Tractable Uncertainty for Structure Learning

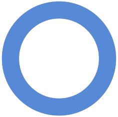
Benjie Wang, Matthew Wicker, Marta Kwiatkowska

ICML 2022

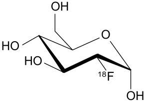
UNIVERSITY OF OXFORD

Motivation: Uncertainty in Causal Structures

[*Shen et al. 2020*]

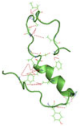


Diabetes



Fludeoxyglucose

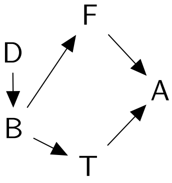
Alzheimer's



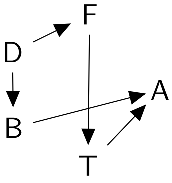
Amyloid **B**eta

Phosphorylated **T**au

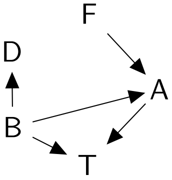
Motivation: Uncertainty in Causal Structures



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

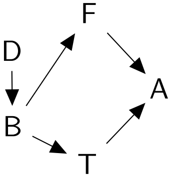


$$CE(D,A) = b_{DB}b_{BA} + b_{DF}b_{FT}b_{TA}$$

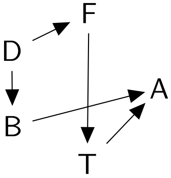


$$CE(D,A) = 0$$

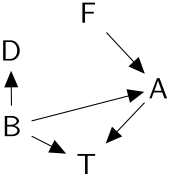
Motivation: Uncertainty in Causal Structures



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



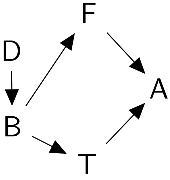
$$CE(D,A) = b_{DB}b_{BA} + b_{DF}b_{FT}b_{TA}$$



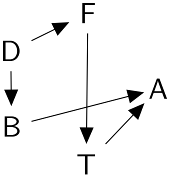
$$CE(D,A) = 0$$

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

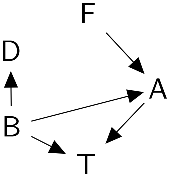
Motivation: Uncertainty in Causal Structures



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



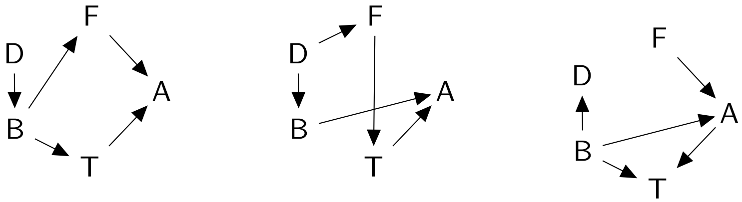
$$CE(D, A) = b_{DB}b_{BA} + b_{DF}b_{FT}b_{TA}$$



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

- 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 $\mathcal{D}$:

$$p(G|\mathcal{D}) \propto p(\mathcal{D}|G)p(G)$$

Bayesian Structure Learning

Model Express uncertainty using prior knowledge and data $\mathcal{D}$:

$$p(G|\mathcal{D}) \propto p(\mathcal{D}|G)p(G)$$

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

Bayesian Structure Learning

Model Express uncertainty using prior knowledge and data $\mathcal{D}$:

$$p(G|\mathcal{D}) \propto p(\mathcal{D}|G)p(G)$$

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

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

Bayesian Structure Learning

Model Express uncertainty using prior knowledge and data $\mathcal{D}$:

$$p(G|\mathcal{D}) \propto p(\mathcal{D}|G)p(G)$$

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

Approximation Derive some approximation $q_\phi(G) \approx p(G|\mathcal{D})$, 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 $\mathcal{D}$:

$$p(G|\mathcal{D}) \propto p(\mathcal{D}|G)p(G)$$

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

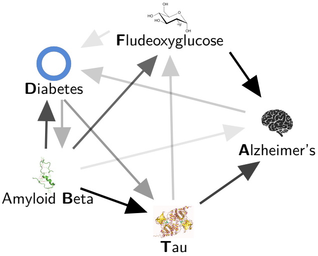
Approximation Derive some approximation $q_\phi(G) \approx p(G|\mathcal{D})$, and use q to estimate the query.

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

How do we encode acyclicity?

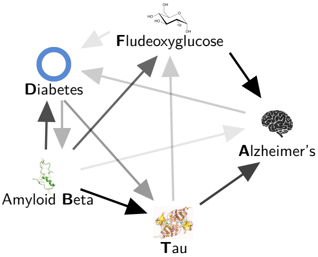
Distributions over Directed Graphs

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



Distributions over Directed Graphs

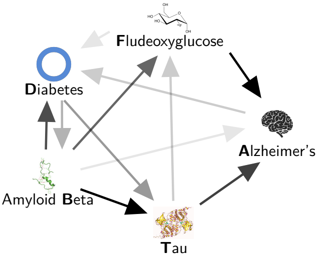
- **Mean-field**: $q_\phi(G) \propto \prod_{i,j=1}^{d} \text{Bernoulli}(G_{ij}; \phi_{ij}))$



- Does not consider correlations due to acyclicity;

Distributions over Directed Graphs

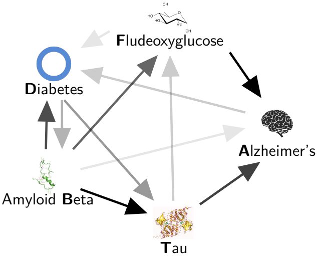
- **Mean-field**: $q_\phi(G) \propto \prod_{i,j=1}^{d} \text{Bernoulli}(G_{ij}; \phi_{ij}))$



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

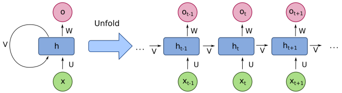
Distributions over Directed Graphs

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



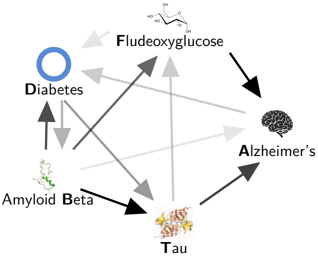
- Does not consider correlations due to acyclicity;
- Not very expressive;

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



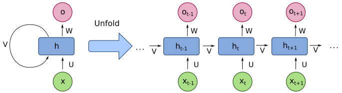
Distributions over Directed Graphs

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



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

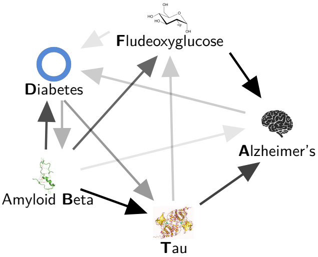
- **Neural Autoregressive**: $q_\phi(G) \propto \prod_{i,j=1}^{d} q_{\phi_{ij}}(G_{ij}|G_{<ij})$



  - Difficult to train to encode acyclicity;

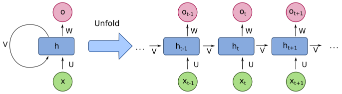
Distributions over Directed Graphs

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



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

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

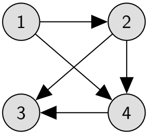


  - Difficult to train to encode acyclicity;
  - Intractable (except for sampling);

DAG Distribution using Tractable Circuits

Orderings We work on the joint space of topological orders σ and directed graphs G:

$$\sigma = \{1, 2, 4, 3\}$$

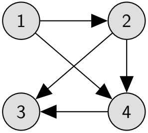


- Every DAG is consistent with at least one order;
- Every directed graph consistent with an order is acyclic;

DAG Distribution using Tractable Circuits

Orderings We work on the joint space of topological orders σ and directed graphs G:

$$\sigma = \{1, 2, 4, 3\}$$



- 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 $q_\phi(\sigma, G)$ 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 $\boldsymbol{X}$.

Sum-Product Networks

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

SPNs are rooted DAGs consisting of three types of nodes:

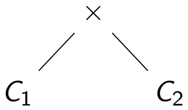
- L: Simple **base** distributions $L(\boldsymbol{X})$

Sum-Product Networks

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

SPNs are rooted DAGs consisting of three types of nodes:

- L: Simple **base** distributions $L(\boldsymbol{X})$
- $\times$: **Factorize** distributions, $P(\boldsymbol{X}) = C_1(\boldsymbol{X}_1) \times C_2(\boldsymbol{X}_2)$

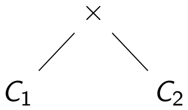


Sum-Product Networks

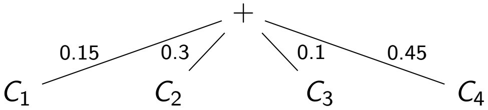
Sum-Product Networks (SPNs) are a type of tractable probabilistic model for expressing a distribution over a set of variables $\boldsymbol{X}$.

SPNs are rooted DAGs consisting of three types of nodes:

- L: Simple **base** distributions $L(\boldsymbol{X})$
- $\times$: **Factorize** distributions, $P(\boldsymbol{X}) = C_1(\boldsymbol{X}_1) \times C_2(\boldsymbol{X}_2)$



- $+$: **Mix** component distributions, $S(\boldsymbol{X}) = \sum_j \phi_j C_j(\boldsymbol{X})$



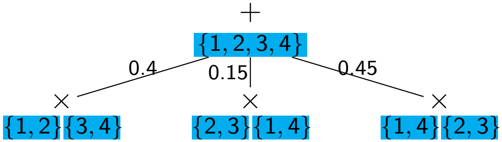
OrderSPNs

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

OrderSPNs

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

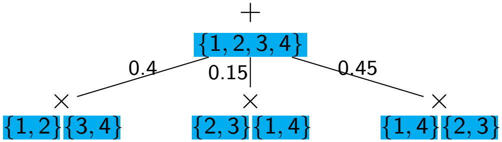
- $+$: **Mix** different partitions of the order $\sigma = (\sigma_1, \sigma_2)$



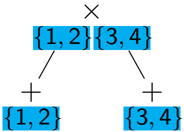
OrderSPNs

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

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



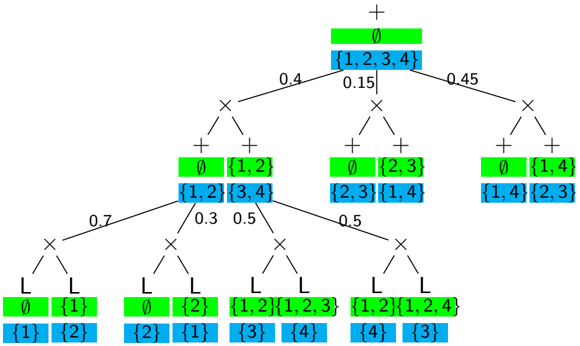
- $\times$: **Factorize** into independent $P(\sigma) = C_1(\sigma_1) \times C_2(\sigma_2)$



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

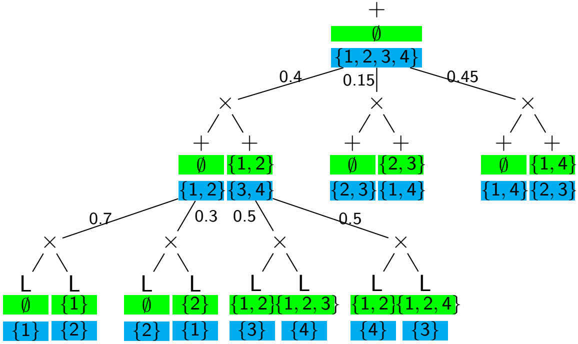
OrderSPNs

Alternate sum and product layers until order σ is fully determined:



OrderSPNs

Alternate sum and product layers until order σ is fully determined:



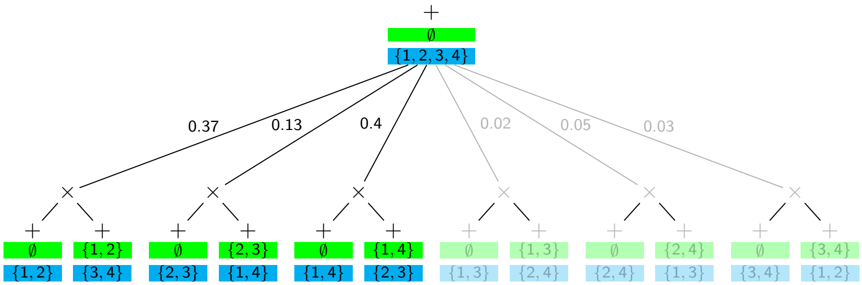
How does this relate to DAGs?

- L: (S, $\{i\}$) indicates that S precedes i in the ordering; thus $L(G_i) = 0$ if $G_i \not\subseteq S$, where G_i is the set of parents of node i.

Are OrderSPNs a good approximation to the true posterior?

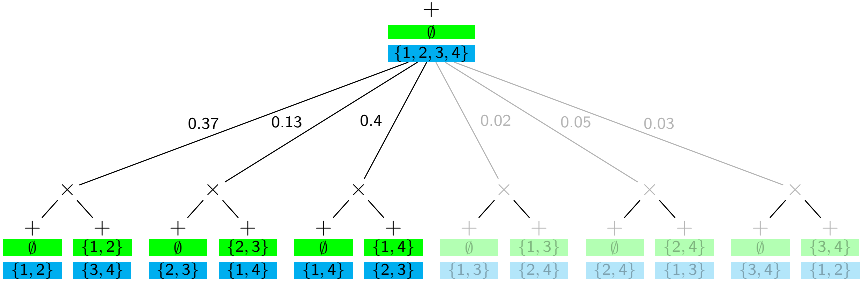
Natural Approximation

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



Natural Approximation

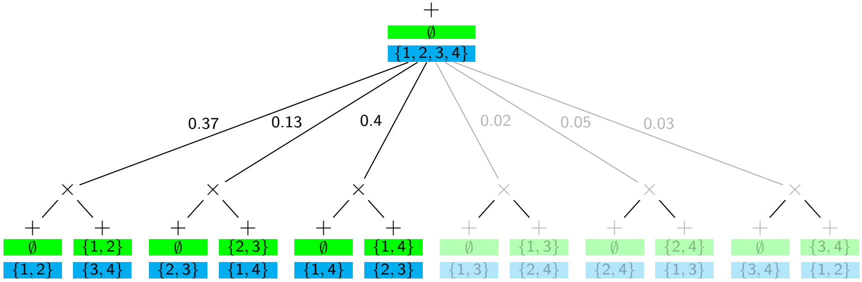
OrderSPNs can be viewed as a hierarchical, width-limited approximation to the true posterior.



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

Natural Approximation

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

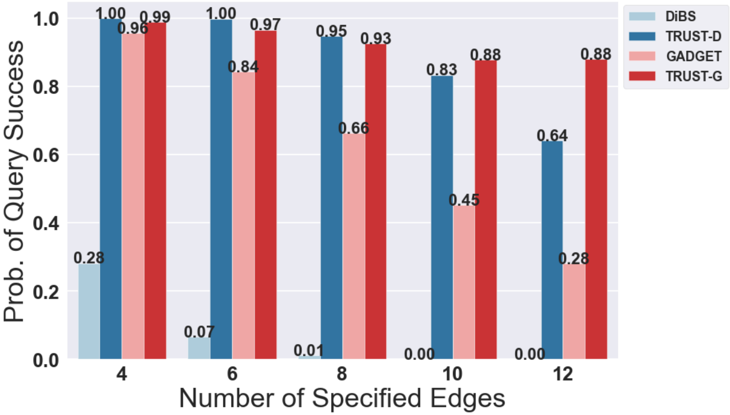


- At $+$-nodes, select the active branches (partitions) using efficient heuristic subroutines.
- $\times$-nodes encode exact conditional independences in the posterior.

OrderSPNs: Coverage

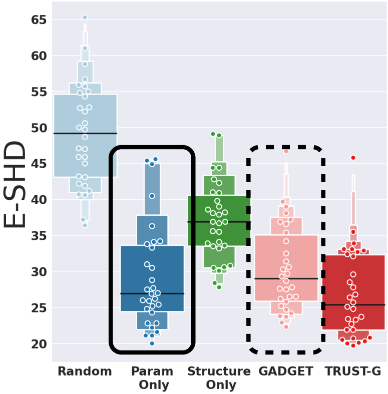
Proposition

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

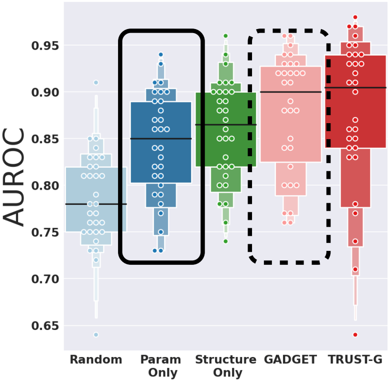


OrderSPNs: Empirical Analysis

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 Benefits 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 Effect
Mean-field	✓	✓	✓	✗	✗
Autoregressive	✓	✗	✗	✗	✗
EBM	✗	✗	✗	✗	✗
OrderSPN	✓ $O(d^2)$	✓ $O(M)$	✓ $O(M)$	✓ $O(M)$	✓ $O(d^3 M)$

Learning OrderSPN Weights

Variational inference is used to optimize the parameters:

$$ELBO = \mathbb{E}_{q_\phi(G)}[\log p(G|\mathcal{D})] + H(q_\phi(G))$$

- 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:

$$ELBO = \mathbb{E}_{q_\phi(G)}[\log p(G|\mathcal{D})] + H(q_\phi(G))$$

- 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 q_ϕ 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_ϕ, we want to be able to extract information about the system.

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

Query Answering

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

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

- **Sampling**: Sample $G \sim q_\phi(\sigma, G| \bigwedge c_i)$;

Query Answering

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

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

- **Sampling**: Sample $G \sim q_\phi(\sigma, G | \bigwedge c_i)$;
- **Marginals**: Evaluate $q_\phi(\bigwedge c_i' | \bigwedge c_i)$;

Query Answering

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

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

- **Sampling**: Sample $G \sim q_\phi(\sigma, G | \bigwedge c_i)$;
- **Marginals**: Evaluate $q_\phi(\bigwedge c_i' | \bigwedge c_i)$;
- **Most Likely**: Evaluate $\max_G q_\phi(\sigma, G | \bigwedge c_i)$;

Query Answering

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

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

- **Sampling**: Sample $G \sim q_\phi(\sigma, G | \bigwedge c_i)$;
- **Marginals**: Evaluate $q_\phi(\bigwedge c_i' | \bigwedge c_i)$;
- **Most Likely**: Evaluate $\max_G q_\phi(\sigma, G | \bigwedge c_i)$;
- **Linear Causal Effects**: Evaluate $\mathbb{E}_{q_\phi}[CE(i, j|G)]$;

Query Answering

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

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

- **Sampling**: Sample $G \sim q_\phi(\sigma, G | \bigwedge c_i)$;
- **Marginals**: Evaluate $q_\phi(\bigwedge c_i' | \bigwedge c_i)$;
- **Most Likely**: Evaluate $\max_G q_\phi(\sigma, G | \bigwedge c_i)$;
- **Linear Causal Effects**: Evaluate $\mathbb{E}_{q_\phi}[CE(i, j|G)]$;

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!