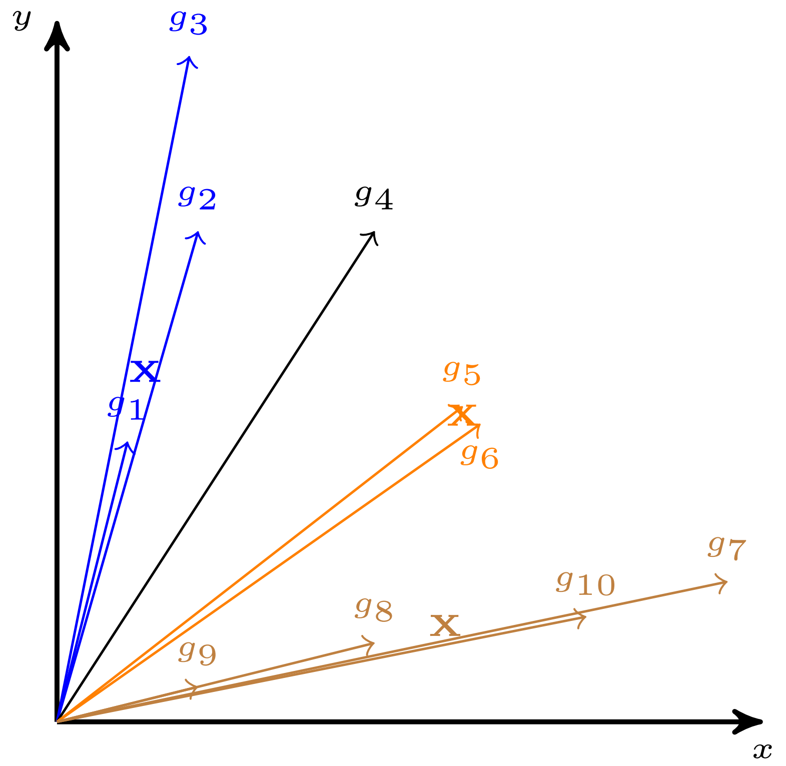
Cluster Groups

- Density-based clustering as unknown number of clusters
  - E.g., DBSCAN
- Cosine similarity as distance function
- Use ANOVA for testing fitness of clustering
  - Hypothesis testing
  - Do the cluster means sufficiently distinguish the clusters?



Merge Groups

- Merge groups of cluster by calculating mean of cluster while accounting for groundings
- Replace old groups with merged group in forward message



Merging Parfactors

- Merge similar parfactors, *accounting for groundings*

$|\text{dom}(X)| = 4$

$Tl(X)$	$S(X)$	g
$false$	$false$	4
$false$	$true$	3
$true$	$false$	2
$true$	$true$	1

$|\text{dom}(X')| = 4$

$Tl(X')$	$S(X')$	g
$false$	$false$	7.9
$false$	$true$	6
$true$	$false$	3.9
$true$	$true$	2.1

$|\text{dom}(X'')| = 2$

$Tl(X'')$	$S(X'')$	g
$false$	$false$	15.7
$false$	$true$	12.2
$true$	$false$	8.1
$true$	$true$	3.8

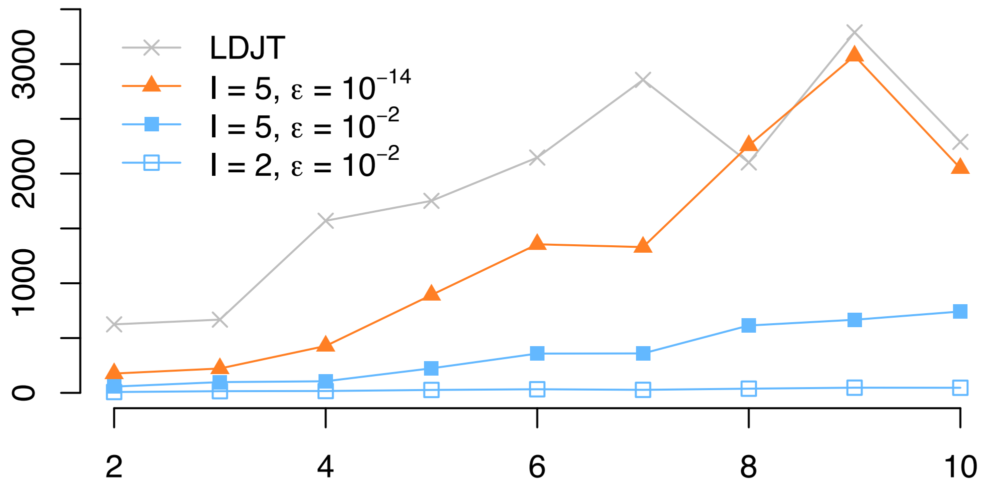
$\rightarrow$

$|\text{dom}(X)| = 10$

$Tl(X)$	$S(X)$	g
$false$	$false$	$\frac{(4 \cdot 4 + 7.9 \cdot 4 + 15.7 \cdot 2)}{10} = 7.9$
$false$	$true$	$\frac{(3 \cdot 4 + 6 \cdot 4 + 12.2 \cdot 2)}{10} = 6.04$
$true$	$false$	$\frac{(2 \cdot 4 + 3.9 \cdot 4 + 8.1 \cdot 2)}{10} = 3.98$
$true$	$true$	$\frac{(1 \cdot 4 + 2.1 \cdot 4 + 3.8 \cdot 2)}{10} = 2$

Runtimes and Error

- DBSCAN for Clustering (neighbourhood: $\varepsilon, n = 2$)
- ANOVA for checking fitness of clusters ($\alpha = 0.005$)
- Right: Runtimes in seconds
  - I denotes how often TAMe is run (every I'th run)
  - ε parameter for DBSCAN
- Below: prediction error of LDJT with TAMe compared to LDJT without TAMe

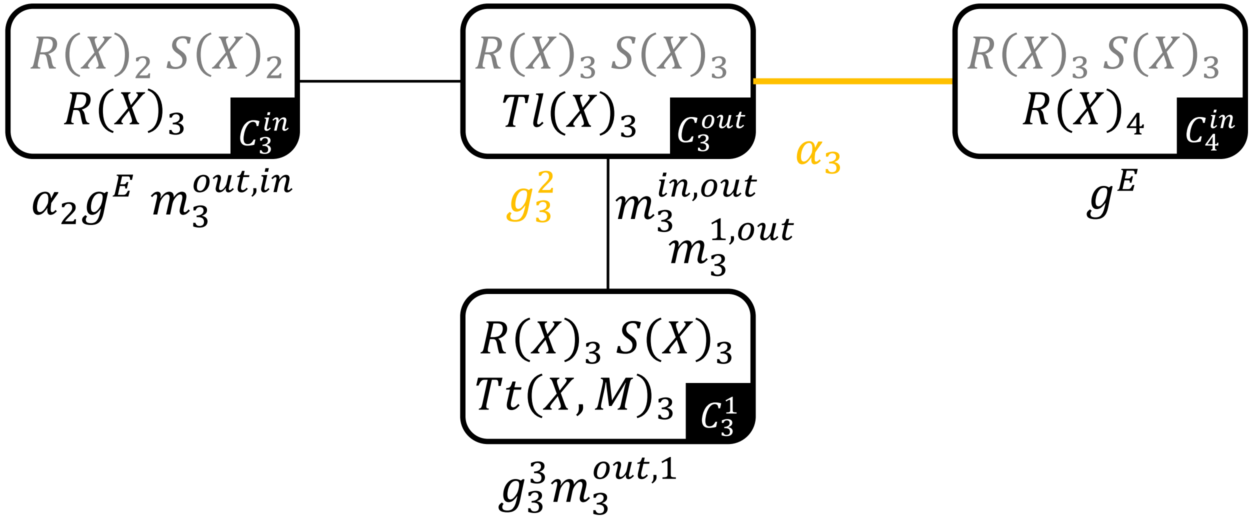


π	**Max**	**Min**	**Average**
0	0.0001537746121	0.0000000001720	0.0000191206488
2	0.0000000851654	0.0000000000001	0.0000000111949
4	0.0000000000478	0	0.0000000000068

Is It Reasonable to Undo Splits?

- *Approximate* forward message
- For each time step, sequential behaviour is multiplied onto the forward message
- Indefinitely bounded error due to sequential behaviour

Without any new evidence, the model would become fully lifted again!



Interim Summary

- Need to undo splits to
  keep reasoning polynomial w.r.t. domain sizes

1. Where can splits be undone efficiently?
   - Undo splits in a forward message
2. How to undo splits?
   - Find approximate symmetries
   - Merge based on groundings
3. Is it reasonable to undo splits?
   - Yes, due to the temporal model behaviour (indefinitely bounded error)
     - Achieve fully lifted model again without new evidence