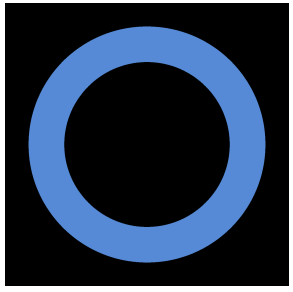
Tractable Uncertainty for Structure Learning

Benjie Wang, Matthew Wicker, Marta Kwiatkowska

ICML 2022

Motivation: Uncertainty in Causal Structures



Diabetes

Fludeoxyglucose

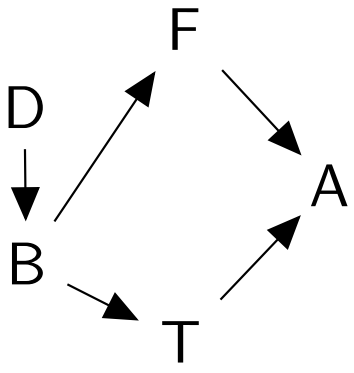
[*Shen et al. 2020*]

Alzheimer's

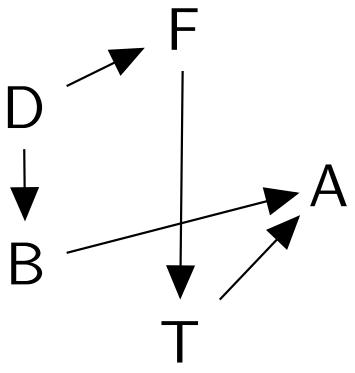
Amyloid **B**eta

Phosphorylated **T**au

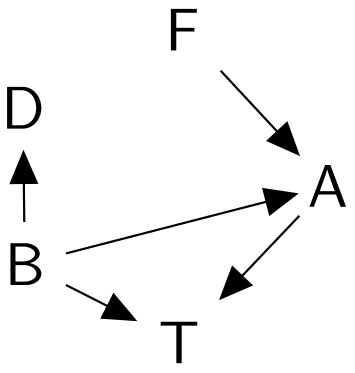
Motivation: Uncertainty in Causal Structures



$$CE(D,A) = b_{DB}b_{BF}b_{FA} + b_{DB}b_{BT}b_{TA}$$

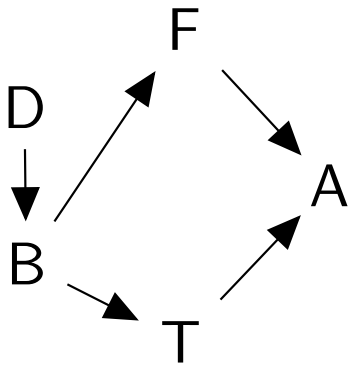


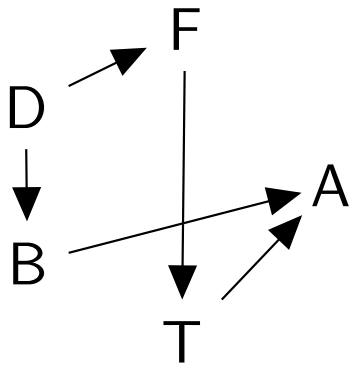
$$CE(D,A) = b_{DB}b_{BA} + b_{DF}b_{FT}b_{TA}$$

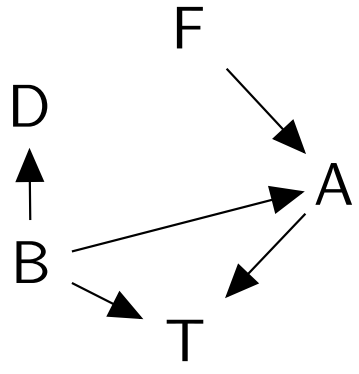


$$CE(D,A) = 0$$

Motivation: Uncertainty in Causal Structures







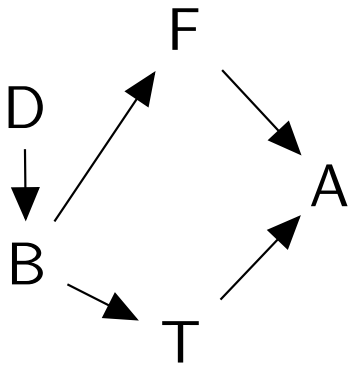
$$CE(D,A) = b_{DB}b_{BF}b_{FA} + b_{DB}b_{BT}b_{TA}$$

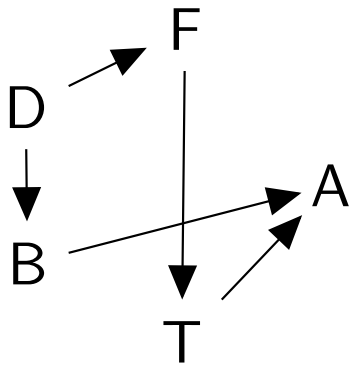
$$CE(D,A) = b_{DB}b_{BA} + b_{DF}b_{FT}b_{TA}$$

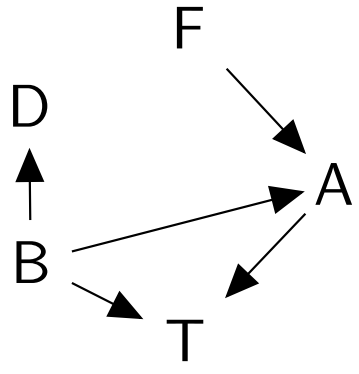
$$CE(D,A) = 0$$

- What is the **probability** that Diabetes causes Amyloid Beta deposition?

Motivation: Uncertainty in Causal Structures







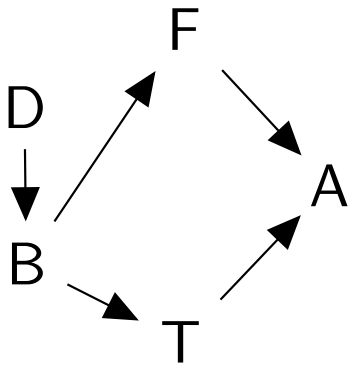
$$CE(D,A) = b_{DB}b_{BF}b_{FA} + b_{DB}b_{BT}b_{TA}$$

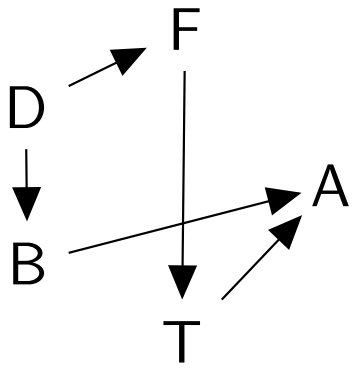
$$CE(D,A) = b_{DB}b_{BA} + b_{DF}b_{FT}b_{TA}$$

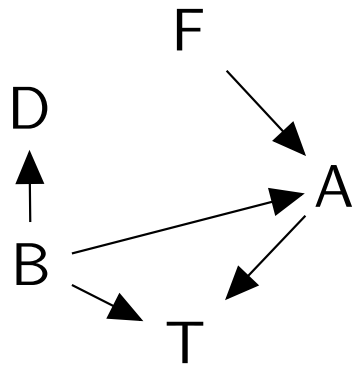
$$CE(D,A) = 0$$

- What is the **probability** that Diabetes causes Amyloid Beta deposition?
- What is the **expected** causal effect of Diabetes on Alzheimer's?

Motivation: Uncertainty in Causal Structures







$$CE(D,A) = b_{DB}b_{BF}b_{FA} + b_{DB}b_{BT}b_{TA}$$

$$CE(D,A) = b_{DB}b_{BA} + b_{DF}b_{FT}b_{TA}$$

$$CE(D,A) = 0$$

- What is the **probability** that Diabetes causes Amyloid Beta deposition?
- What is the **expected** causal effect of Diabetes on Alzheimer's?
- **Given that** Diabetes causes Amyloid Beta deposition, what is the expected causal effect?

Bayesian Structure Learning

Model Express uncertainty using prior knowledge and data D:

$$p(G|D) \quad p(D|G)p(G)$$

Bayesian Structure Learning

Model Express uncertainty using prior knowledge and data D:

$$p(G|D) \quad p(D|G)p(G)$$

Goal Answer some query, typically of the form $\mathrm{E}_{p(G|D)}[f(G)]$

Bayesian Structure Learning

Model Express uncertainty using prior knowledge and data D:

$$p(G|D) \quad p(D|G)p(G)$$

Goal Answer some query, typically of the form $\mathrm{E}_{p(G|D)}[f(G)]$

Approximation Derive some approximation $q_\varphi(G) \approx p(G|D)$, and use q to estimate the query.

Bayesian Structure Learning

Model Express uncertainty using prior knowledge and data D:

$$p(G|D) \quad p(D|G)p(G)$$

Goal Answer some query, typically of the form $\mathrm{E}_{p(G|D)}[f(G)]$

Approximation Derive some approximation $q_\varphi(G) \approx p(G|D)$, and use q to estimate the query.

- **Expressive** family of distributions over **acyclic** directed graphs G

Bayesian Structure Learning

Model Express uncertainty using prior knowledge and data D:

$$p(G|D) \quad p(D|G)p(G)$$

Goal Answer some query, typically of the form $\mathrm{E}_{p(G|D)}[f(G)]$

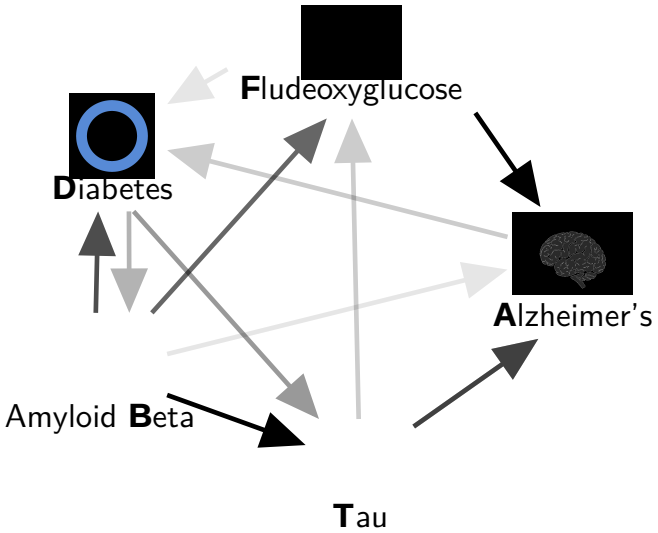
Approximation Derive some approximation $q_\varphi(G) \approx p(G|D)$, and use q to estimate the query.

- **Expressive** family of distributions over **acyclic** directed graphs G
- **Tractable** to answer the queries of interest

How do we encode acyclicity?

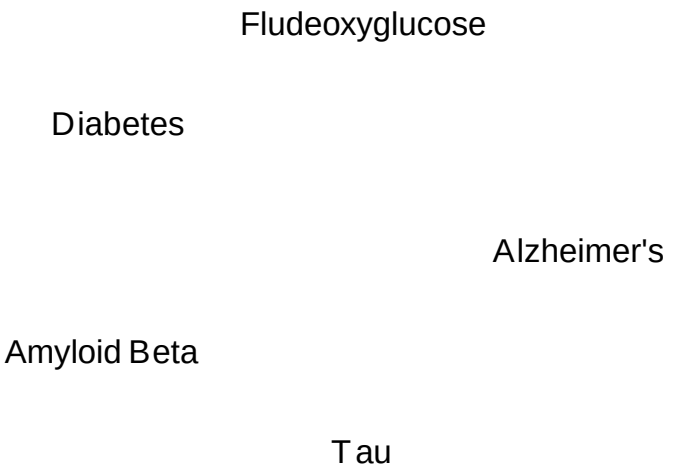
Distributions over Directed Graphs

- **Mean-field**: $q_\varphi(G) \quad \square_{i,j=1}^{d} \text{Bernoulli}(G_{ij}; \varphi_{ij}))$



Distributions over Directed Graphs

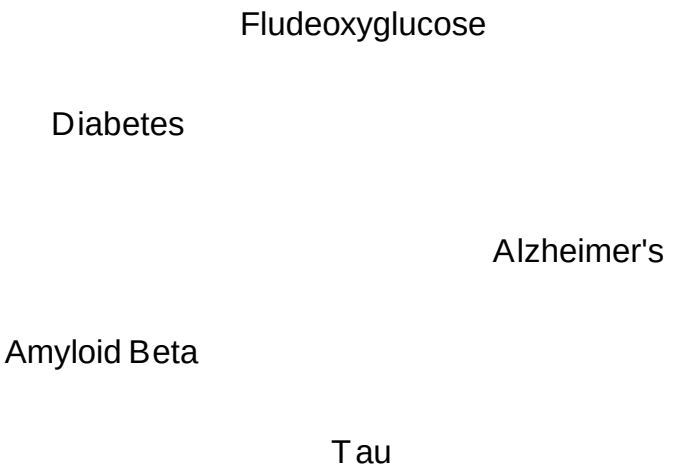
- Mean-field: $q(G) / \prod_{i,j=1}^{d} \text{Bernoulli}(G_{ij}; \theta_{ij})$



- Does not consider correlations due to acyclicity;

Distributions over Directed Graphs

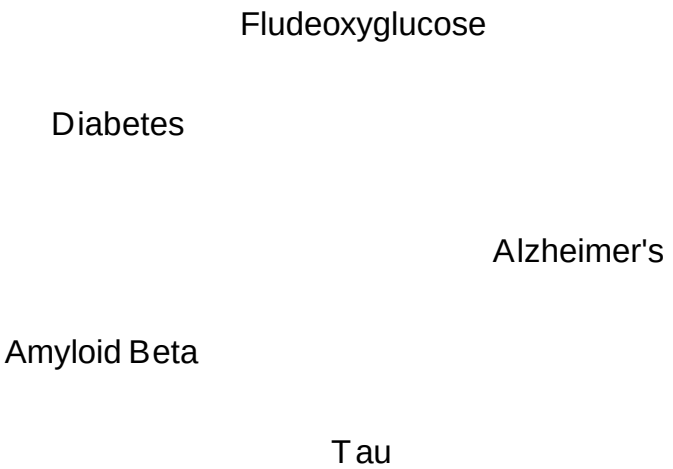
- Mean-field: $q(G) \propto \prod_{i,j=1}^{d} \text{Bernoulli}(G_{ij}; \theta_{ij})$



- Does not consider correlations due to acyclicity;
- Not very expressive;

Distributions over Directed Graphs

- Mean-field: $q(G) \propto \prod_{i,j=1}^{d} \text{Bernoulli}(G_{ij}; \theta_{ij})$

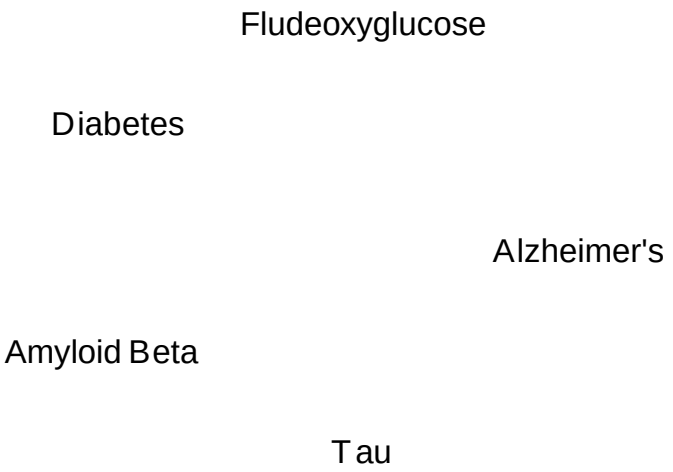


  - Does not consider correlations due to acyclicity;
  - Not very expressive;

- Neural Autoregressive: $q(G) \propto \prod_{i,j=1}^{d} q_{\theta_{ij}}(G_{ij} | G_{<ij})$

Distributions over Directed Graphs

- Mean-field: $q(G) = \prod_{i,j=1}^{d} \text{Bernoulli}(G_{ij}; \theta_{ij})$

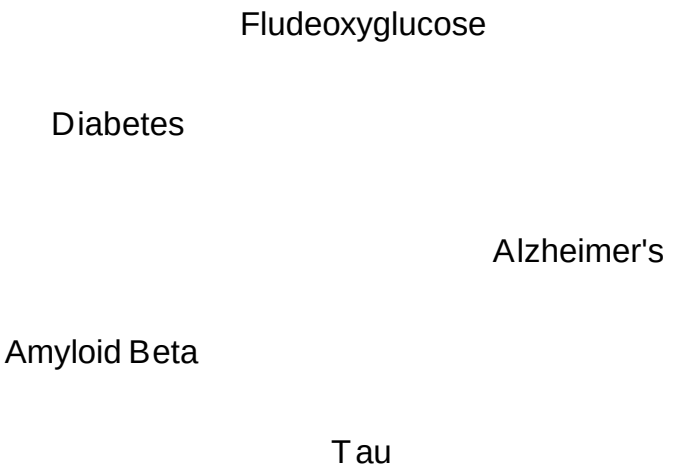


  - Does not consider correlations due to acyclicity;
  - Not very expressive;

- Neural Autoregressive: $q(G) = \prod_{i,j=1}^{d} q_{\theta_{ij}}(G_{ij} | G_{<ij})$
  - Difficult to train to encode acyclicity;

Distributions over Directed Graphs

- Mean-field: $q(G) \propto \prod_{i,j=1}^{d} \text{Bernoulli}(G_{ij}; \theta_{ij})$



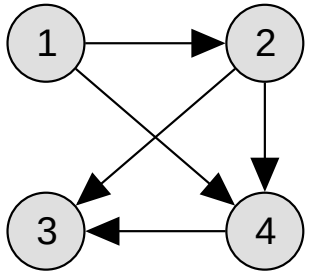
  - Does not consider correlations due to acyclicity;
  - Not very expressive;

- Neural Autoregressive: $q(G) \propto \prod_{i,j=1}^{d} q_{\theta_{ij}}(G_{ij} \mid G_{<ij})$
  - Difficult to train to encode acyclicity;
  - Intractable (except for sampling);

DAG Distribution using Tractable Circuits

Orderings We work on the joint space of topological ordersand directed graphsG:

$= f 1; 2; 4; 3g$

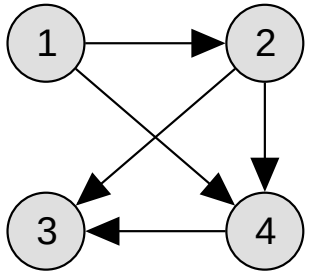


- Every DAG is consistent with at least one order;
- Every directed graph consistent with an order is acyclic;

DAG Distribution using Tractable Circuits

Orderings We work on the joint space of topological ordersand directed graphsG:

$= f 1; 2; 4; 3g$



- Every DAG is consistent with at least one order;
- Every directed graph consistent with an order is acyclic;

Solution We introduce a parameterized distribution family $q_{\theta}(\ ;\ G)$ for orders and graphs based on tractable probabilistic circuits.

Sum-Product Networks

Sum-Product Networks (SPNs) are a type of tractable probabilistic model for expressing a distribution over a set of variables X.

Sum-Product Networks

Sum-Product Networks (SPNs) are a type of tractable probabilistic model for expressing a distribution over a set of variables X.

SPNs are rooted DAGs consisting of three types of nodes:

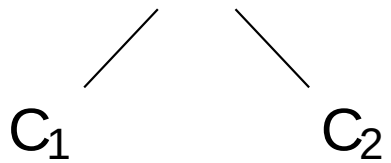
- L: Simple base distributions $L(X)$

Sum-Product Networks

Sum-Product Networks (SPNs) are a type of tractable probabilistic model for expressing a distribution over a set of variables X.

SPNs are rooted DAGs consisting of three types of nodes:

- L: Simple base distributions $L(X)$
- : Factorize distributions, $P(X) = C_1(X_1) \quad C_2(X_2)$

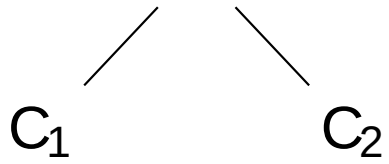


Sum-Product Networks

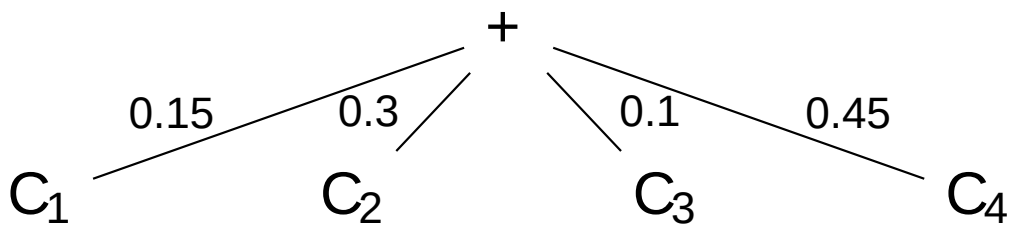
Sum-Product Networks (SPNs) are a type of tractable probabilistic model for expressing a distribution over a set of variables X.

SPNs are rooted DAGs consisting of three types of nodes:

- L: Simple base distributions $L(X)$
- $\times$: Factorize distributions, $P(X) = C_1(X_1) \times C_2(X_2)$



- +: Mix component distributions, $S(X) = \sum_j \lambda_j C_j(X)$



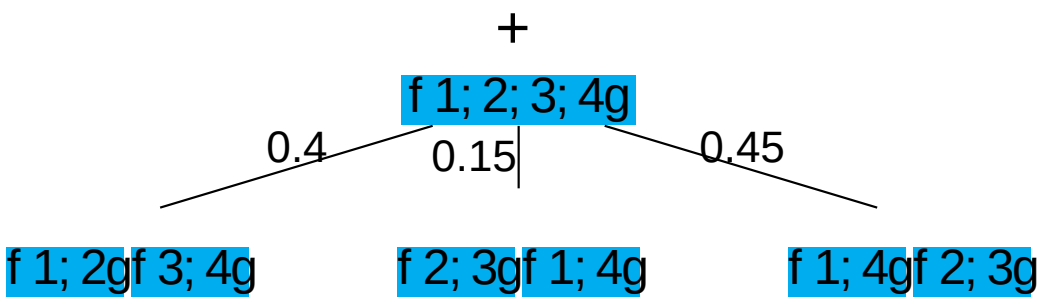
OrderSPNs

We introduce OrderSPNs, SPNs which express distributions over orderings of a set $\{1, \ldots, d\}$.

OrderSPNs

We introduce OrderSPNs, SPNs which express distributions over orderings of a set $\{1, \ldots, d\}$.

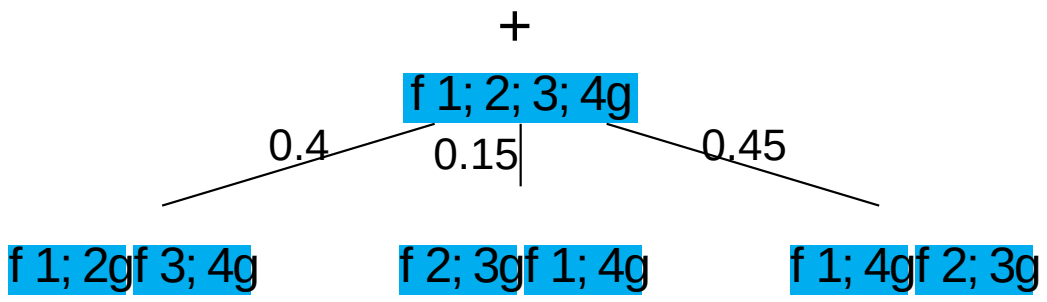
- $+$: Mix different partitions of the order $= (\ _1; \ _2)$



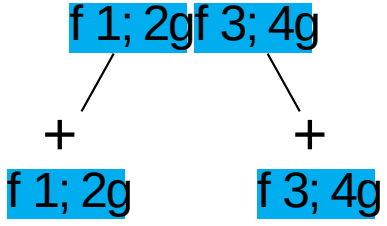
OrderSPNs

We introduce OrderSPNs, SPNs which express distributions over orderings of a set $\{1, \ldots, d\}$.

- $+$: Mix different partitions of the order $\sigma = (\sigma_1, \sigma_2)$



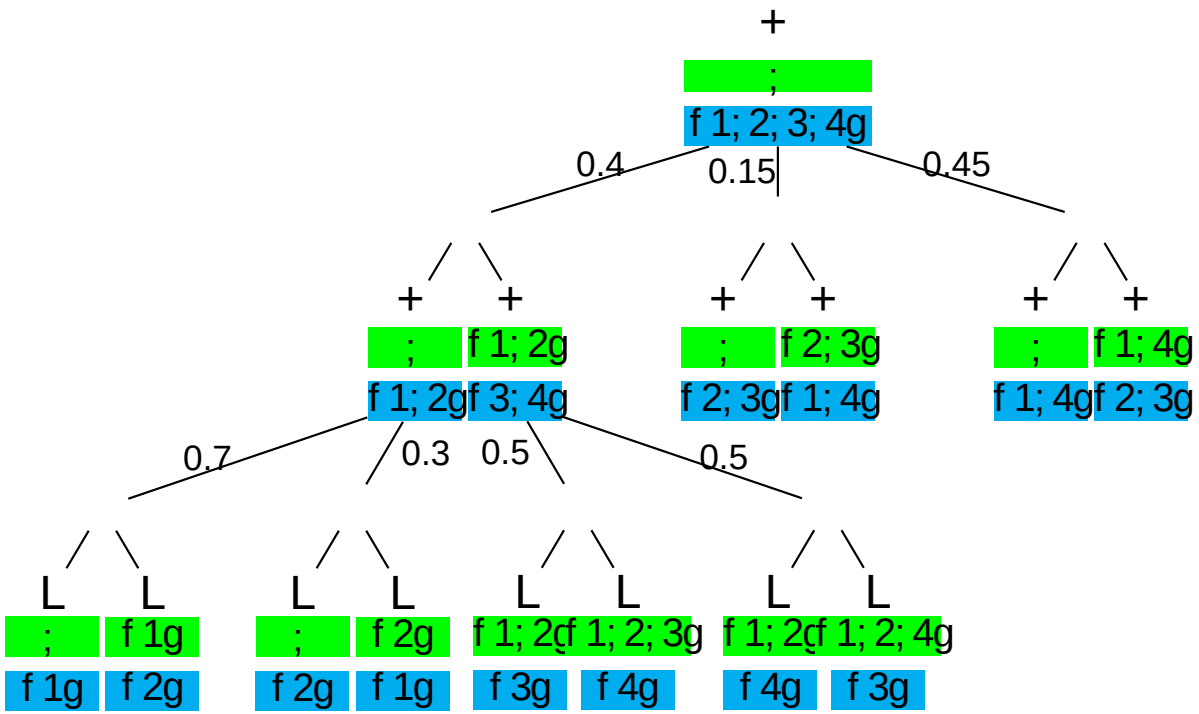
- $\times$: Factorize into independent $P(\sigma) = C_1(\sigma_1) \times C_2(\sigma_2)$



Note that the order of the children of a product node does matter!

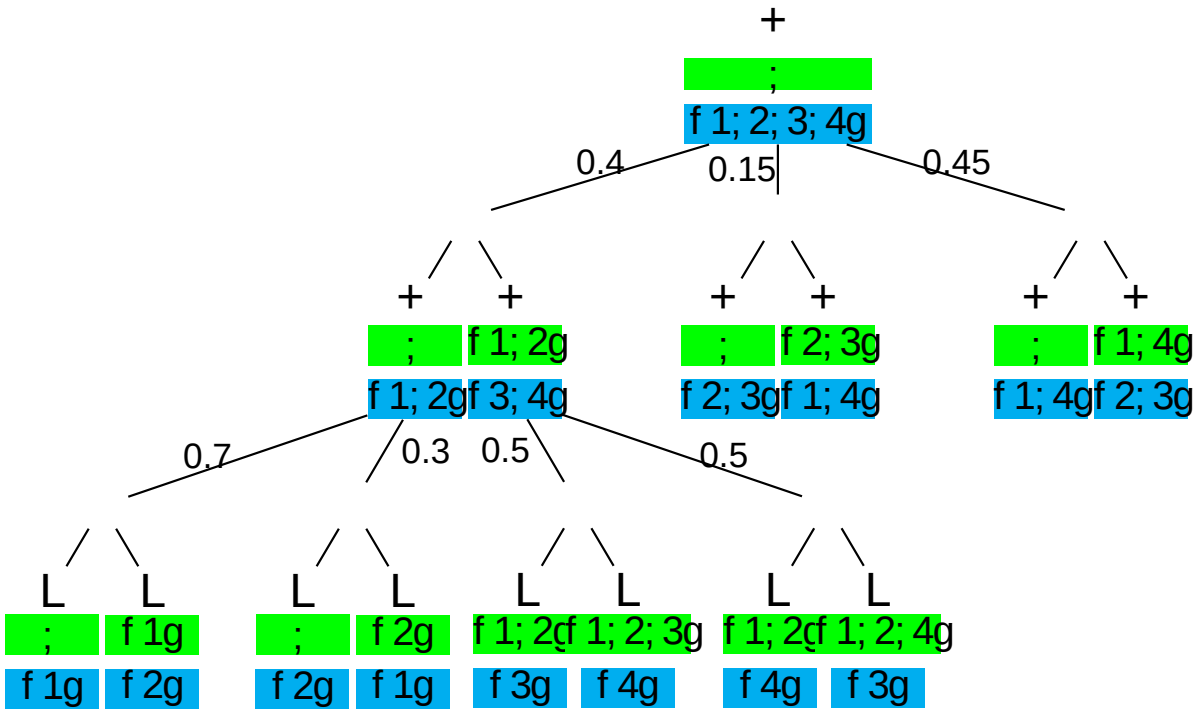
OrderSPNs

Alternate sum and product layers until order is fully determined:



OrderSPNs

Alternate sum and product layers until order is fully determined:



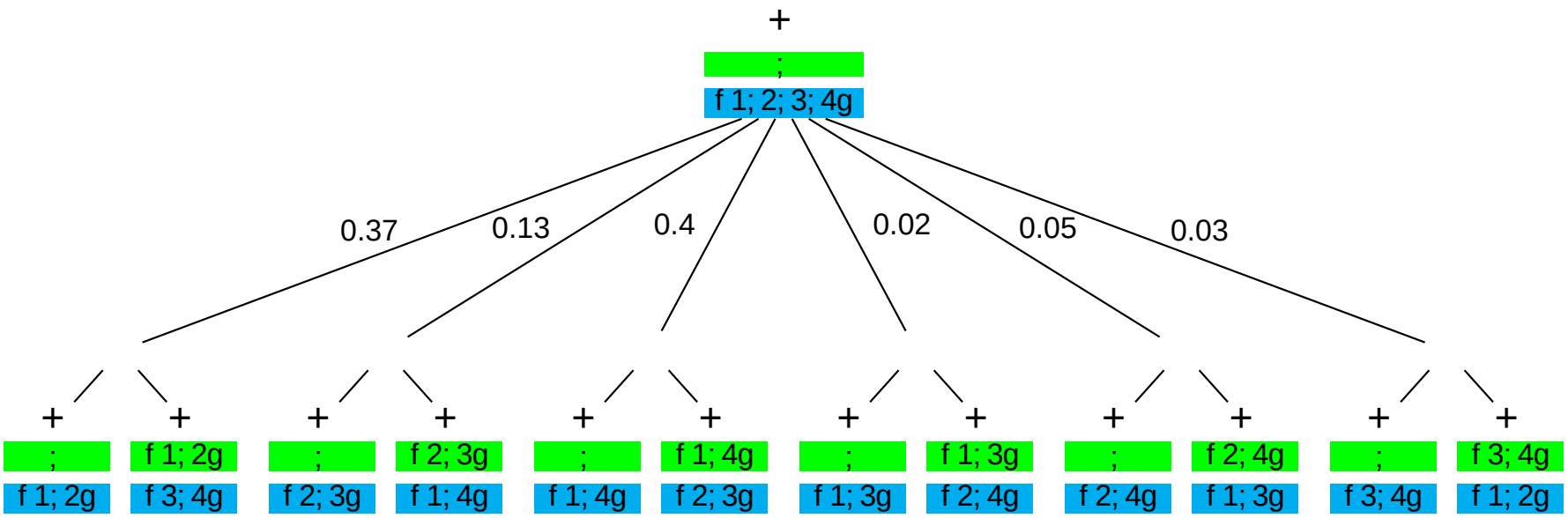
How does this relate to DAGs?

- L: (S, f ig) indicates that S precedes $_i$ in the ordering; thus $L(G_i) = 0$ if $G_i * S$, where G_i is the set of parents of node $_i$

Are OrderSPNs a good approximation to the tru
posterior?

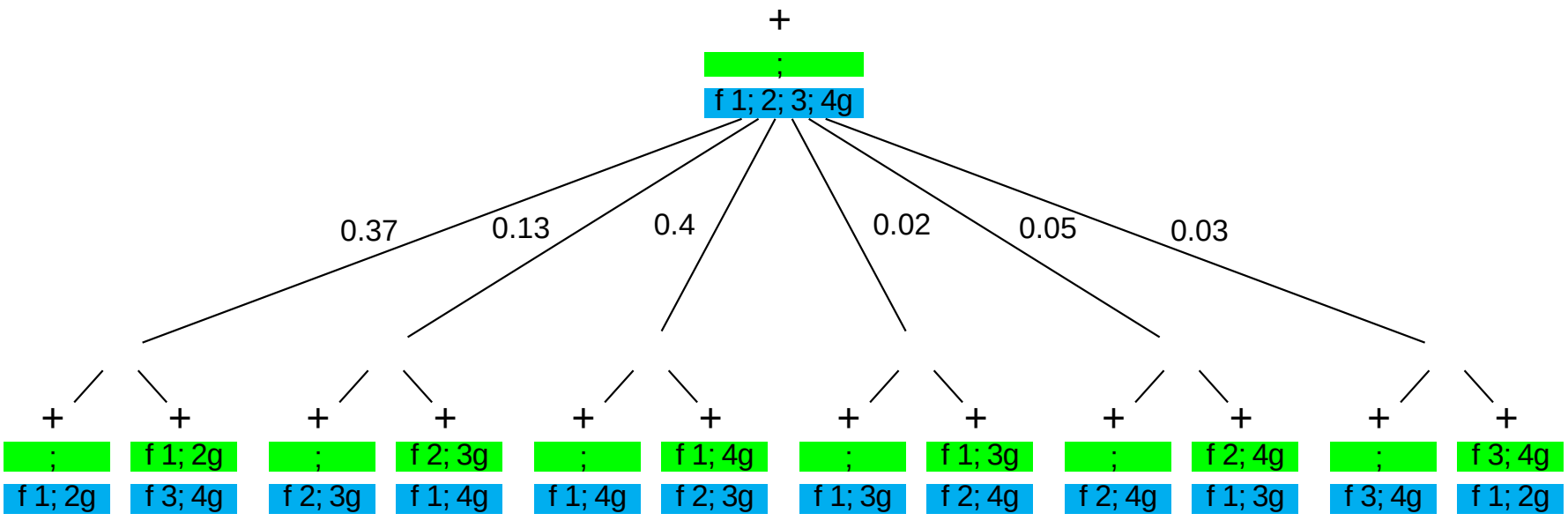
Natural Approximation

OrderSPNs can be viewed as a hierarchical, width-limited approximation to the true posterior.



Natural Approximation

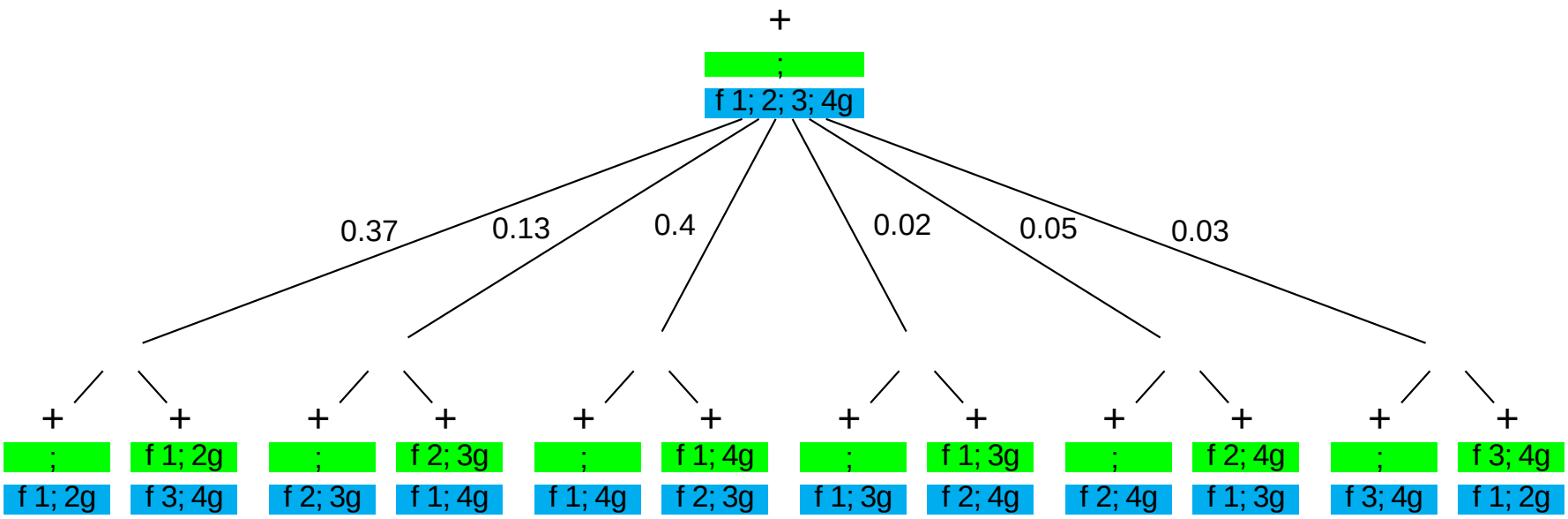
OrderSPNs can be viewed as a hierarchical, width-limited approximation to the true posterior.



- At +-nodes, select the active branches (partitions) using e cient heuristic subroutines.

Natural Approximation

OrderSPNs can be viewed as a hierarchical, width-limited approximation to the true posterior.



- At +-nodes, select the active branches (partitions) using e cient heuristic subroutines.
- -nodes encode exact conditional independences in the posterior.

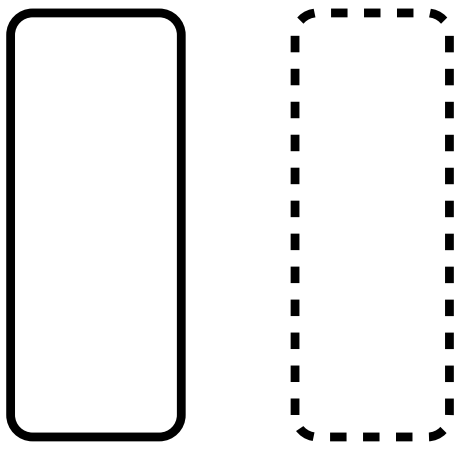
OrderSPNs: Coverage

Proposition

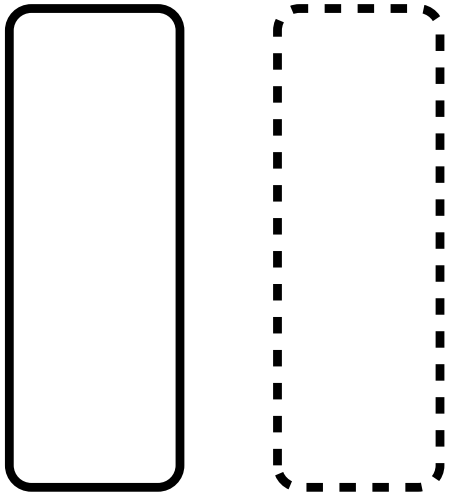
OrderSPNs can be exponentially more compact than a tabular representation of orders/DAGs.

OrderSPNs: Empirical Analysis

Even if one chooses the partitions randomly, and only learns the weights of the OrderSPN, it can outperform baselines on some metrics.



Expected-SHD: Lower is better



AUROC: Higher is better

The Bene ts of Tractability

Tractable Queries

The tractability of SPNs depends on their structural properties.

Tractable Queries

The tractability of SPNs depends on their structural properties.

Proposition

Regular OrderSPNs are complete and decomposable, and deterministic .

Tractable Queries

The tractability of SPNs depends on their structural properties.

Proposition

Regular OrderSPNs are complete and decomposable, and deterministic .

	Sampling	Marginals	Most Likely	ELBO	Causal E ect
Mean- eld	3	3	3	7	7
Autoregressive	3	7	7	7	7
EBM	7	7	7	7	7
OrderSPN	3 $O(d^2)$	3 $O(M)$	3 $O(M)$	3 $O(M)$	3 $O(d^3M)$

Learning OrderSPN Weights

Variational inference is used to optimize the parameters:

$$\mathrm{ELBO} = \mathrm{E}_{q\ (G)}[\log p(G|D)] + \mathrm{H}(q\ (G))$$

- For existing variational families, this has to be estimated through sampling and/or continuous relaxation

Learning OrderSPN Weights

Variational inference is used to optimize the parameters:

$$\mathrm{ELBO} = \mathrm{E}_{q\ (G)}[\log p(G|D)] + \mathrm{H}(q\ (G))$$

- For existing variational families, this has to be estimated through sampling and/or continuous relaxation

Proposition

The ELBO and its gradients for any regular OrderSPN and modular distribution p can be computed exactly in linear time in the size of the SPN.

- Eliminates variance in the high-dimensional, discrete space of graphs G, leading to stable optimization.

Query Answering

Given approximate posterior q, we want to be able to extract information about the system.

Let $\bigvee_i c_i$ be some feature of the causal graph, e.g. a set of edges.

Query Answering

Given approximate posterior q, we want to be able to extract information about the system.

Let $\bigvee_i c_i$ be some feature of the causal graph, e.g. a set of edges.

- Sampling: $\text{Sample}G \quad q(; G_j \bigvee_i c_i)$;

Query Answering

Given approximate posterior q, we want to be able to extract information about the system.

Let $\bigvee_i c_i$ be some feature of the causal graph, e.g. a set of edges.

- Sampling: $\text{SampleG} \sim q(\cdot\,; G \mid \bigvee_i c_i)$;
- Marginals: Evaluate $q(\bigvee_i c_i^0 \mid \bigvee_i c_i)$;

Query Answering

Given approximate posterior q, we want to be able to extract information about the system.

Let $\bigvee_i c_i$ be some feature of the causal graph, e.g. a set of edges.

- Sampling: Sample $G \sim q(\cdot; G) \models \bigvee_i c_i$;
- Marginals: Evaluate $q(\bigvee_i c_i' \mid \bigvee_i c_i)$;
- Most Likely: Evaluate $\max_G q(\cdot; G) \models \bigvee_i c_i$;

Query Answering

Given approximate posterior q, we want to be able to extract information about the system.

Let $\bigvee_i c_i$ be some feature of the causal graph, e.g. a set of edges.

- Sampling: $\text{SampleG} \sim q(\cdot; G | \bigvee_i c_i)$;
- Marginals: $\text{Evaluate} q(\bigvee_i c_i | \bigvee_i c_i)$;
- Most Likely: $\text{Evaluate} \max_G q(\cdot; G | \bigvee_i c_i)$;
- Linear Causal Effects: $\text{Evaluate} E_q[CE(i, j | G)]$;

Query Answering

Given approximate posterior q, we want to be able to extract information about the system.

Let $\bigvee_i c_i$ be some feature of the causal graph, e.g. a set of edges.

- Sampling: Sample $G \sim q(\cdot; G) \models \bigvee_i c_i$;
- Marginals: Evaluate $q(\bigvee_i c_i' \mid \bigvee_i c_i)$;
- Most Likely: Evaluate $\max_G q(\cdot; G) \models \bigvee_i c_i$;
- Linear Causal Effects: Evaluate $E_q[CE(i, j \mid G)]$;

No. Edges	Method	AUROC	
4	Gadget	0:905	0:073
	Trust-g	0:903	0:057
8	Gadget	0:888	0:089
	Trust-g	0:933	0:048
16	Gadget	0:876	0:081
	Trust-g	0:957	0:077

Conclusion

- We present a novel, tractable representation for approximate Bayesian structure learning.

Conclusion

- We present a novel, tractable representation for approximate Bayesian structure learning.
- We compactly model distributions over DAGs and topological orders using OrderSPNs, a novel type of tractable probabilistic circuit.

Conclusion

- We present a novel, **tractable** representation for approximate Bayesian structure learning.
- We compactly model distributions over DAGs and topological orders using OrderSPNs, a novel type of tractable probabilistic circuit.
- Tractability offers benefits both for optimizing the variational objective, as well as in answering queries about the domain.

Thank you!

Benjie Wang

Matthew Wicker

Marta Kwiatkowska

Find out more at Poster #722!