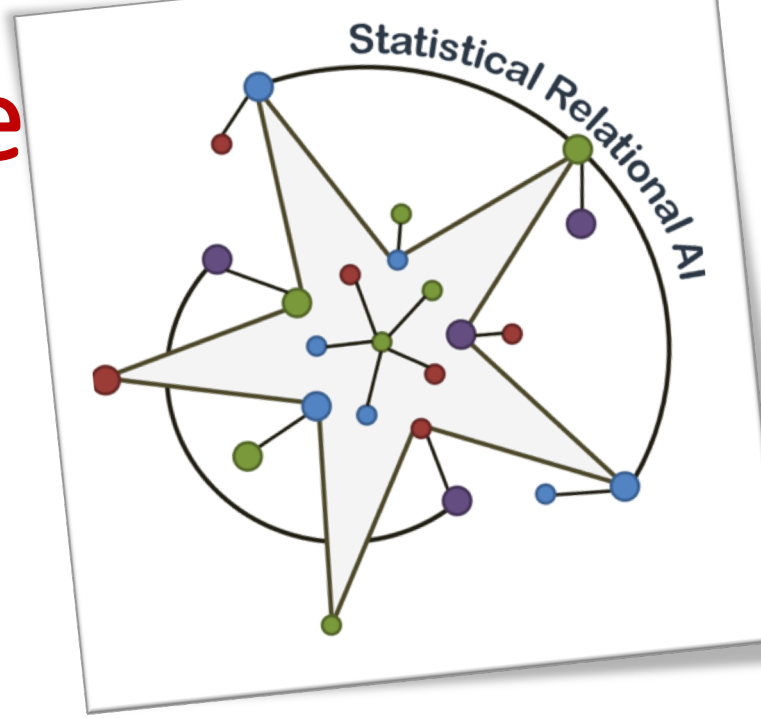
Exact Lifted Inference on Relational Temporal Models



Statistical Relational AI

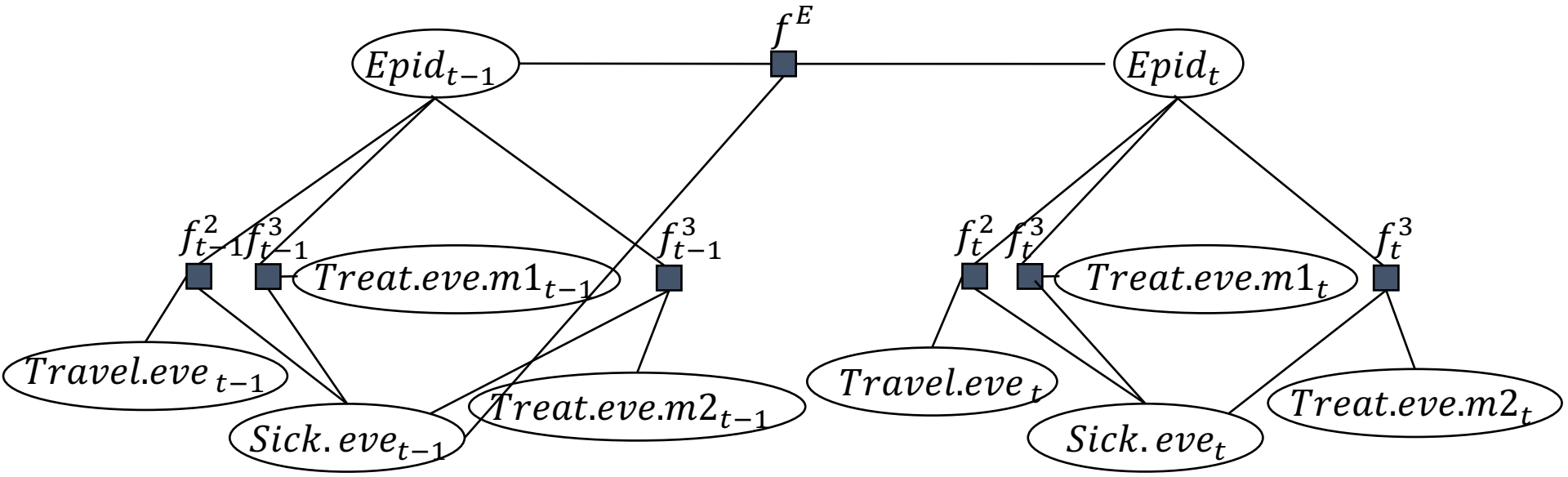
Tutorial at ICCS 2019

Marcel Gehrke, University of Lübeck

UNIVERSITÄT ZU LÜBECK

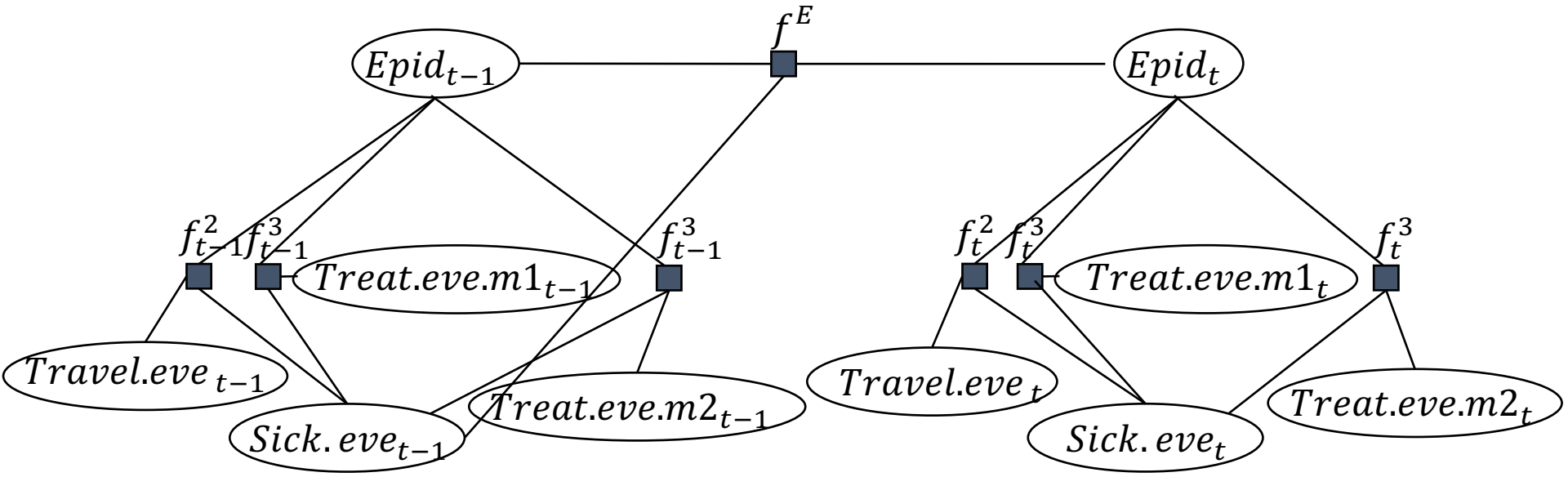
Propositional: Dynamic Model

- Temporal pattern
- Instantiate and unroll pattern
- Infer on unrolled model



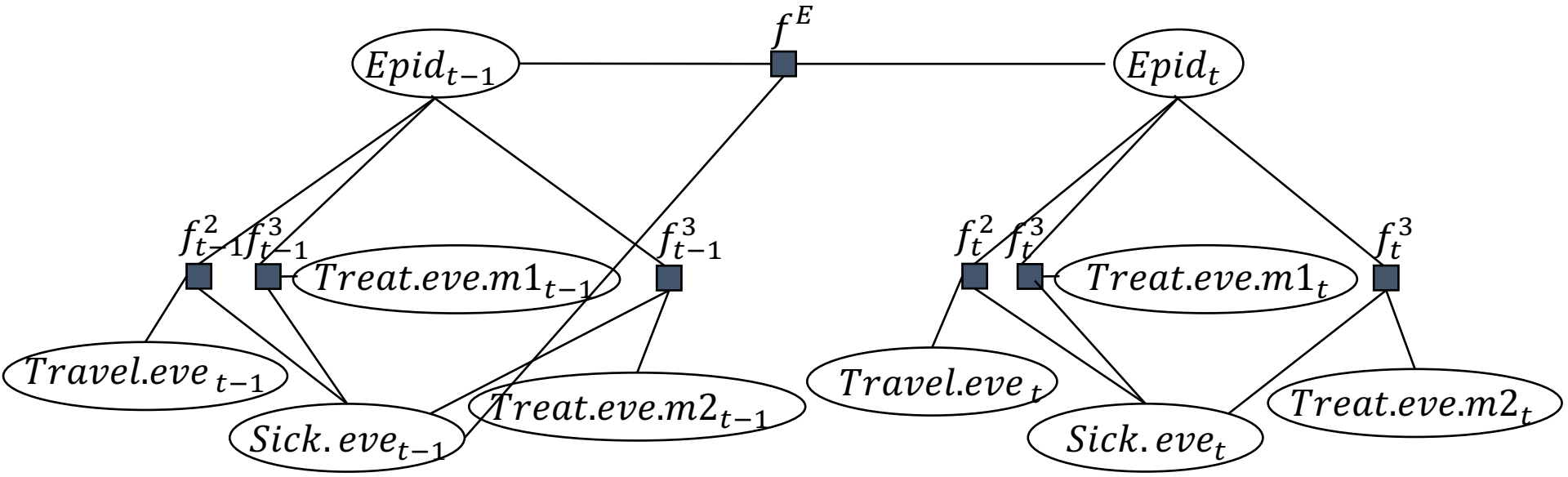
Dynamic Model: Inference Problems

- Marginal distribution query: $P\left(A_\pi^i \middle| E_{0:t}\right)$ w.r.t. the model:
  - Hindsight: $\pi < t$ (was there an epidemic $\pi - t$ days ago?)
  - Filtering: $\pi = t$ (is there an currently an epidemic?)
  - Prediction: $\pi > t$ (is there an epidemic in $\pi - t$ days?)



Propositional: Dynamic Model

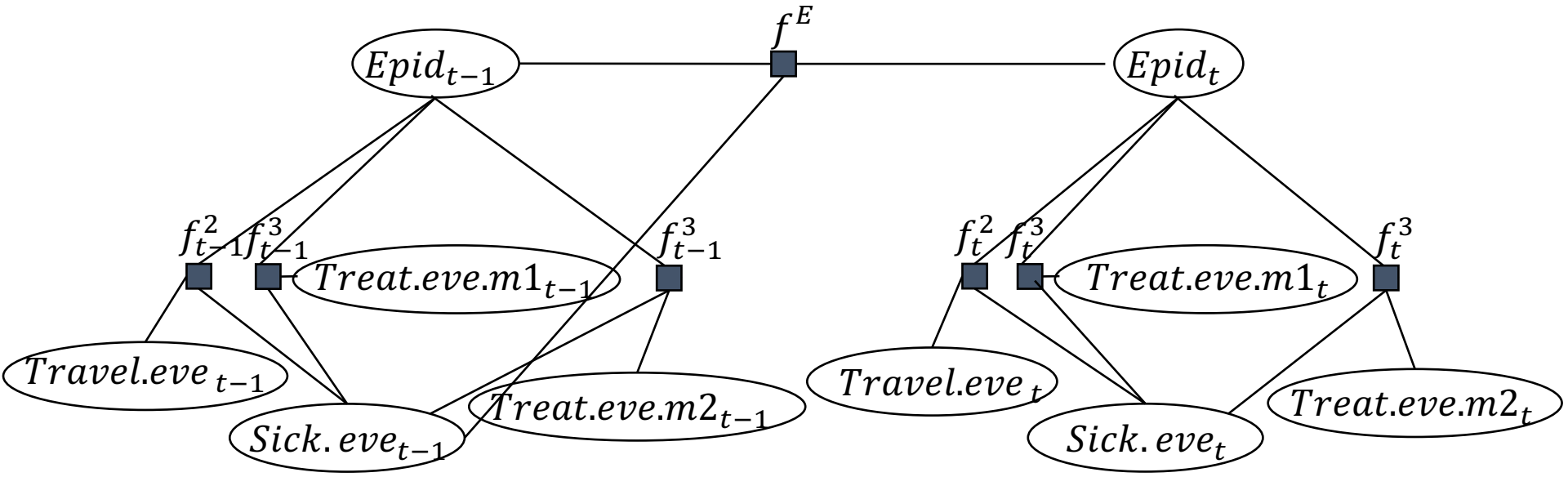
- Problems with unrolling:
  - Huge model (unrolled for T timesteps)
  - Redundant temporal information and calculations
  - Redundant calculations to answer multiple queries



Propositional: Interface Algorithm

Murphy (2002)

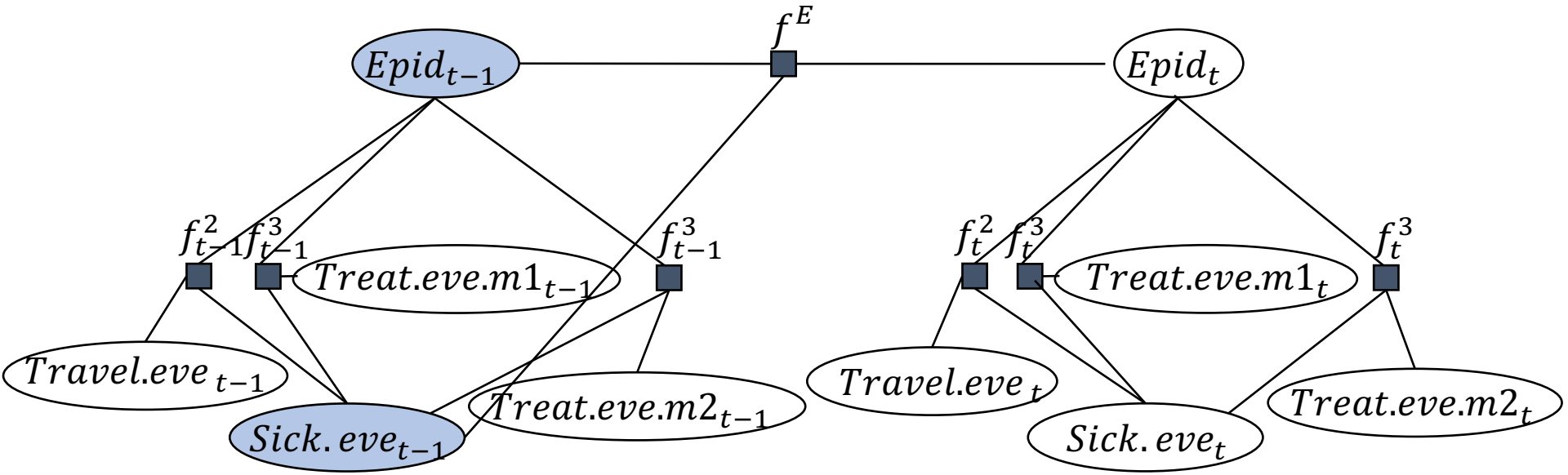
- Main idea: Use temporal conditional independences to perform inference on smaller model



Propositional: Interface Algorithm

Murphy (2002)

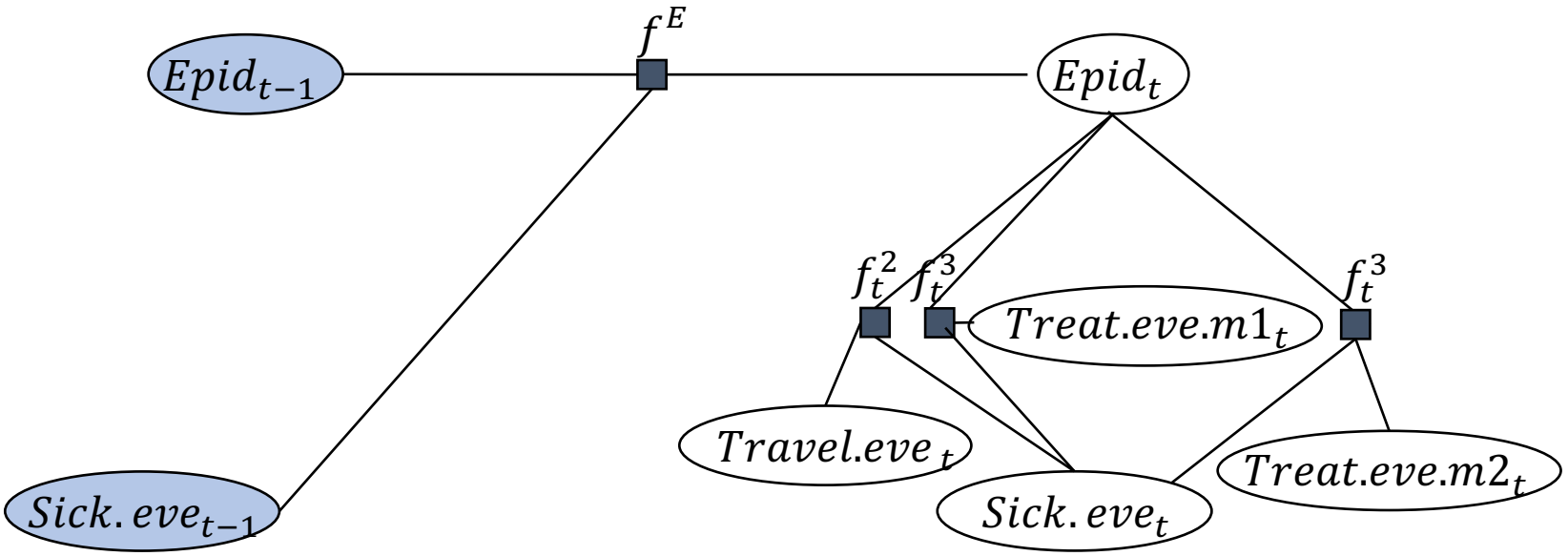
- Main idea: Use temporal conditional independences to perform inference on smaller model
  - Normally only a subset of random variables influence next time step



Propositional: Interface Algorithm

Murphy (2002)

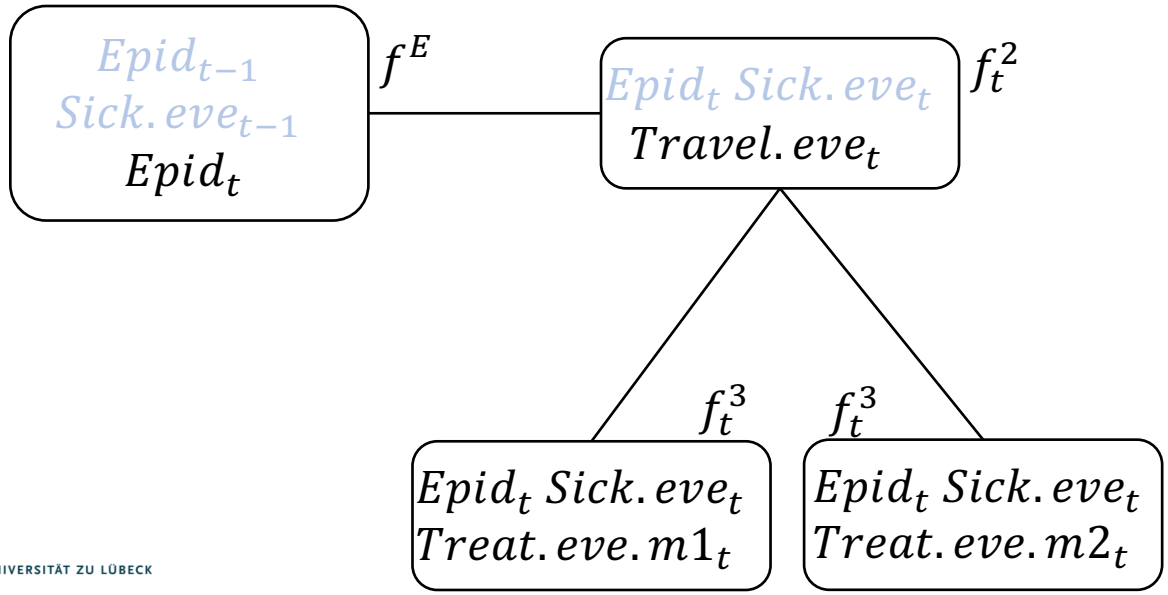
- Main idea: Use temporal conditional independences to perform inference on smaller model
  - Normally only a subset of random variables influence next time step
  - State description of interface variables ($Epid_{t-1}$ and $Sick.eve_{t-1}$) suffice to perform inference on time slice t
  - Proceed forward one time step at a time, using the same structure



Propositional: Interface Algorithm

Murphy (2002)

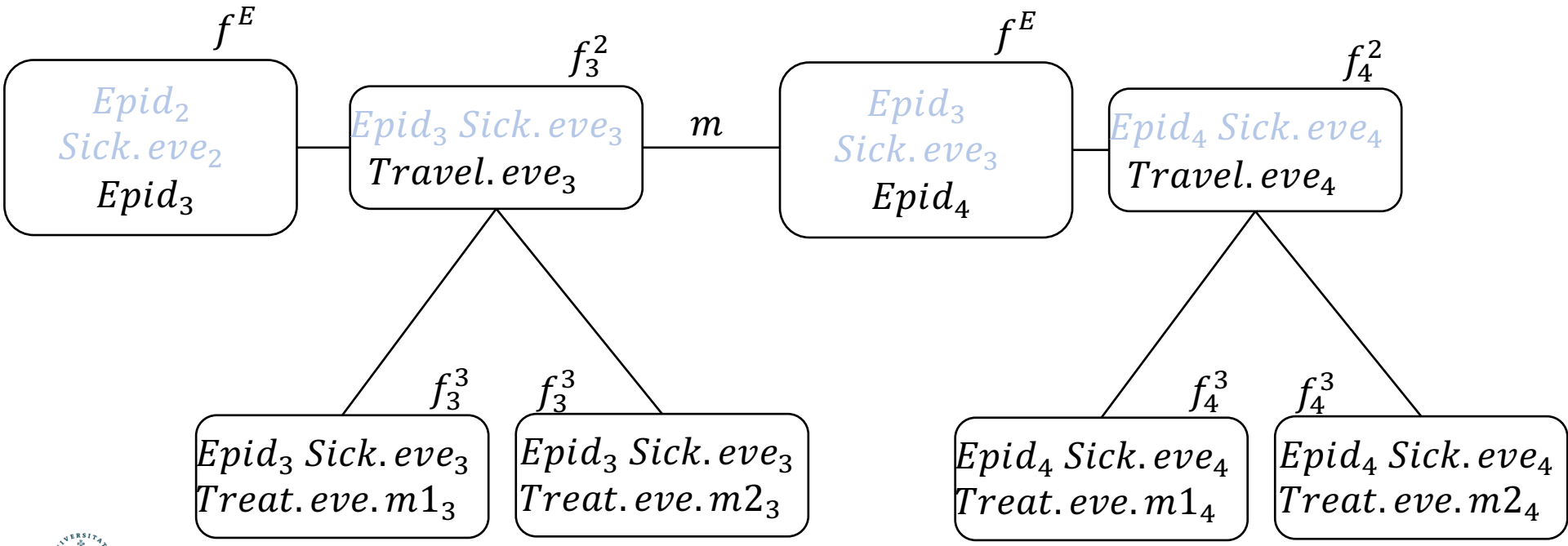
- Build Junction Tree from reoccurring structure
- Ensure that interface variable for time slice $t-1$ occur in one cluster and that interface variable for time slice t occur in one cluster



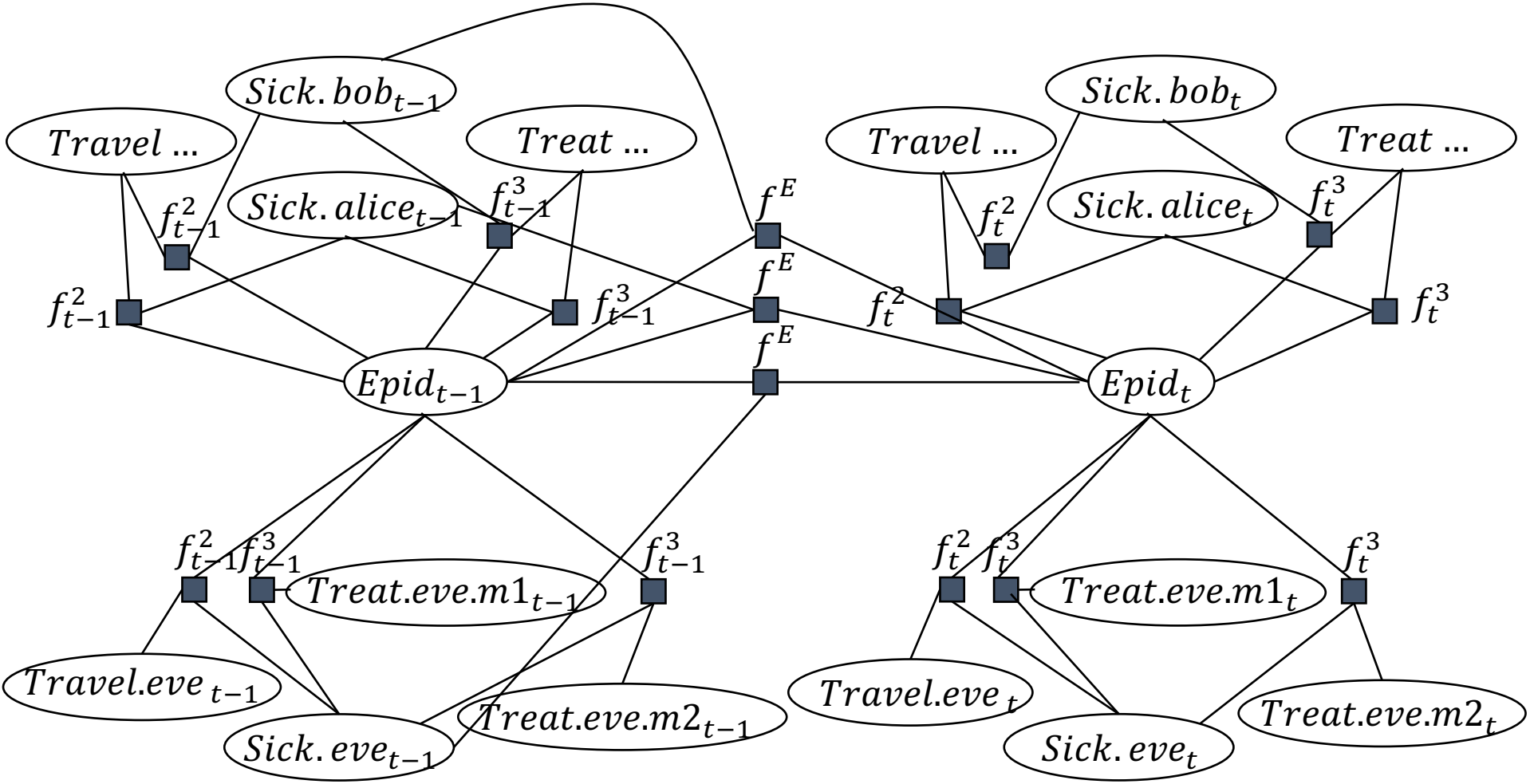
Propositional: Interface Algorithm

Murphy (2002)

- Perform inference on time slice 3
- How to perform inference on time slice 4?
  - Store state descriptions of interface variables in m
  - Distribute m with message pass

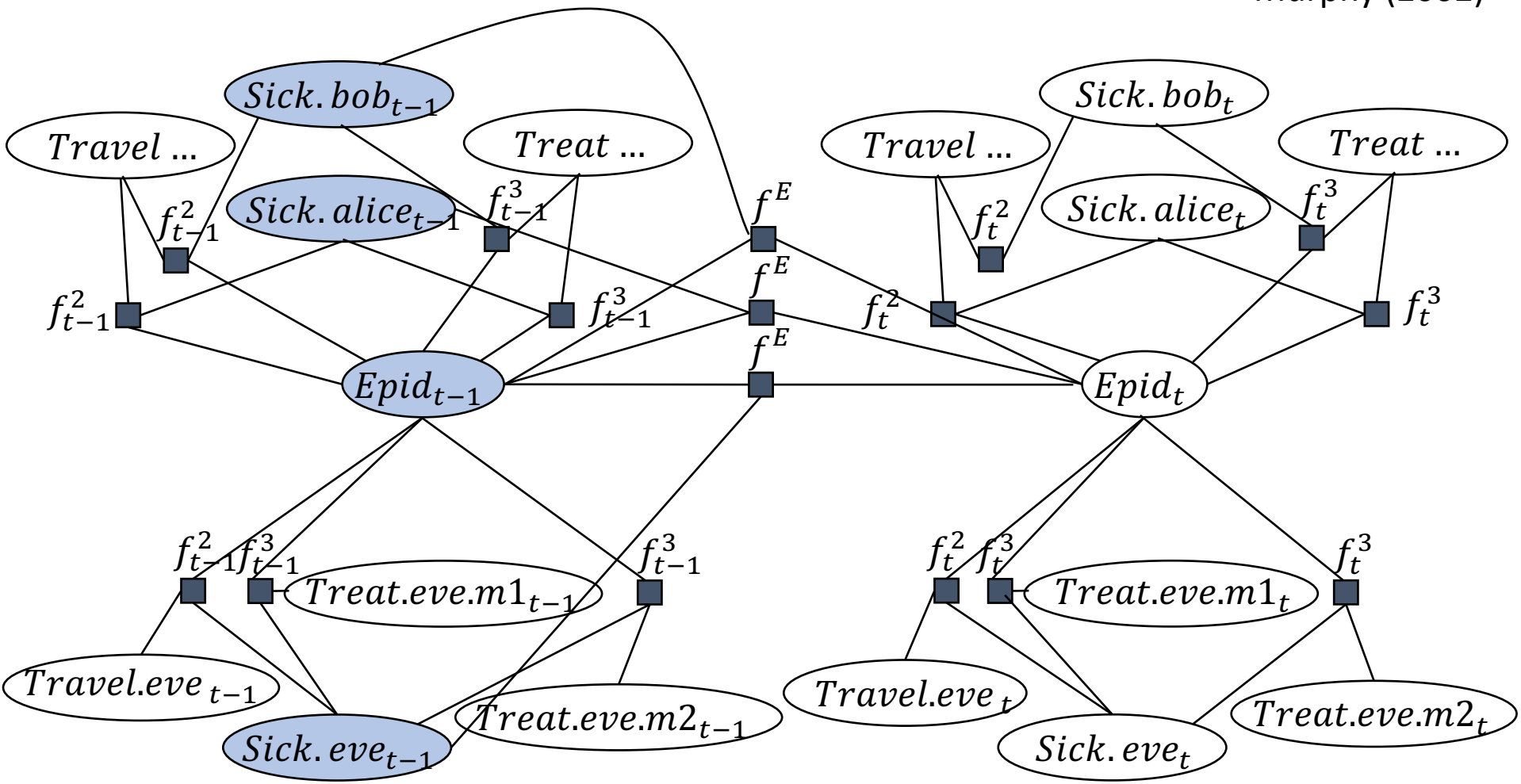


Propositional: Dynamic Model



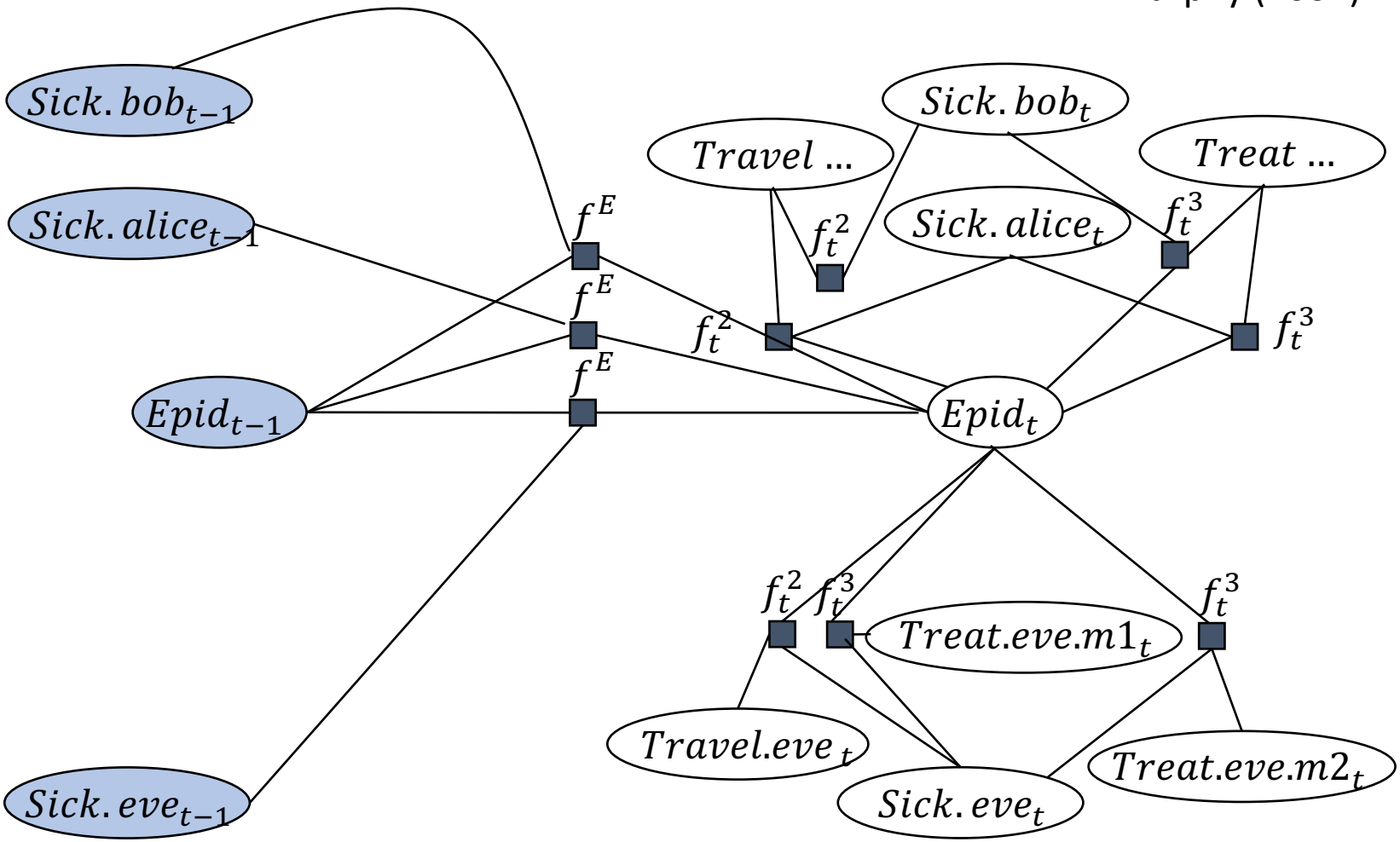
Propositional: Interface Algorithm

Murphy (2002)



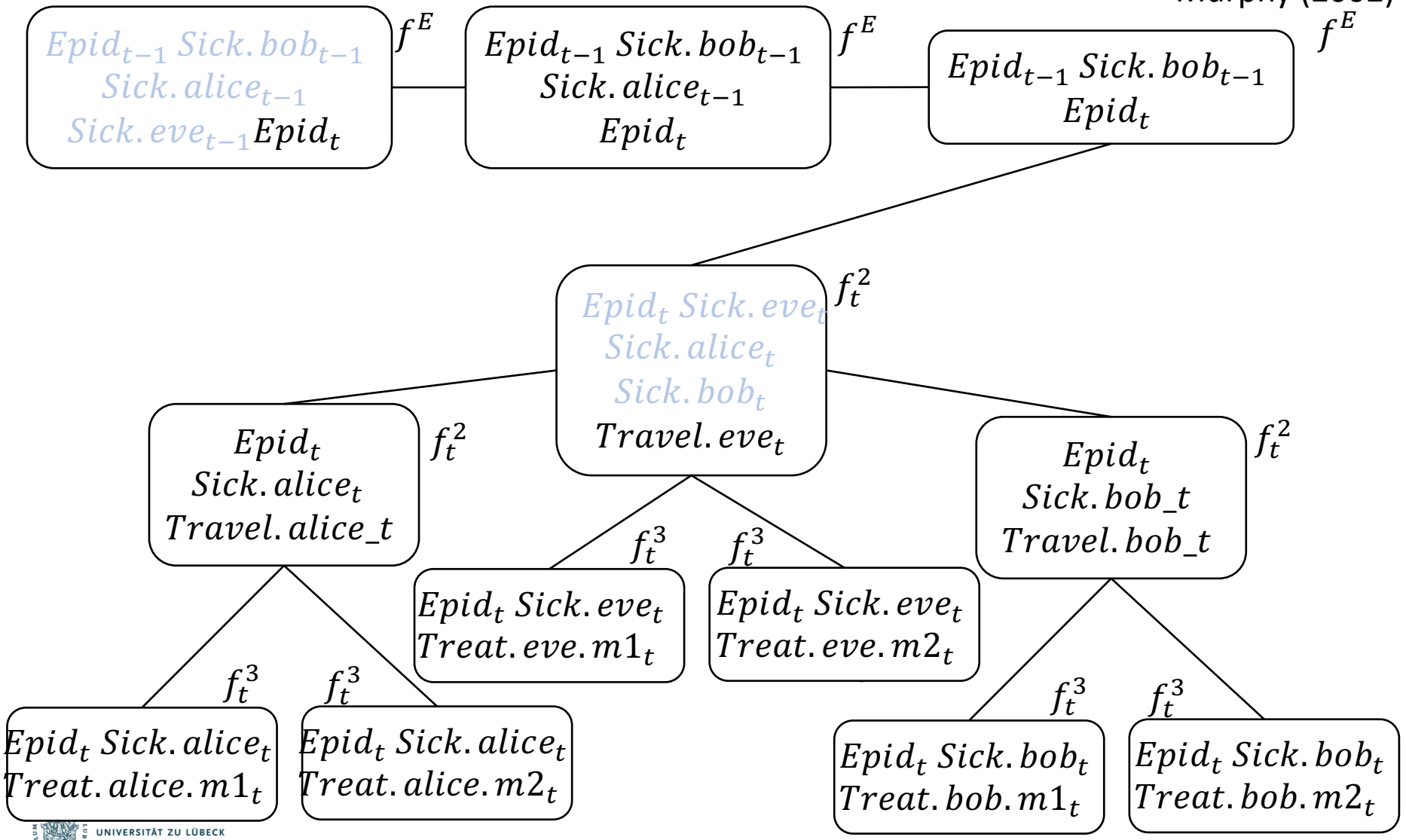
Propositional: Interface Algorithm

Murphy (2002)



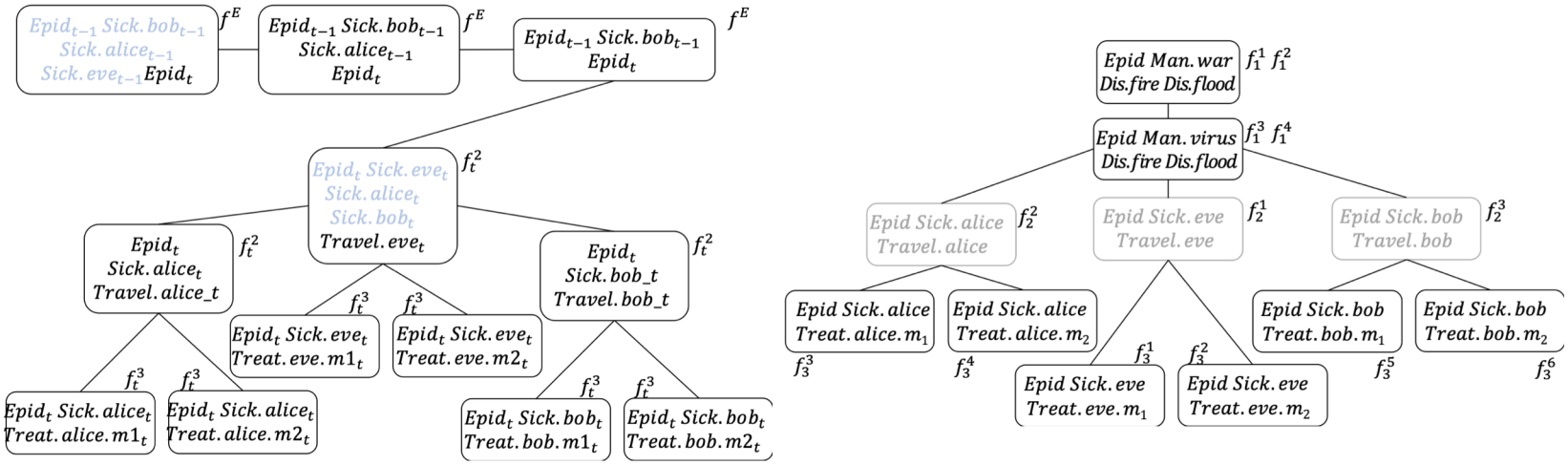
Propositional: Interface Algorithm

Murphy (2002)



Propositional: Interface Algorithm

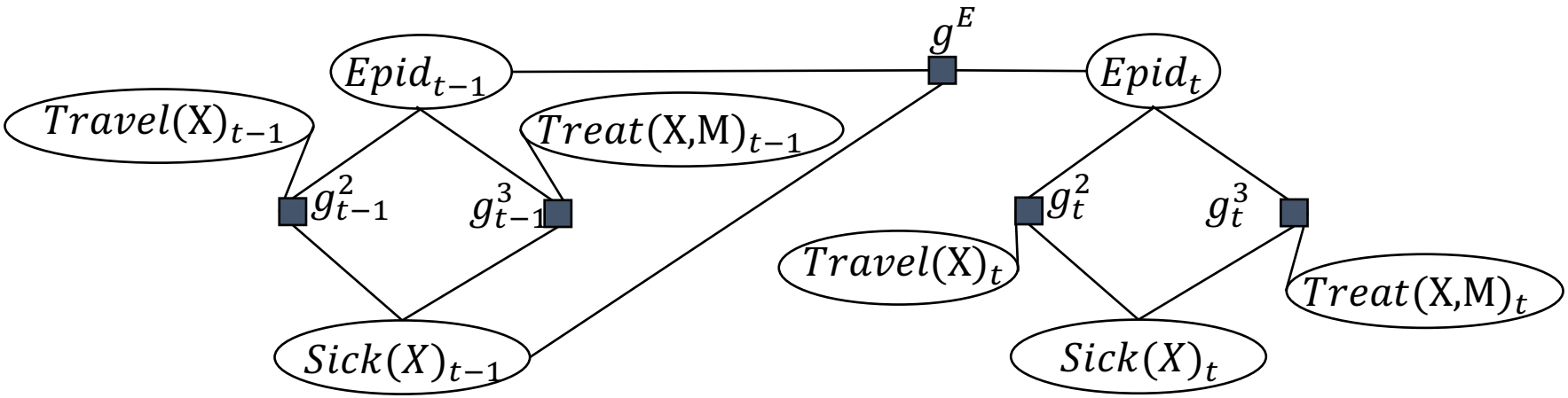
- For the static example increasing the domain size only increased the number of clusters
- Increasing the domain size of interface variables, increases the number of clusters and cluster size
- Inference is exponential in the largest cluster
- Can we lift the interface algorithm?



Lifted: Dynamic Model

Gehrke et al. (2018)

- Marginal distribution query: $P\left(A_\pi^i \middle| E_{0:t}\right)$ w.r.t. the model:
  - Hindsight: $\pi < t$ (was there an epidemic $\pi - t$ days ago?)
  - Filtering: $\pi = t$ (is there an currently an epidemic?)
  - Prediction: $\pi > t$ (is there an epidemic in $\pi - t$ days?)



Lifted Dynamic Junction Tree Algorithm: LDJT

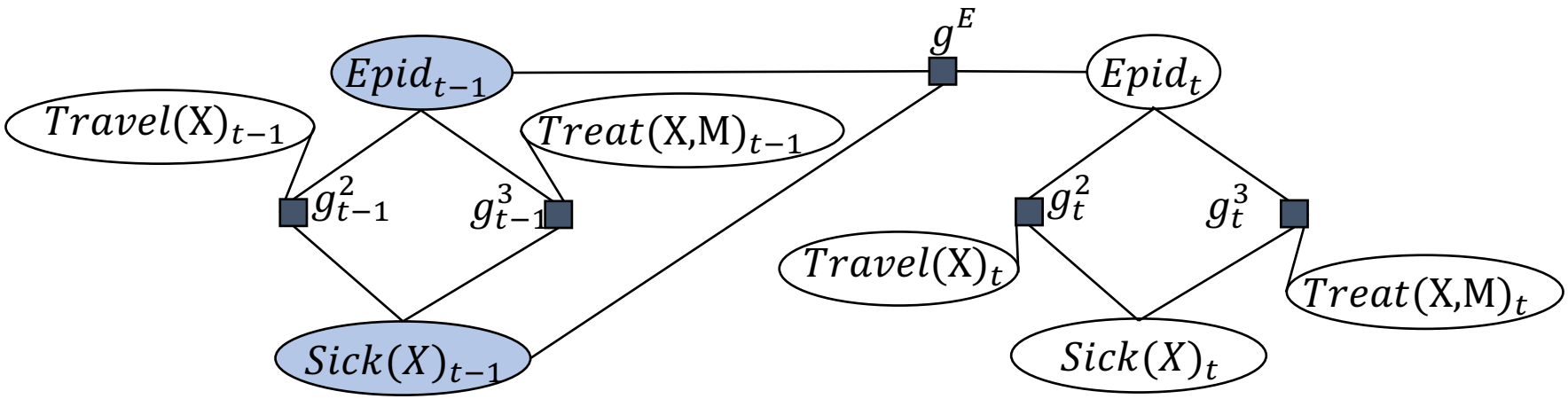
Gehrke et al. (2018)

- Input
  - Temporal model G
  - Evidence $\boldsymbol{E}$
  - Queries $\boldsymbol{Q}$
- Algorithm
  1. Identify interface variables
  2. Build FO jtree structures J for G
  3. Instantiate J_t
  4. Restore state description of interface variables from m_{t-1}
  5. Enter evidence $\boldsymbol{E}_t$ into J_t
  6. Pass messages in J_t
  7. Answer queries $\boldsymbol{Q_t}$
  8. Store state description of interface variables in m_t
  9. Proceed to next time step (step 3)

LDJT: Identify Interface Variables

Gehrke et al. (2018)

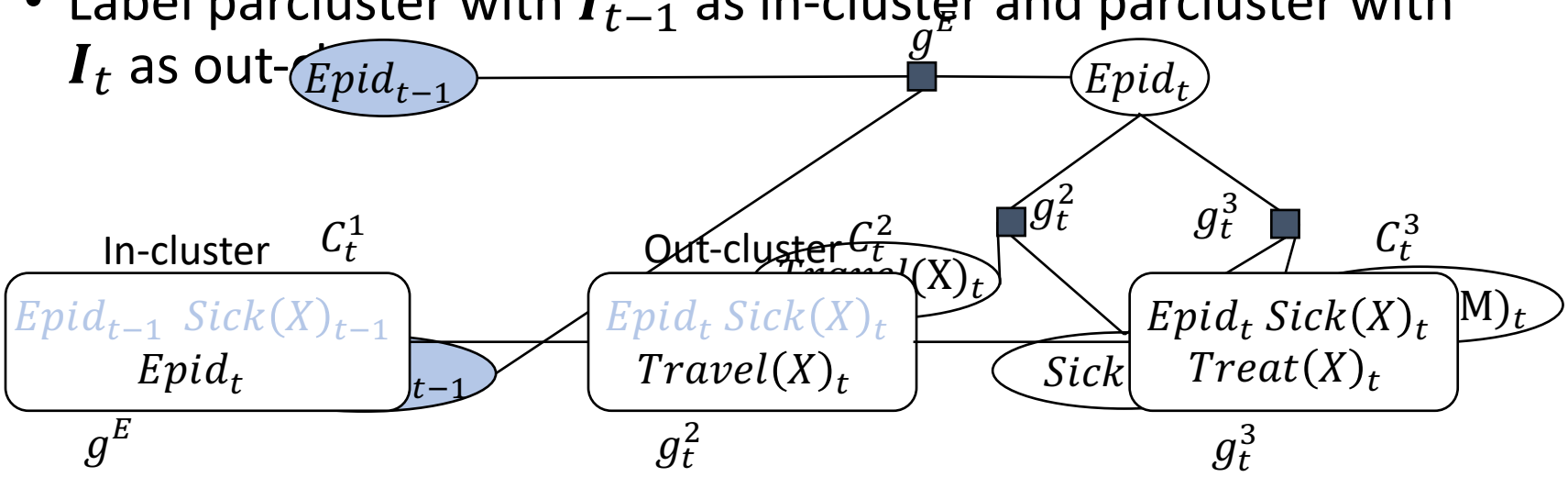
- $\boldsymbol{I}_{t-1} = \{A^i_{t-1} \mid \exists\, \phi(\mathcal{A})_{|C} \in G : A^i_{t-1} \in \mathcal{A} \ \wedge A^j_{t-1} \in \mathcal{A}\}$
- Set of interface variable $\boldsymbol{I}_{t-1}$ consists of all PRVs from time slice $t-1$ that occur in a parfactor with PRVs from time slice t



LDJT: Construct FO jtree Structure

Gehrke et al. (2018)

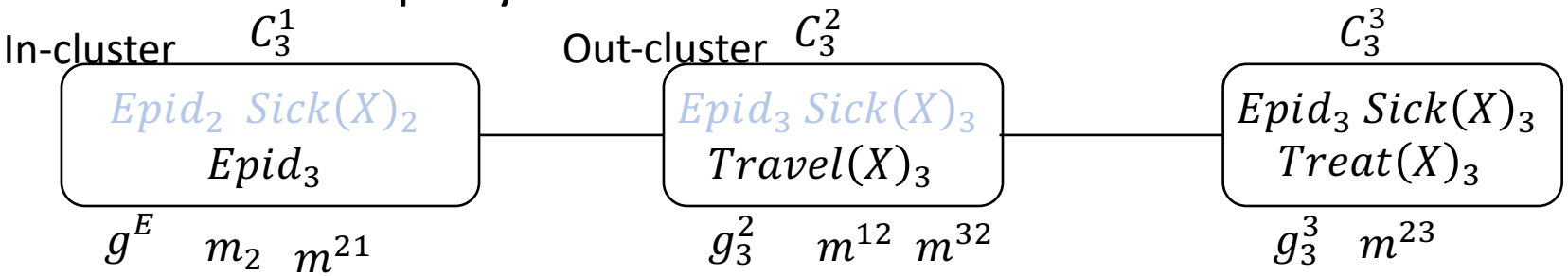
- Turn model in 1.5 time slice model
- Suffices to perform inference over time slice t
- From 1.5 time slice model construct FO jtree structure
- Ensure $\boldsymbol{I}_{t-1}$ is contained in a parcluster and $\boldsymbol{I}_t$ is contained in a parcluster
- Label parcluster with $\boldsymbol{I}_{t-1}$ as in-cluster and parcluster with $\boldsymbol{I}_t$ as out-cluster



LDJT: Query answering

Gehrke et al. (2018)

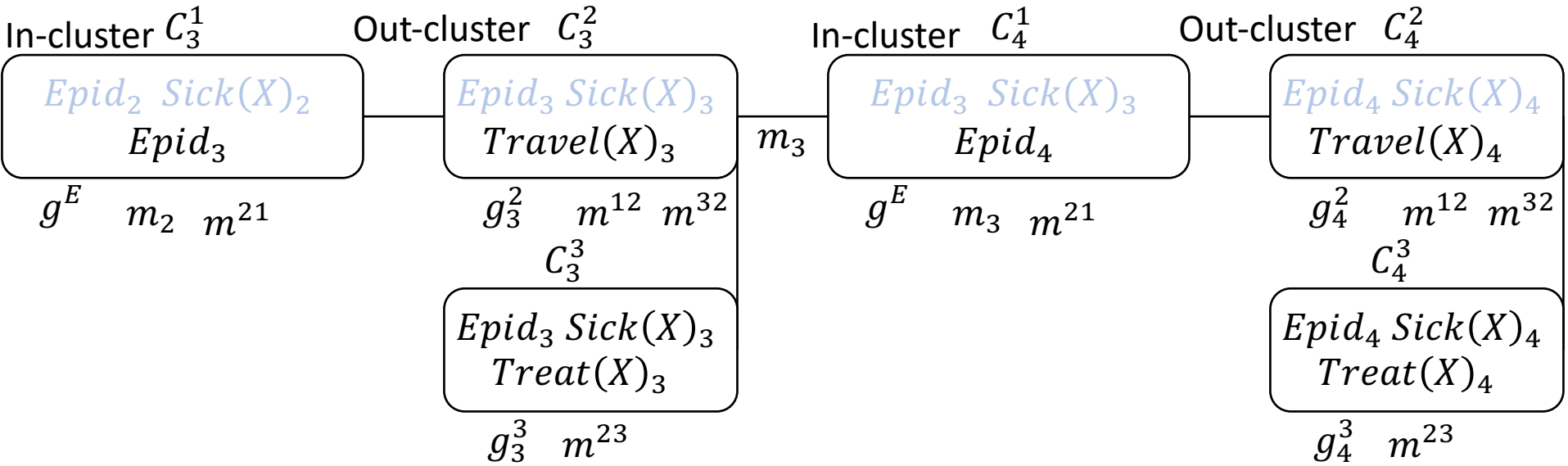
- Instantiate FO jtree structure
- Restore state description of interface variables
- Enter evidence
- Pass messages
- Query answering:
  - Find parcluster contain query term
  - Extract submodel
  - Answer query with LVE



LDJT: Proceed in time

Gehrke et al. (2018)

- Calculate m_3 using out-cluster (C_3^2)
- Eliminate $Travel(X)_3$ from C_3^2's local model
- Instantiate next FO jtree and enter m_3
- Enter evidence and pass messages



LDJT: Intermediate Overview

Gehrke et al. (2018)

- So far only a temporal forward pass
- Reason over one time step
- Keep only one time step in memory
- Filtering queries
- Prediction queries (filtering without new evidence)
- Hindsight queries

LDJT: Forward and Backward Pass

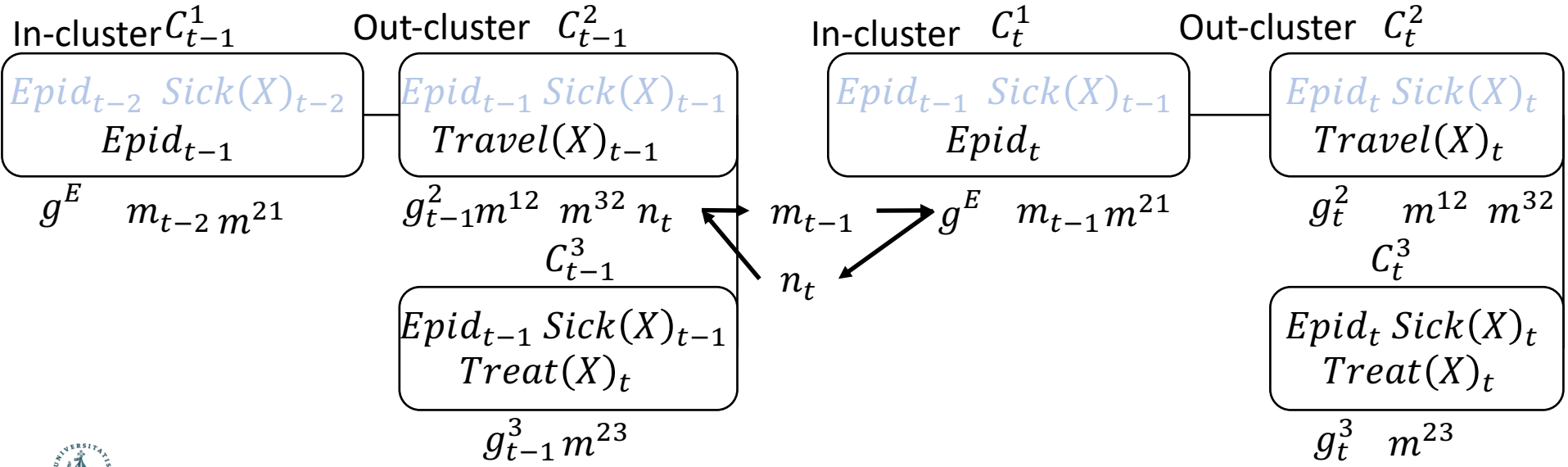
Gehrke et al. (2019)

- Use same FO jtree structures for backward pass
- Calculate a message n using an in-cluster over interface variables and pass n to previous time step
- LDJT needs to keep FO jtrees of previous time steps
- Different instantiation approaches during a backward pass
  - Keep all computations for all time steps in memory (not always feasible)
  - Instantiate time steps on demand (same as for the forward pass, possible due to the separation between time steps)

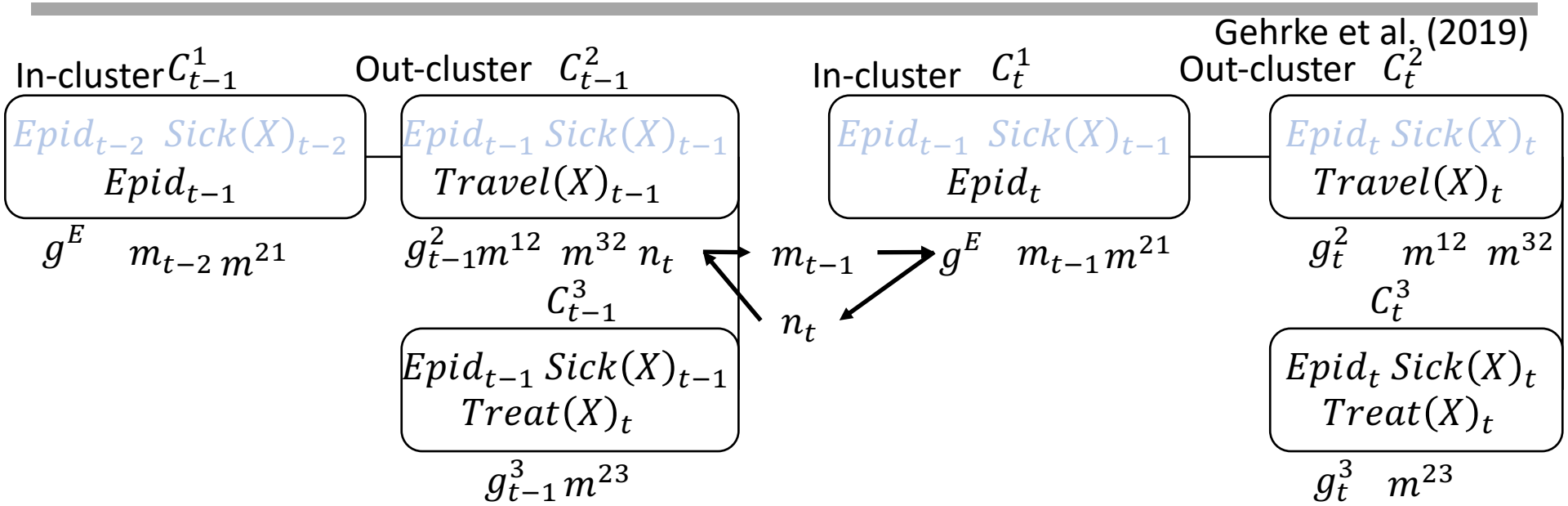
LDJT: Backward Pass

Gehrke et al. (2019)

- Calculate n_t using in-cluster (C_t^1)
- Eliminate $Epid_t$ from C_t^1's local model, without m_{t-1}
- Add n_t to local model of out-cluster C_{t-1}^2
- Pass messages for $t-1$ to account for n_t

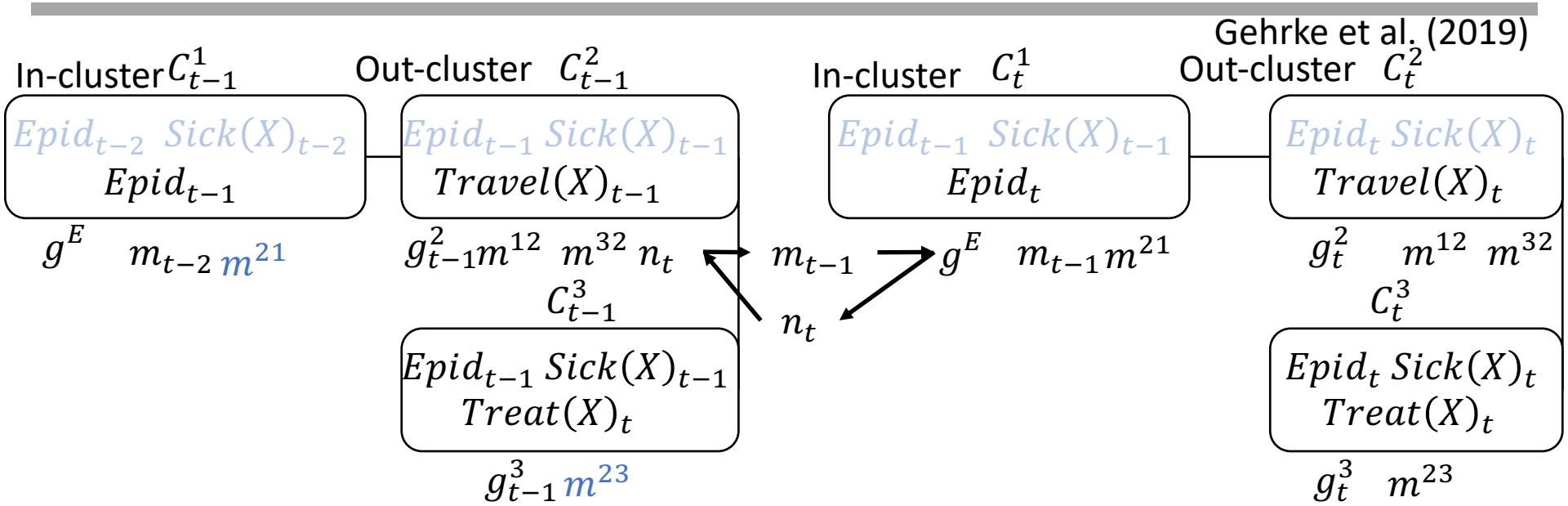


LDJT: Instantiations during a Backward Pass



	Keep Instantiations	Instantiate on demand
Messages to prepare for queries		
Messages to solely calculate n_{t-1}		
Additional memory for each time step		

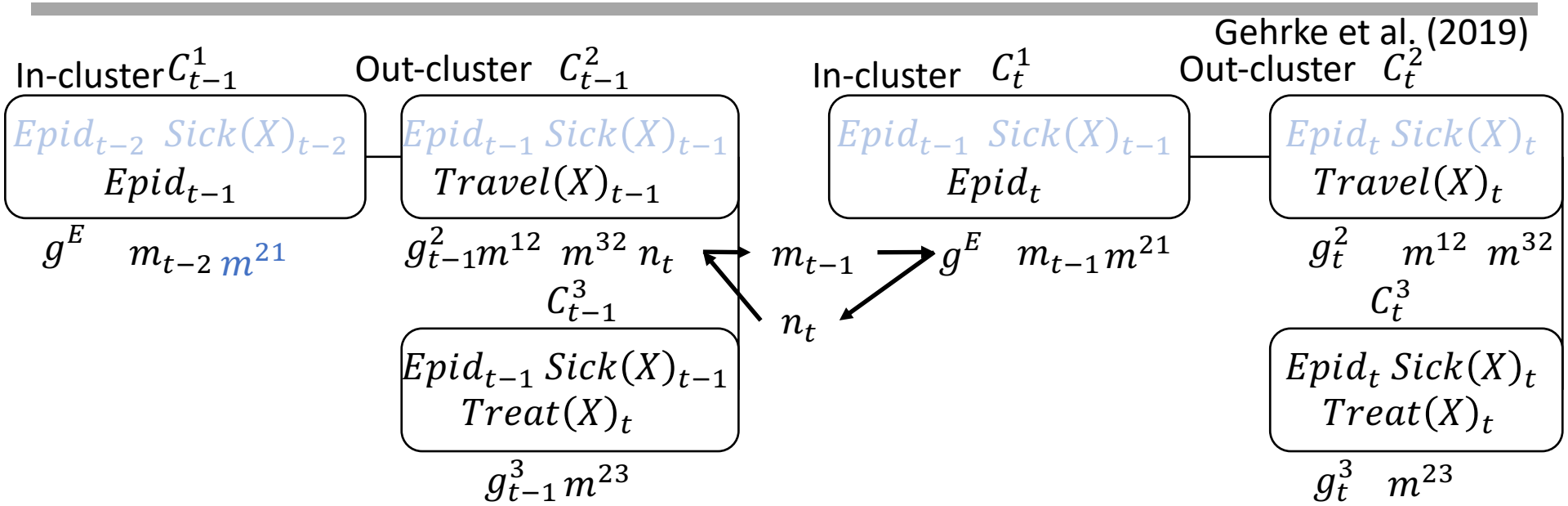
LDJT: Instantiations during a Backward Pass



	Keep Instantiations	Instantiate on demand
Messages to prepare for queries	$n-1$	
Messages to solely calculate n_{t-1}		
Additional memory for each time step		

n is the number of parclusters for each time step

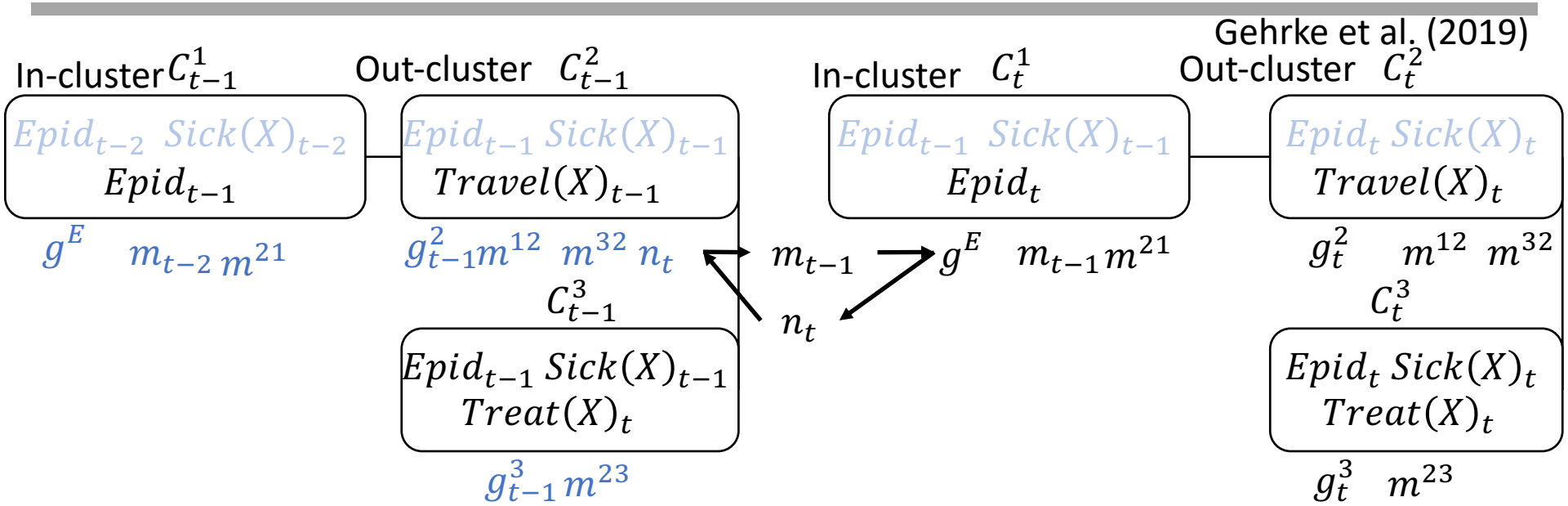
LDJT: Instantiations during a Backward Pass



	Keep Instantiations	Instantiate on demand
Messages to prepare for queries	$n-1$	
Messages to solely calculate n_{t-1}	$\leq n-1$	
Additional memory for each time step		

n is the number of parclusters for each time step

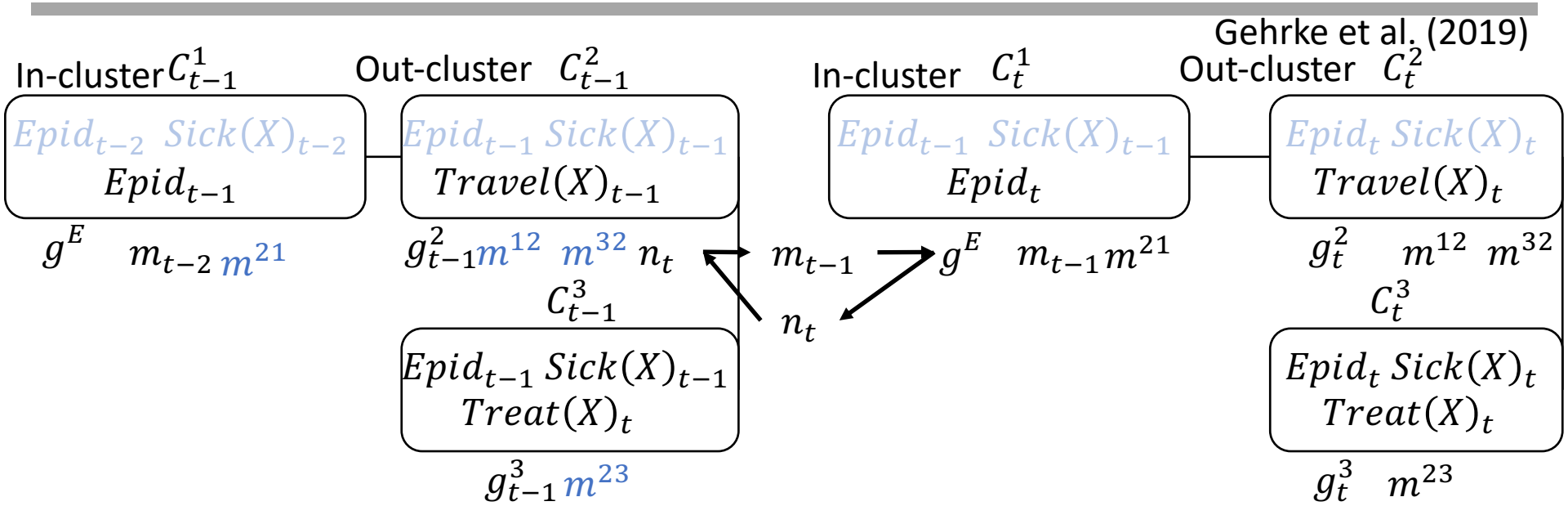
LDJT: Instantiations during a Backward Pass



	Keep Instantiations	Instantiate on demand
Messages to prepare for queries	$n-1$	
Messages to solely calculate n_{t-1}	$\leq n-1$	
Additional memory for each time step	All local models	

n is the number of parclusters for each time step

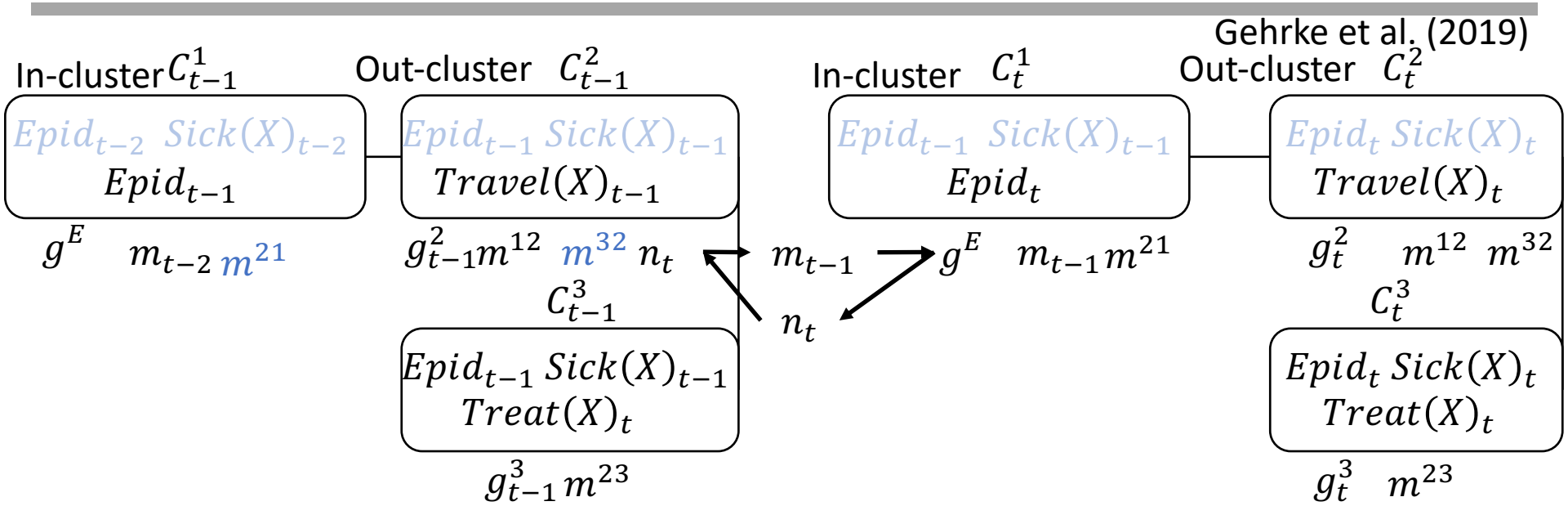
LDJT: Instantiations during a Backward Pass



	Keep Instantiations	Instantiate on demand
Messages to prepare for queries	$n-1$	$2*(n-1)$
Messages to solely calculate n_{t-1}	$\leq n-1$	
Additional memory for each time step	All local models	

n is the number of parclusters for each time step

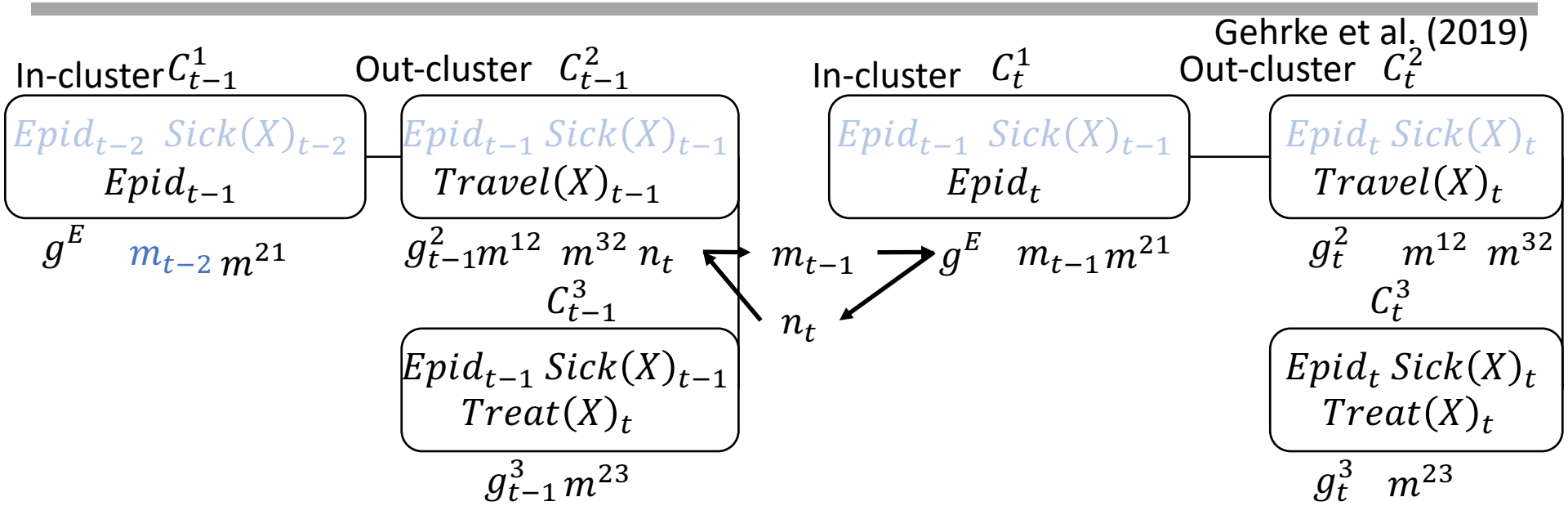
LDJT: Instantiations during a Backward Pass



	Keep Instantiations	Instantiate on demand
Messages to prepare for queries	$n-1$	$2*(n-1)$
Messages to solely calculate n_{t-1}	$\leq n-1$	$n-1$
Additional memory for each time step	All local models	

n is the number of parclusters for each time step

LDJT: Instantiations during a Backward Pass



	Keep Instantiations	Instantiate on demand
Messages to prepare for queries	$n-1$	$2*(n-1)$
Messages to solely calculate n_{t-1}	$\leq n-1$	$n-1$
Additional memory for each time step	All local models	Only forward (m_t) messages

n is the number of parclusters for each time step

LDJT: Relational Forward Backward Algorithm

Gehrke et al. (2019)

- LDJT can answer hindsight queries, even to the first time step
- By combining the instantiation approaches, LDJT can trade off memory consumption and reusing computations
- LDJT is in the worst case quadratic to T, but normally remains linear w.r.t. T (T max # time steps)
- But does it really suffice to lift the interface algorithm?

LDJT: Preventing Unnecessary Groundings

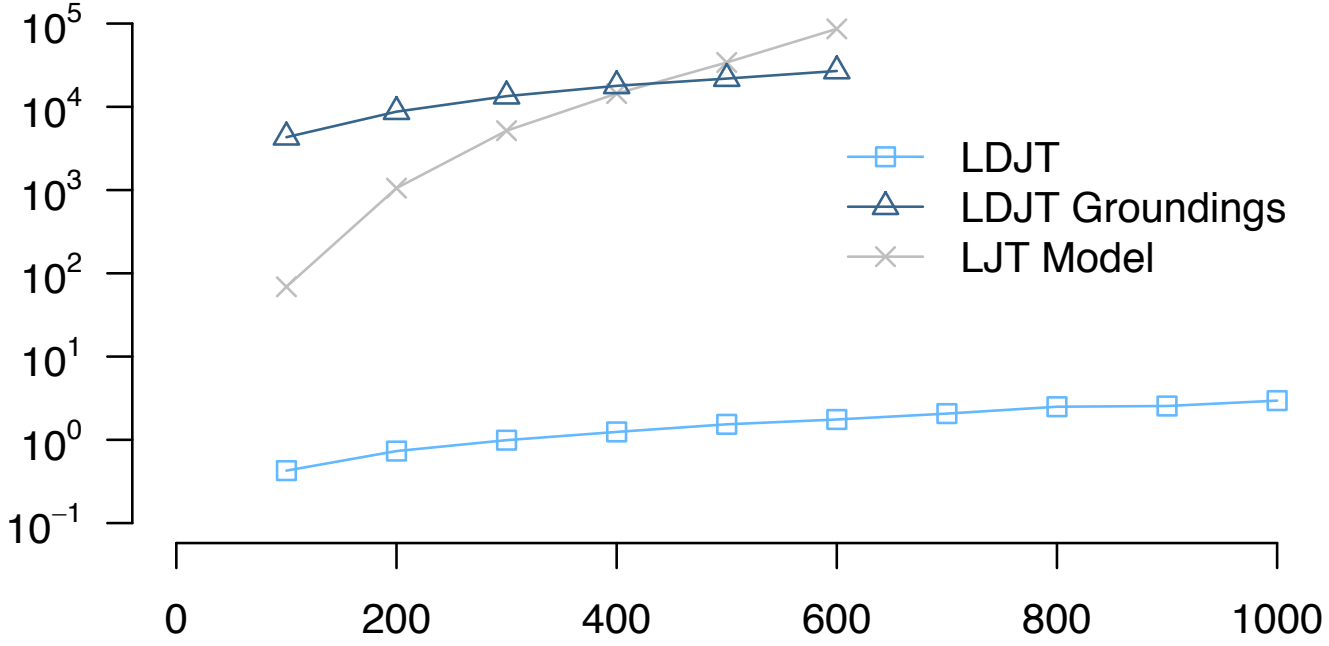
Gehrke et al. (2018b,c)

- Groundings in inter time slice messages (especially forward messages) can lead to grounding the model for all time steps
- Elimination order predetermined in FO jtree
- Non-ideal elimination order leads to groundings
  - Minimal set of interface variables not always ideal
  - Delay eliminations for inter time slice messages to prevent unnecessary groundings
  - Simply lifting the interface algorithm does not suffice, one also needs to ensure preconditions of lifting
- Trade off between lifting and handling temporal aspects due to restrictions on elimination orders

LDJT: Preventing Unnecessary Groundings

Gehrke et al. (2018b,c)

- Depending on the settings, either lifting or handling of temporal aspects is more efficient
- Preventing groundings to calculate a lifted solution pays off



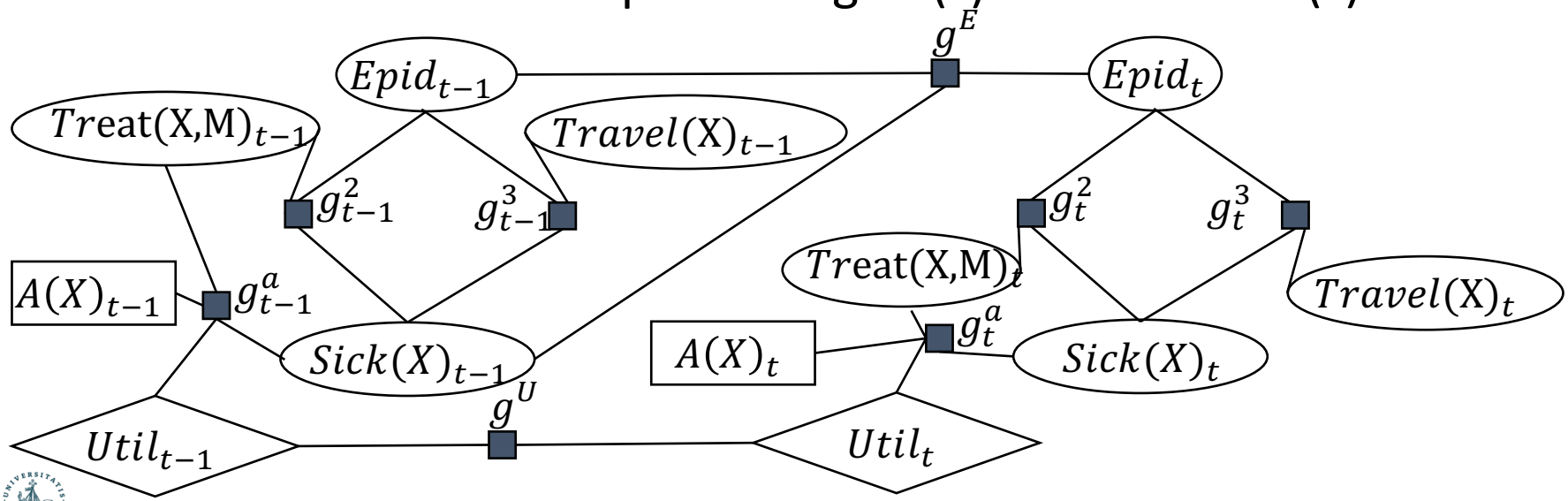
LDJT: Additional Queries

- Conjunctive queries over different time steps
  - Can be used for event detection
  - What is the probability that someone travelled from X to Y and that afterwards there is a epidemic in Y given there is an epidemic in X?
- Maximum expected utility
  - Decision support
  - Well studied within one time step (Apsel and Brafman (2011), Nath and Domingos (2009))
- Assignment queries
  - Most likely state sequence
  - Well studied for static models (Dawid (1992), Dechter (1999), de Salvo Braz et al. (2006), Apsel and Brafman (2012), Braun and Möller (2018))

LDJT: Maximum expected utility

Gehrke et al. (2018b,c)

- Extend representation with actions and utilities
- Problem: Find the action sequence that maximises the expected utility value w.r.t. a utility function.
- Only possible for a finite horizon
  - Combinatorial in horizon
  - Combinatorial in splits of logvar(s) of action PRV(s)



Outlook

- Continue optimising
  - Parallelisation
  - Caching
- From discrete time interval to time continuous
- Preserving symmetries
- Learning?
  - Structure
  - Potentials (Idea of Baum Welch now possible)
  - Symmetries
  - *Transfer learning*
- Open world?
  - Unknown domains
  - Unknown behaviour

Wrap-up Exact Lifted Dynamic Inference

- Parfactor models for sparse encoding
  - Factorisation of full joint distribution
  - Logical variables to model objects
- Algorithms for exact query answering
  - LDJT for repeated inference
  - Extensions possible
    - Parameterised, conjunctive queries
    - Maximum expected utility
    - Assignment queries (Tomorrow)

Mission and Schedule of the Tutorial*

Providing an introduction into inference in StaRAI

- Introduction — 20 min
  - StaR AI
- Overview: Probabilistic relational modeling — 30 min
  - Semantics (grounded-distributional, maximum entropy)
  - Inference problems and their applications
  - Algorithms and systems
- Scalable static inference — 40 + 30 min
  - Exact propositional inference
  - Exact lifted inference
- Scalable dynamic inference — 50 min
  - Exact propositional inference
  - Exact lifted inference
- Summary — 10 min

*Thank you to the SRL/StaRAI crowd for all their exciting contributions! The tutorial is necessarily incomplete. Apologies to anyone whose work is not cited

References

- Apsel and Brafman (2012)
  Udi Apsel and Ronen I. Brafman. Exploiting Uniform Assignments in First-Order MPE. *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, 2012.
- Apsel and Brafman (2011)
  Udi Apsel and Ronen I. Brafman. Extended Lifted Inference with Joint Formulas. Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence. pp. 11–18, 2011.
- Dawid (1992)
  Alexander Philip Dawid. Applications of a General Propagation Algorithm for Probabilistic Expert Systems. *Statistics and Computing*, 2(1):25–36, 1992.
- Dechter (1999)
  Rina Dechter. Bucket Elimination: A Unifying Framework for Probabilistic Inference. In *Learning and Inference in Graphical Models*, pages 75–104. MIT Press, 1999.

References

- De Salvo Braz et al. (2006)
  Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. MPE and Partial Inversion in Lifted Probabilistic Variable Elimination. *AAAI-06 Proceedings of the 21st Conference on Artificial Intelligence*, 2006.
- Murphy (2002)
  Kevin P. Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning. *PhD Thesis University of California, Berkeley*, 2002.
- Nath and Domingos (2009)
  Aniruddh Nath and Pedro Domingos, A language for relational decision theory, Proceedings of the International Workshop on Statistical Relational Learning, 2009.

Work @ IFIS

- Braun and Möller (2018b)

  Tanya Braun and Ralf Möller. Lifted Most Probable Explanation. In *Proceedings of the International Conference on Conceptual Structures*, 2018.

- Gehrke et al. (2018)

  Marcel Gehrke, Tanya Braun, and Ralf Möller. Lifted Dynamic Junction Tree Algorithm. In *Proceedings of the International Conference on Conceptual Structures*, 2018.

- Gehrke et al. (2018b)

  Marcel Gehrke, Tanya Braun, and Ralf Möller. Towards Preventing Unnecessary Groundings in the Lifted Dynamic Junction Tree Algorithm. In Proceedings of KI 2018: Advances in Artificial Intelligence, 2018.

- Gehrke et al. (2018c)

  Marcel Gehrke, Tanya Braun, and Ralf Möller. Preventing Unnecessary Groundings in the Lifted Dynamic Junction Tree Algorithm. In Proceedings of the AI 2018: Advances in Artificial Intelligence, 2018.

Work @ IFIS

- Gehrke et al. (2019)

  Marcel Gehrke, Tanya Braun, and Ralf Möller. Relational Forward Backward Algorithm for Multiple Queries. In *FLAIRS-32 Proceedings of the 32nd International Florida Artificial Intelligence Research Society Conference*, 2019.

- Gehrke et al. (2019b)

  Marcel Gehrke, Tanya Braun, Ralf Möller, Alexander Waschkau, Christoph Strumann, and Jost Steinhäuser. Lifted Maximum Expected Utility. In Artificial Intelligence in Health, 2019.

- Gehrke et al. (2019c)

  Marcel Gehrke, Tanya Braun, and Ralf Möller. Lifted Temporal Maximum Expected Utility. In Proceedings of the 32nd Canadian Conference on Artificial Intelligence, Canadian AI 2019, 2019.