UCLA Computer Science

STAR AI RESEARCH LAB UCLA

Recent Developments in Probabilistic Circuits

Guy Van den Broeck

Google - Feb 9, 2022

Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?

Outline

1. **What are tractable probabilistic circuits?**
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?

Fully factorized

NaiveBayes AndOrGraphs PDGs

Trees PSDDs CNets LTMs SPNs NADEs

Thin Junction Trees ACs MADEs MAFs VAEs

DPPs FVSBNs TACs IAFs NAFs RAEs

Mixtures BNs NICE FGs GANs

RealNVP MNs

The *Alphabet Soup* of probabilistic models

Fully factorized

NaiveBayes AndOrGraphs PDGs

Trees PSDDs CNets LTMs SPNs NADEs

Thin Junction Trees ACs MADEs MAFs VAEs

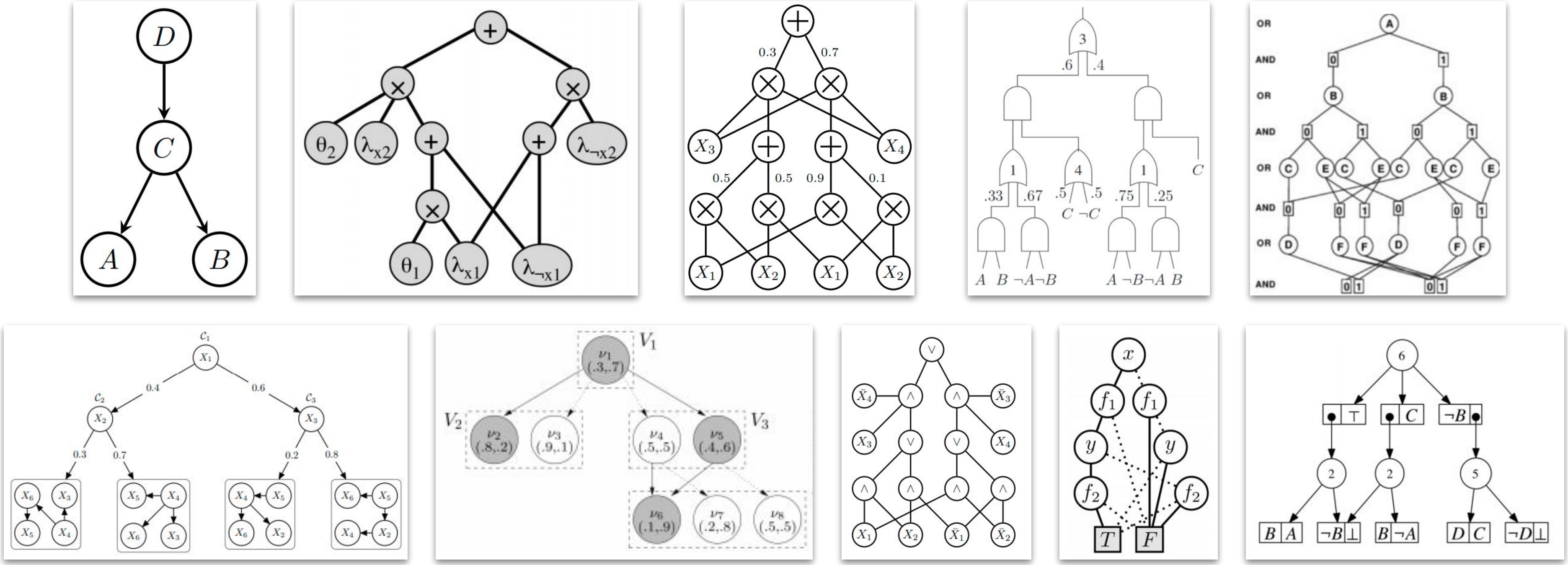
DPPs FVSBNs TACs IAFs NAFs RAEs

Mixtures BNs NICE FGs GANs

RealNVP MNs

Intractable and *tractable* models

Tractable Probabilistic Models



"Every talk needs a joke and a literature overview slide, not necessarily distinct"

- after Ron Graham

Fully factorized

NaiveBayes AndOrGraphs PDGs

Trees PSDDs CNets LTMs SPNs NADEs

Thin Junction Trees ACs MADEs MAFs VAEs

DPPs FVSBNs TACs IAFs NAFs RAEs

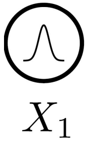
Mixtures BNs NICE FGs GANs

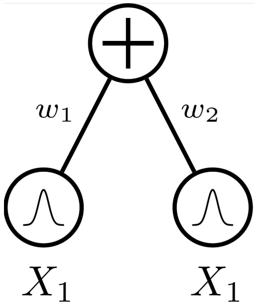
RealNVP MNs

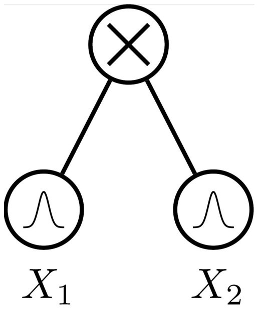
a *unifying framework* for tractable models

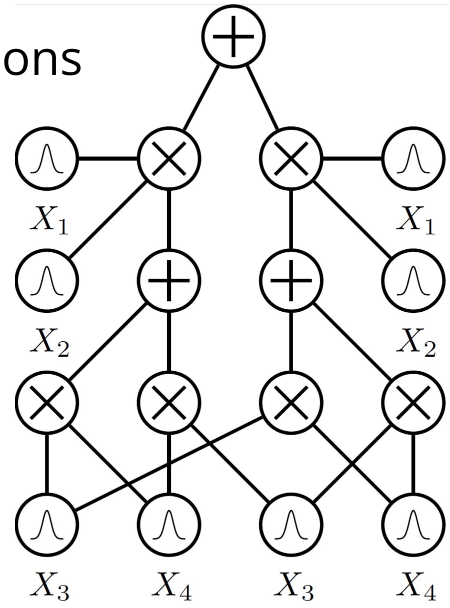
Probabilistic circuits

computational graphs that recursively define distributions









$$p(X_1) = w_1 p_1(X_1) + w_2 p_2(X_1)$$

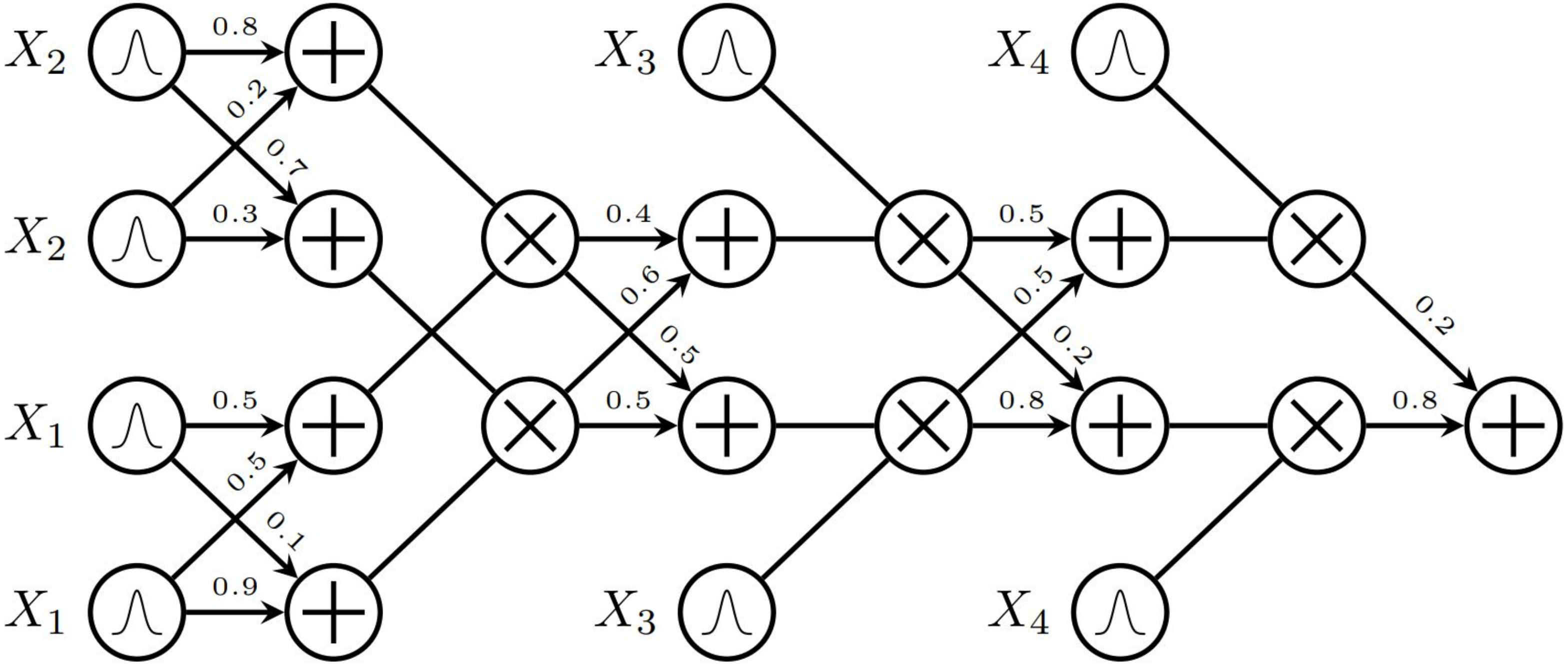
$\Rightarrow$ *mixtures*

$$p(X_1, X_2) = p(X_1) \cdot p(X_2)$$

$\Rightarrow$ *factorizations*

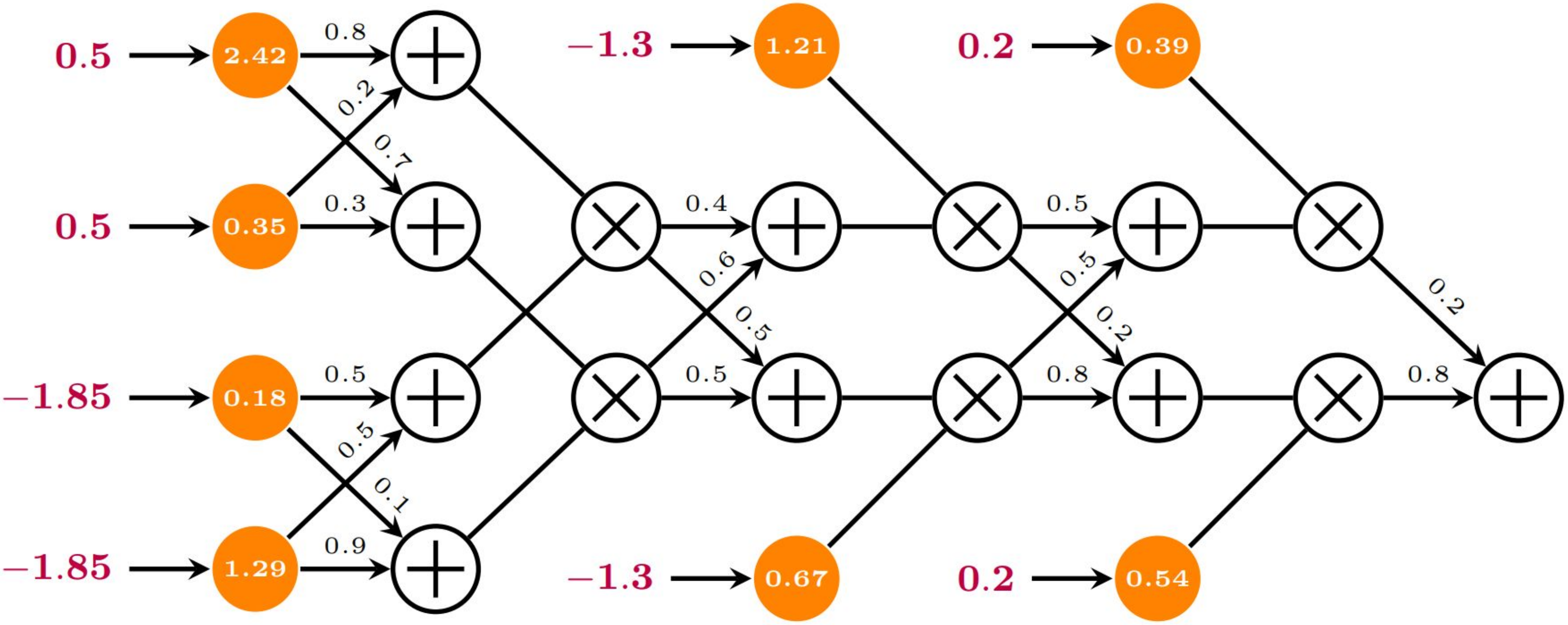
Likelihood

$$p(X_1 = \mathbf{-1.85}, X_2 = \mathbf{0.5}, X_3 = \mathbf{-1.3}, X_4 = \mathbf{0.2})$$



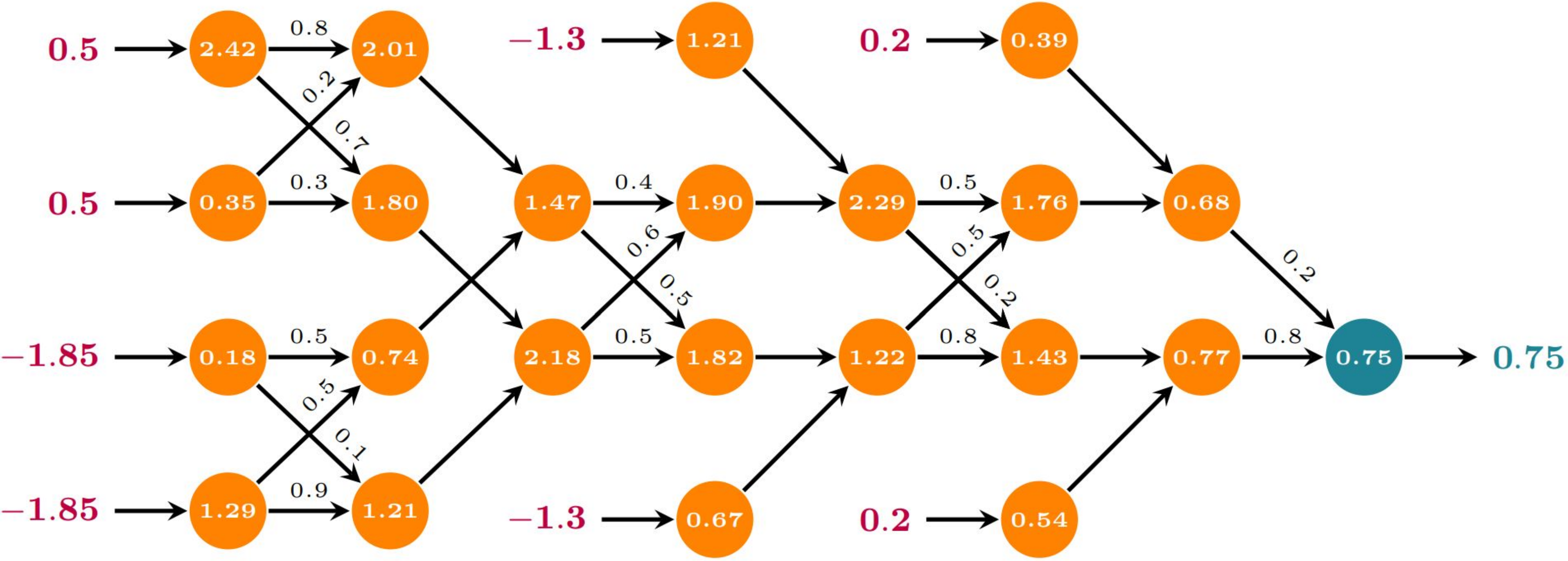
Likelihood

$$p(X_1 = \mathbf{-1.85}, X_2 = \mathbf{0.5}, X_3 = \mathbf{-1.3}, X_4 = \mathbf{0.2})$$



Likelihood

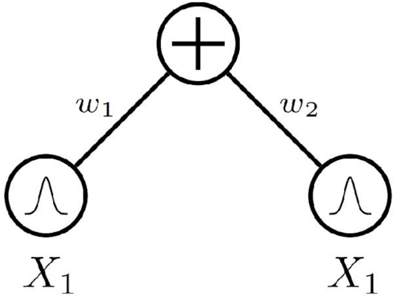
$$p(X_1 = \mathbf{-1.85}, X_2 = \mathbf{0.5}, X_3 = \mathbf{-1.3}, X_4 = \mathbf{0.2})$$



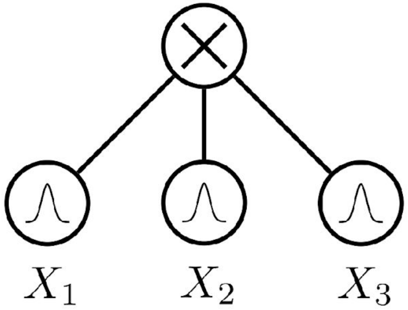
Tractable marginals

A sum node is ***smooth*** if its children depend on the same set of variables.

A product node is ***decomposable*** if its children depend on disjoint sets of variables.



smooth circuit



decomposable circuit

Darwiche and Marquis, “A Knowledge Compilation Map”, 2002

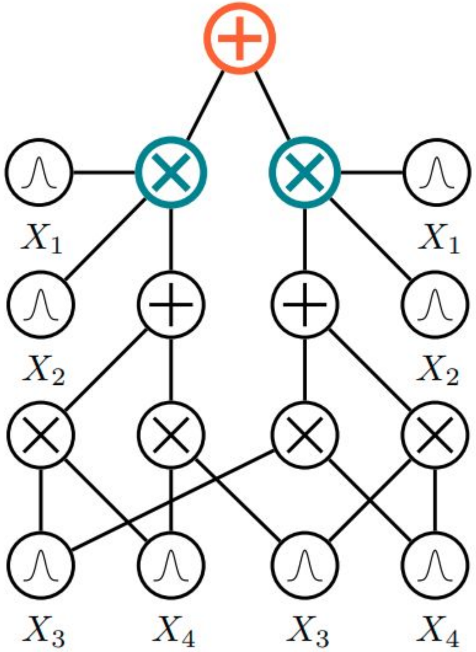
Smoothness + decomposability = tractable MAR

If $p(\mathbf{x}) = \sum_i w_i p_i(\mathbf{x})$, (***smoothness***):

$$\int p(\mathbf{x})d\mathbf{x} = \int \sum_i w_i p_i(\mathbf{x})d\mathbf{x} =$$

$$= \sum_i w_i \int p_i(\mathbf{x})d\mathbf{x}$$

$\Rightarrow$ *integrals are "pushed down" to children*



[Darwiche & Marquis JAIR 2001, Poon & Domingos UAI11]

Smoothness + **decomposability** = **tractable MAR**

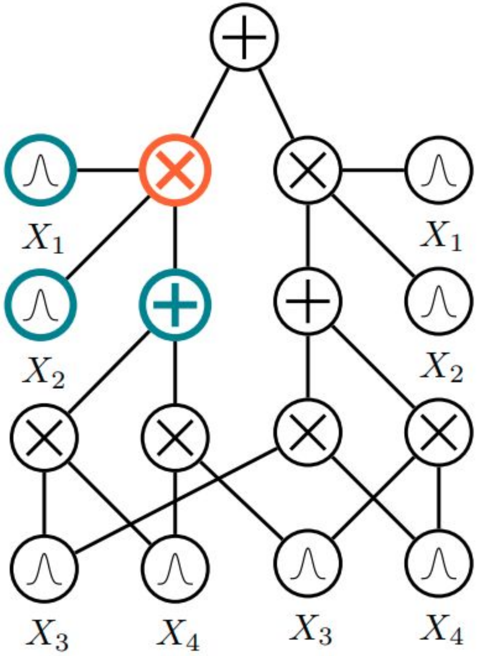
If $p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{x})p(\mathbf{y})p(\mathbf{z})$, (***decomposability***):

$$\int\int\int p(\mathbf{x}, \mathbf{y}, \mathbf{z})d\mathbf{x}d\mathbf{y}d\mathbf{z} =$$

$$= \int\int\int p(\mathbf{x})p(\mathbf{y})p(\mathbf{z})d\mathbf{x}d\mathbf{y}d\mathbf{z} =$$

$$= \int p(\mathbf{x})d\mathbf{x} \int p(\mathbf{y})d\mathbf{y} \int p(\mathbf{z})d\mathbf{z}$$

$\Rightarrow$ *integrals decompose into easier ones*



Smoothness + decomposability = tractable MAR

Forward pass evaluation for MAR

$\Rightarrow$ *linear in circuit size!*

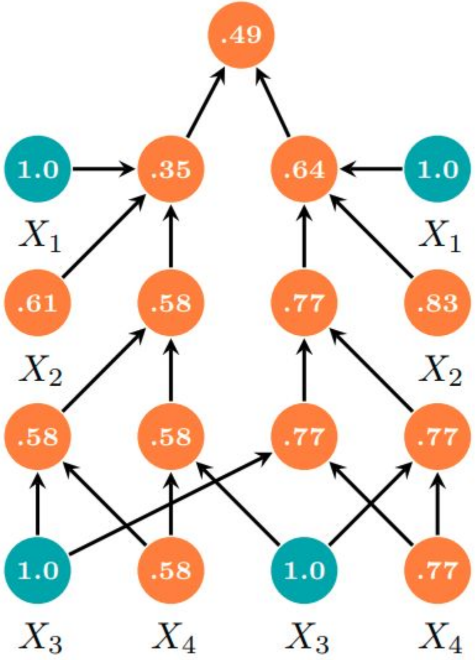
E.g. to compute $p(x_2, x_4)$:

- leafs over X_1 and X_3 output $\boldsymbol{Z}_i = \int p(x_i)dx_i$

  $\Rightarrow$ *for normalized leaf distributions:* 1.0

- leafs over X_2 and X_4 output ***EVI***

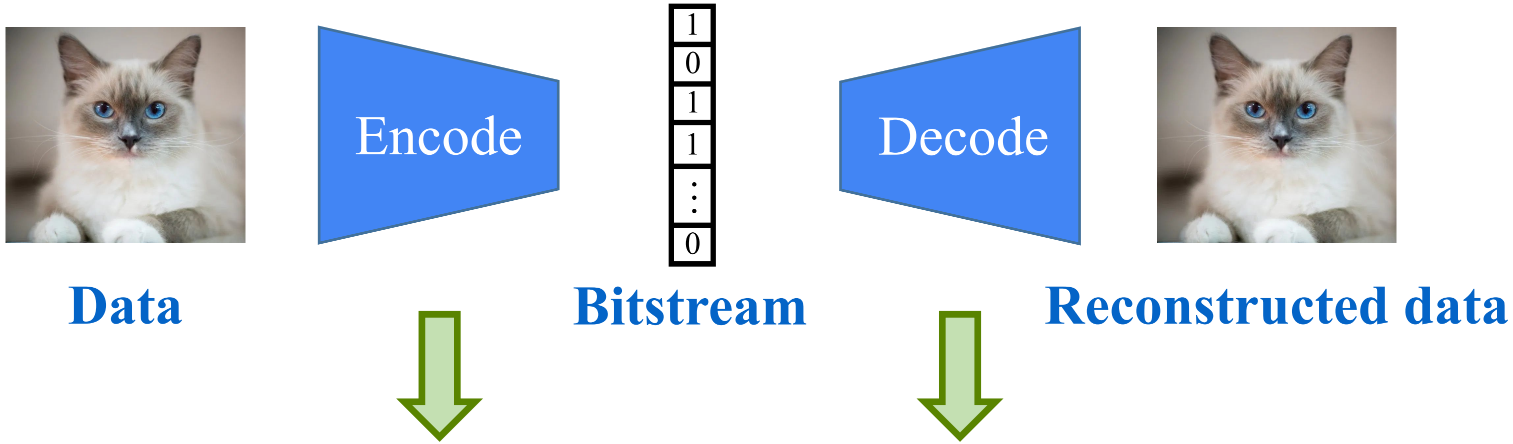
- feedforward evaluation (bottom-up)



Outline

1. What are tractable probabilistic circuits?
2. **Are these models any good?**
3. What is their expressive power?
4. How far can we push tractable inference?

Lossless Data Compression



Expressive probabilistic model $p(\boldsymbol{x})$ ⇨ Determines the theoretical limit of compression rate

+

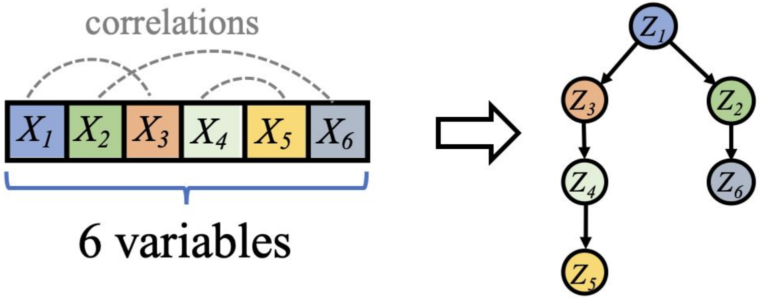
Efficient coding algorithm ⇨ How close we can approach the theoretical limit

Anji Liu, Stephan Mandt and Guy Van den Broeck. Lossless Compression with Probabilistic Circuits, 2021.

Learning Expressive Probabilistic Circuits

Hidden Chow-Liu Trees: CLT-based latent variable PGM/PC

Learned **CLT structure** captures strong pairwise dependencies



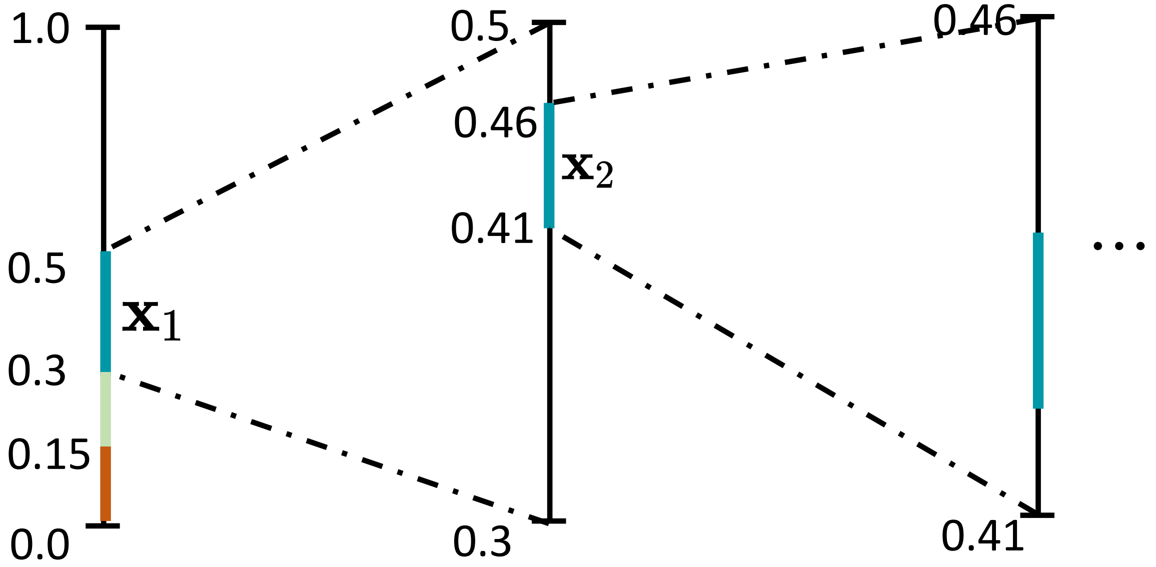
⇨ **Compile** into an equivalent PC

⇨ Mini-batch Stochastic **Expectation Maximization**

Anji Liu and Guy Van den Broeck. Tractable Regularization of Probabilistic Circuits, *NeurIPS*, 2021.

A Typical Streaming Code – Arithmetic Coding

We want to compress a set of variables (e.g., pixels, letters) $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$



Compress $\mathbf{x}_1$ with $-\log p(\mathbf{x}_1)$ bits

Compress $\mathbf{x}_2$ with $-\log p(\mathbf{x}_2|\mathbf{x}_1)$ bits

Compress $\mathbf{x}_3$ with $-\log p(\mathbf{x}_3|\mathbf{x}_1, \mathbf{x}_2)$ bits

Need to compute

$$p(X_1 < x_1)$$
$$p(X_1 \leq x_1)$$
$$p(X_2 < x_2|x_1)$$
$$p(X_2 \leq x_2|x_1)$$
$$p(X_3 < x_3|x_1, x_2)$$
$$p(X_3 \leq x_3|x_1, x_2)$$
$$\vdots$$

Efficient Lossless Compression

Need to compute

$$p(X_1 < x_1)$$
$$p(X_1 \leq x_1)$$
$$p(X_2 < x_2 | x_1)$$
$$p(X_2 \leq x_2 | x_1)$$
$$p(X_3 < x_3 | x_1, x_2)$$
$$p(X_3 \leq x_3 | x_1, x_2)$$
$$\vdots$$

Fully factorized

- Fast inference
- Not expressive

High tree-width PGMs

- Expressive
- Slow inference

Existing Flow- and VAE-based lossless compression algorithms learn to transform fully factorized distributions into the target distribution.

But en/decoding speed is still relatively slow.

Anji Liu, Stephan Mandt and Guy Van den Broeck. Lossless Compression with Probabilistic Circuits, 2021.

Efficient Lossless Compression with Probabilistic Circuits

Need to compute

$$p(X_1 < x_1)$$
$$p(X_1 \leq x_1)$$
$$p(X_2 < x_2 | x_1)$$
$$p(X_2 \leq x_2 | x_1)$$
$$p(X_3 < x_3 | x_1, x_2)$$
$$p(X_3 \leq x_3 | x_1, x_2)$$
$$\vdots$$

Fully factorized

- Fast inference
- Not expressive

High tree-width PGMs

- Expressive
- Slow inference

Probabilistic Circuits

- Expressive → SoTA likelihood on MNIST.
- Fast inference → Time complexity of en/decoding is $\mathcal{O}(\log(D) \cdot |p|)$, where D is the # variables and $|p|$ is the size of the PC.

Anji Liu, Stephan Mandt and Guy Van den Broeck. Lossless Compression with Probabilistic Circuits, 2021.

Efficient Lossless Compression with Probabilistic Circuits

SoTA compression rates

Dataset	HCLT (ours)	IDF	BitSwap	BB-ANS	JPEG2000	WebP	McBits
MNIST	**1.24** (1.20)	1.96 (1.90)	1.31 (1.27)	1.42 (1.39)	3.37	2.09	(1.98)
FashionMNIST	3.37 (3.34)	3.50 (3.47)	**3.35** (3.28)	3.69 (3.66)	3.93	4.62	(3.72)
EMNIST (Letter)	**1.84** (1.80)	2.02 (1.95)	1.90 (1.84)	2.29 (2.26)	3.62	3.31	(3.12)
EMNIST (ByClass)	**1.89** (1.85)	2.04 (1.98)	1.91 (1.87)	2.24 (2.23)	3.61	3.34	(3.14)

Compress and decompress 5-20x faster than NN methods with similar bitrates

Method	# parameters	Theoretical bpd	Codeword bpd	Comp. time (s)	Decomp. time (s)
PC (HCLT, $M=16$)	3.3M	1.26	1.30	15	44
PC (HCLT, $M=24$)	5.1M	1.22	1.26	26	89
PC (HCLT, $M=32$)	7.0M	1.20	1.24	44	155
IDF	24.1M	1.90	1.96	288	592
BitSwap	2.8M	1.27	1.31	578	326

Anji Liu, Stephan Mandt and Guy Van den Broeck. Lossless Compression with Probabilistic Circuits, 2021.

Efficient Lossless Compression with Probabilistic Circuits

Can be effectively combined with Flow models to achieve better generative performance

Model	CIFAR10	ImageNet32	ImageNet64
RealNVP	3.49	4.28	3.98
Glow	3.35	4.09	3.81
IDF	3.32	4.15	3.90
IDF++	**3.24**	4.10	3.81
PC+IDF	3.28	**3.99**	**3.71**

Anji Liu, Stephan Mandt and Guy Van den Broeck. Lossless Compression with Probabilistic Circuits, 2021.

Fully factorized

NaiveBayes *AndOrGraphs* *PDGs*

Trees *PSDDs* *CNets* *LTMs* *SPNs* *NADEs*

Thin Junction Trees *ACs* *MADEs* *MAFs* *VAEs*

DPPs *FVSBNs* *TACs* *IAFs* *NAFs* *RAEs*

Mixtures *BNs* *NICE* *FGs* *GANs*

RealNVP *MNs*

Expressive **models without** ***compromises***

Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. **What is their expressive power?**
4. How far can we push tractable inference?

Probabilistic circuits seem awfully general.

Are all tractable probabilistic models probabilistic circuits?

Enter: Determinantal Point Processes (DPPs)

DPPs are models where probabilities are specified by (sub)determinants

$$L = \begin{bmatrix} 1 & 0.9 & 0.8 & 0 \\ 0.9 & 0.97 & 0.96 & 0 \\ 0.8 & 0.96 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

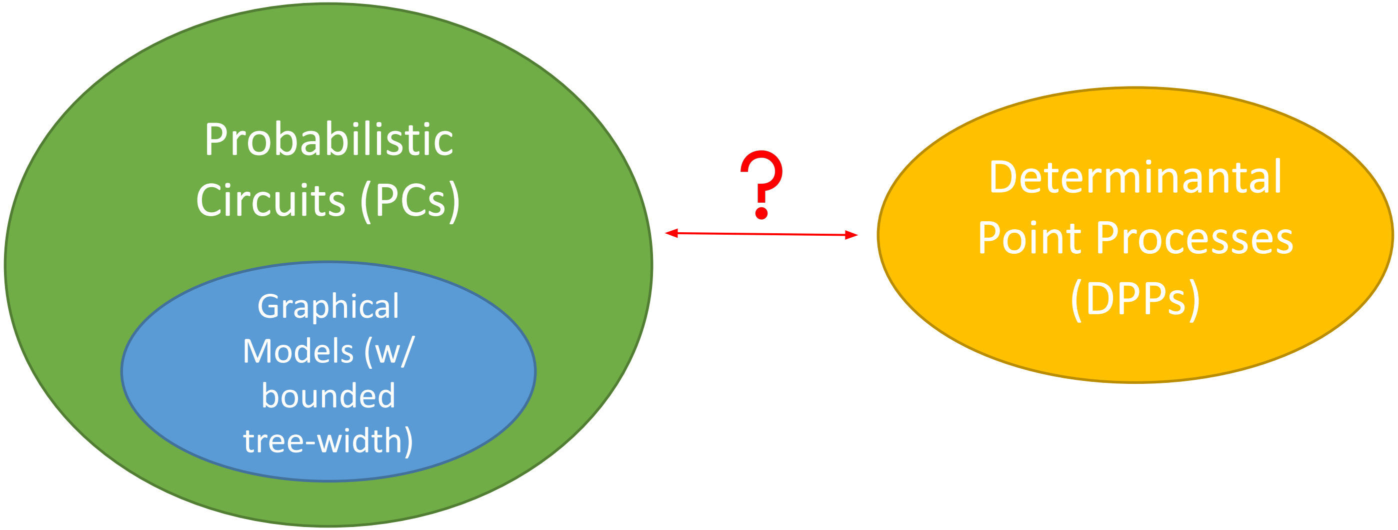
Tractable likelihoods and marginals

Global Negative Dependence

Diversity in recommendation systems

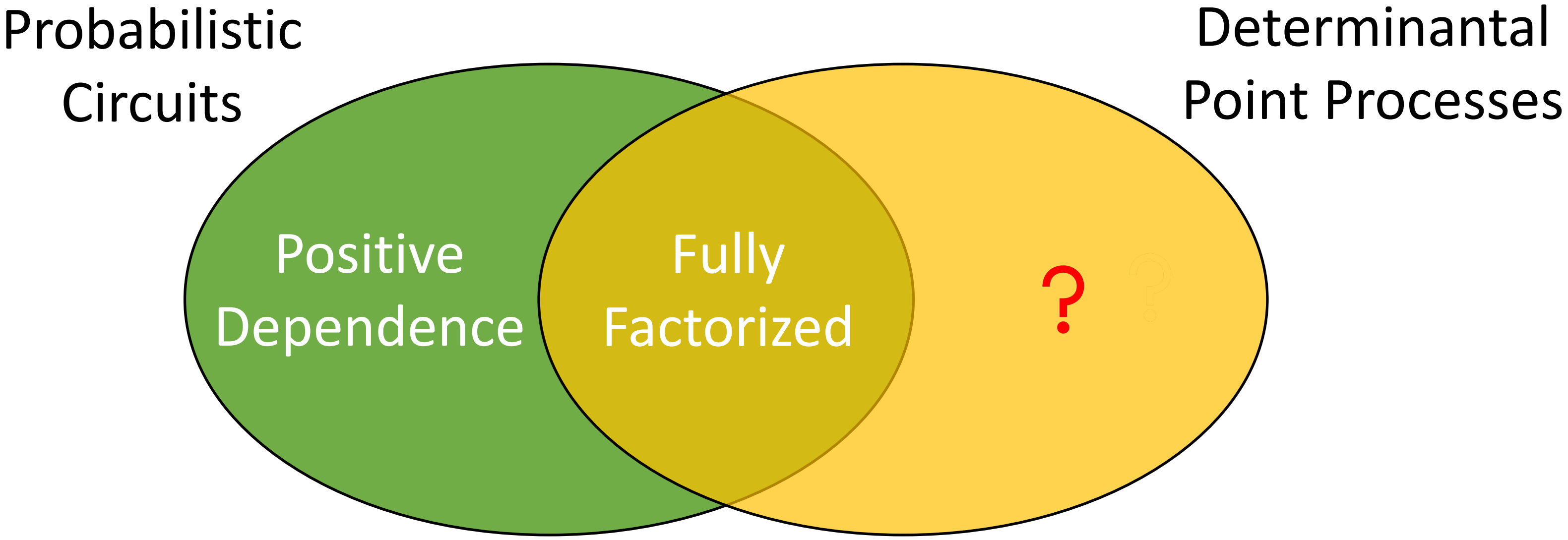
$$\Pr_L(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 0) = \frac{1}{\det(L+I)} \det(L_{\{1,2\}})$$

Are all tractable probabilistic models probabilistic circuits?



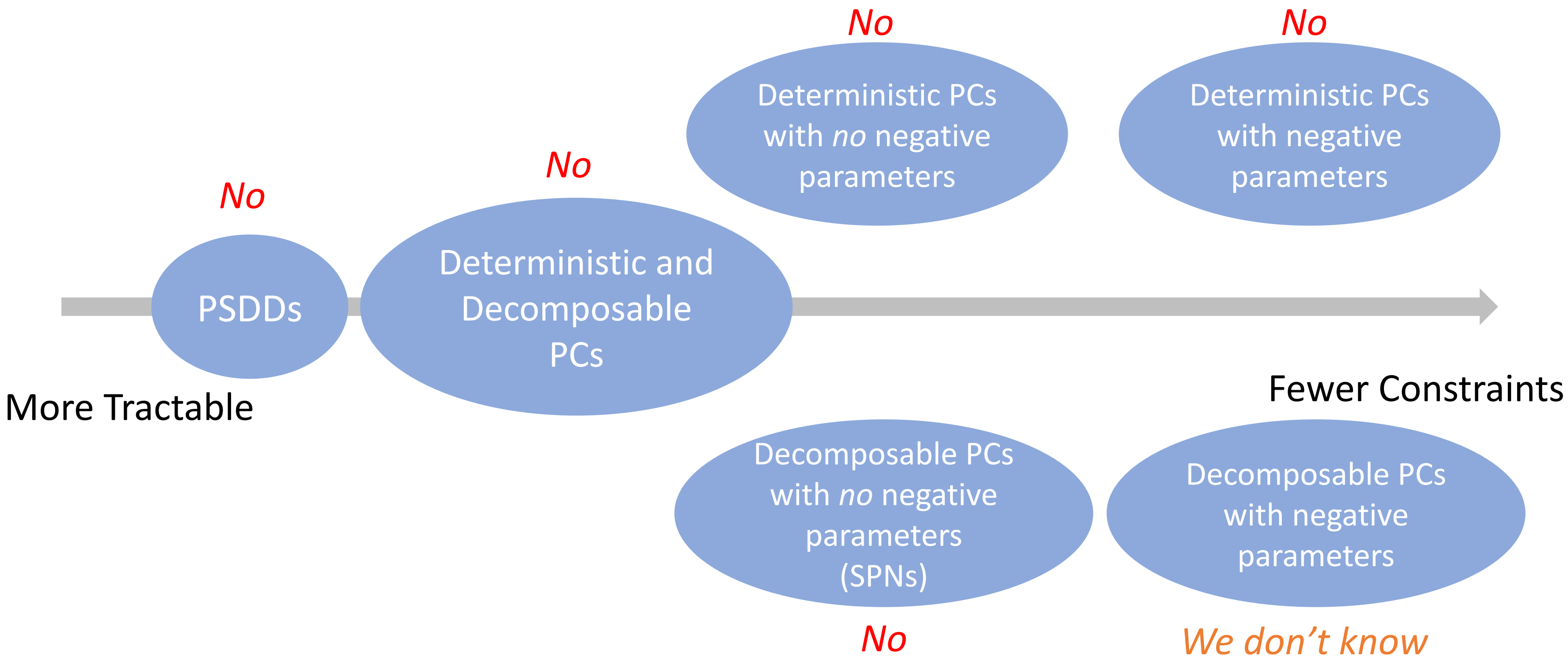
Honghua Zhang, Steven Holtzen and Guy Van den Broeck. On the Relationship Between Probabilistic Circuits and Determinantal Point Processes, *UAI*, 2020.

Relationship between PCs and DPPs



Honghua Zhang, Steven Holtzen and Guy Van den Broeck. On the Relationship Between Probabilistic Circuits and Determinantal Point Processes, *UAI*, 2020.

We cannot tractably represent DPPs with subclasses of PCs



Honghua Zhang, Steven Holtzen and Guy Van den Broeck. On the Relationship Between Probabilistic Circuits and Determinantal Point Processes, *UAI*, 2020.

PCs cannot Tractably Represent DPPs

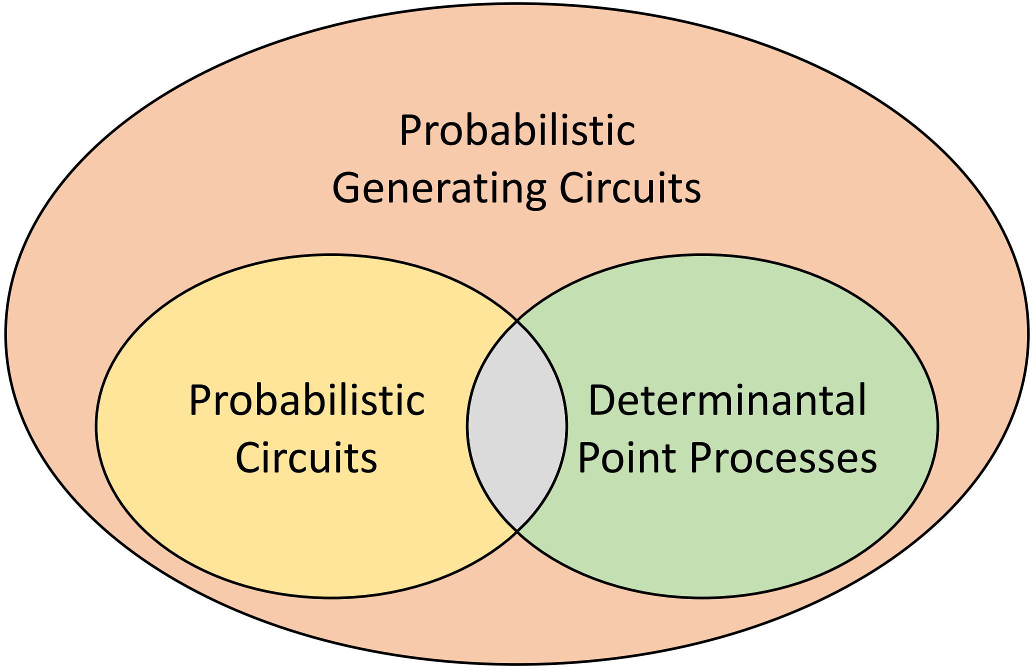
Theorem 1. *For a DPP with kernel* L=B^T * B*, where* B *is randomly generated, with probability 1, this DPP cannot be represented by polynomial-size PSDDs.*

Theorem 2. *There exists a class of DPPs that cannot be tractably represented by deterministic PCs with (possibly) negative parameters.*

Theorem 3. *There exists a class of DPPs that cannot be tractably represented by decomposable PCs with non-negative parameters (SPNs).*

Honghua Zhang, Steven Holtzen and Guy Van den Broeck. On the Relationship Between Probabilistic Circuits and Determinantal Point Processes, *UAI*, 2020.

Probabilistic Generating Circuits



A Tractable Unifying Framework for PCs and DPPs

Honghua Zhang, Brendan Juba and Guy Van den Broeck. Probabilistic Generating Circuits, *ICML*, 2021.

Probability Generating Functions

X_1	X_2	X_3	$\Pr_\beta$
0	0	0	0.02
0	0	1	0.08
0	1	0	0.12
0	1	1	0.48
1	0	0	0.02
1	0	1	0.08
1	1	0	0.04
1	1	1	0.16

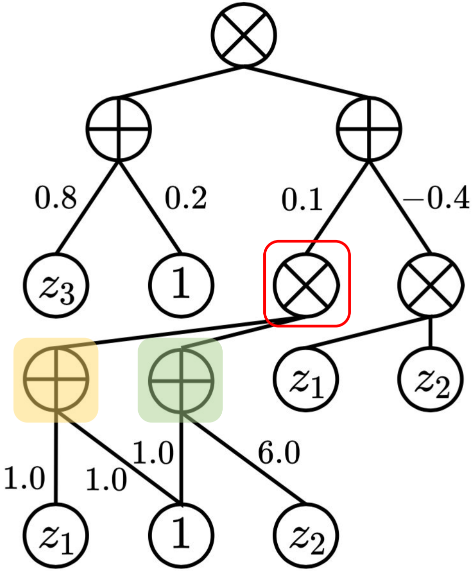
$$g_\beta = \boxed{0.16 z_1 z_2 z_3} + 0.04 z_1 z_2 + 0.08 z_1 z_3 + 0.02 z_1 + 0.48 z_2 z_3 + 0.12 z_2 + 0.08 z_3 + 0.02.$$

$$g_\beta = (0.1(z_1 + 1)(6z_2 + 1) - 0.4 z_1 z_2)(0.8 z_3 + 0.2)$$

Honghua Zhang, Brendan Juba and Guy Van den Broeck. Probabilistic Generating Circuits, *ICML*, 2021.

Probabilistic Generating Circuits (PGCs)

$$g_\beta = (0.1\boxed{(z_1+1)(6z_2+1)} - 0.4z_1z_2)(0.8z_3+0.2)$$



1. Sum nodes ⊕ with weighted edges to children.
2. Product nodes ⊗ with unweighted edges to children.
3. Leaf nodes: z_i or constant.

Honghua Zhang, Brendan Juba and Guy Van den Broeck. Probabilistic Generating Circuits, *ICML*, 2021.

DPPs as PGCs

The generating polynomial for a DPP with kernel L is given by:

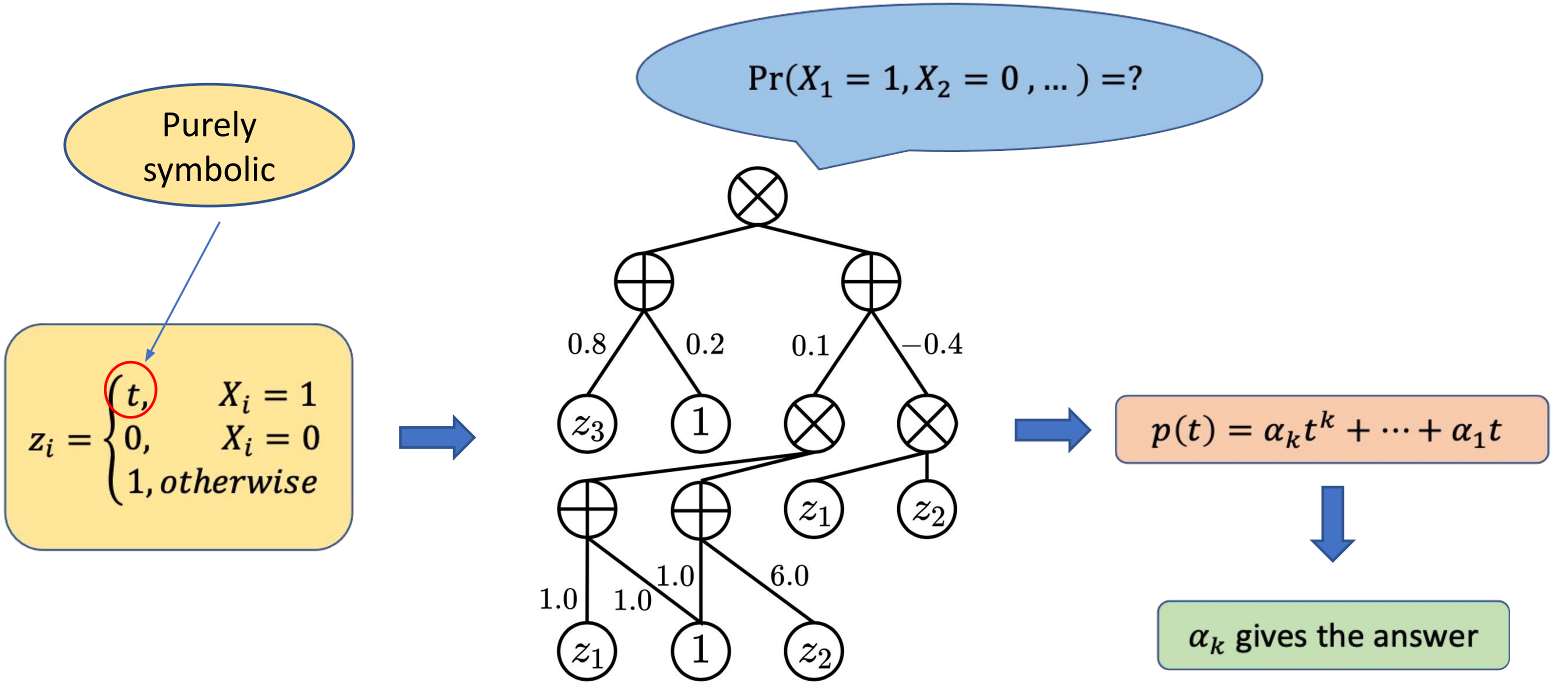
$$g_L = \frac{1}{\det(L+I)} \det(I + L\mathrm{diag}(z_1, \ldots, z_n)).$$

Constant

Division-free determinant algorithm
(Samuelson-Berkowitz algorithm)

g_L can be represented as a PGC of size $O(n^4)$

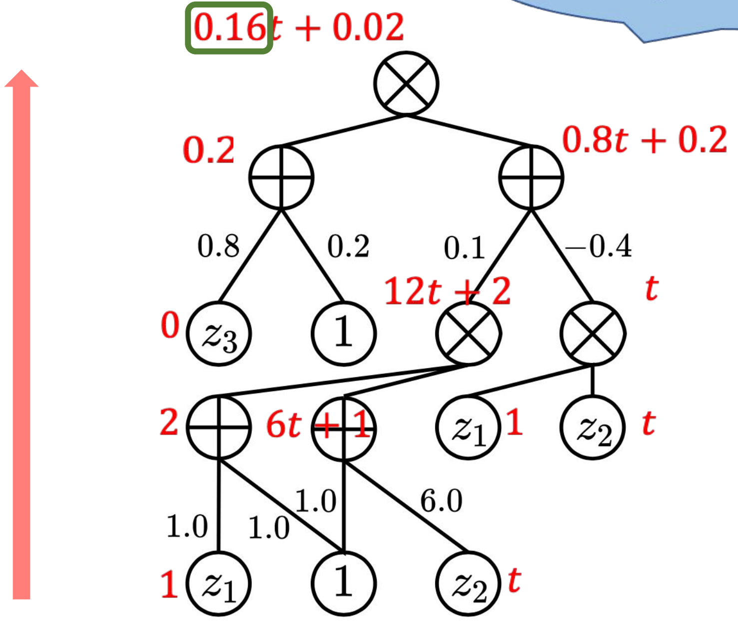
PGCs Support Tractable Likelihoods/Marginals



Honghua Zhang, Brendan Juba and Guy Van den Broeck. Probabilistic Generating Circuits, *ICML*, 2021.

Example

$\Pr(X_2 = 1, X_3 = 0) = ?$



X_1	X_2	X_3	$\Pr_\beta$
0	0	0	0.02
0	0	1	0.08
0	1	0	0.12
0	1	1	0.48
1	0	0	0.02
1	0	1	0.08
1	1	0	0.04
1	1	1	0.16

Honghua Zhang, Brendan Juba and Guy Van den Broeck. Probabilistic Generating Circuits, *ICML*, 2021.

Experiment Results: Amazon Baby Registries

	DPP	Strudel	EiNet	MT	SimplePGC
apparel	-9.88	-9.51	-9.24	-9.31	$\mathbf{-9.10}^{*\dagger\circ}$
bath	-8.55	-8.38	-8.49	-8.53	$\mathbf{-8.29}^{*\dagger\circ}$
bedding	-8.65	-8.50	-8.55	-8.59	$\mathbf{-8.41}^{*\dagger\circ}$
carseats	-4.74	-4.79	-4.72	-4.76	$\mathbf{-4.64}^{*\dagger\circ}$
diaper	-10.61	-9.90	-9.86	-9.93	$\mathbf{-9.72}^{*\dagger\circ}$
feeding	-11.86	-11.42	-11.27	-11.30	$\mathbf{-11.17}^{*\dagger\circ}$
furniture	-4.38	-4.39	-4.38	-4.43	$\mathbf{-4.34}^{*\dagger\circ}$
gear	-9.14	-9.15	-9.18	-9.23	$\mathbf{-9.04}^{*\dagger\circ}$
gifts	-3.51	$\mathbf{-3.39}$	-3.42	-3.48	-3.47°
health	-7.40	-7.37	-7.47	-7.49	$\mathbf{-7.24}^{*\dagger\circ}$
media	-8.36	$\mathbf{-7.62}$	-7.82	-7.93	$-7.69^{\dagger\circ}$
moms	-3.55	-3.52	$\mathbf{-3.48}$	-3.54	-3.53°
safety	-4.28	-4.43	-4.39	-4.36	$\mathbf{-4.28}^{*\dagger\circ}$
strollers	-5.30	-5.07	-5.07	-5.14	$\mathbf{-5.00}^{*\dagger\circ}$
toys	-8.05	$\mathbf{-7.61}$	-7.84	-7.88	$-7.62^{\dagger\circ}$

SimplePGC achieves SOTA result on 11/15 datasets

Honghua Zhang, Brendan Juba and Guy Van den Broeck. Probabilistic Generating Circuits, *ICML*, 2021.

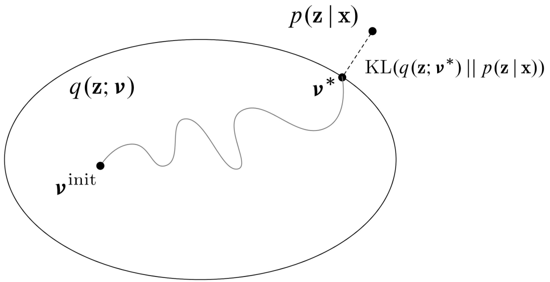
Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. **How far can we push tractable inference?**

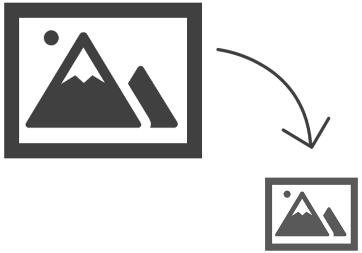
Outline

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. **How far can we push tractable inference?**
   Cool things we can do with circuits :-)

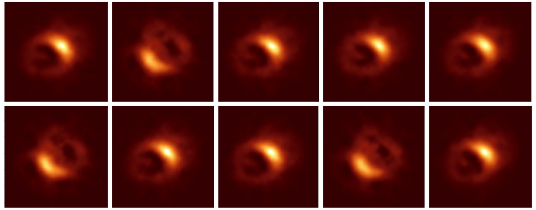
Information-theoretic quantities



Variational inference



Compression

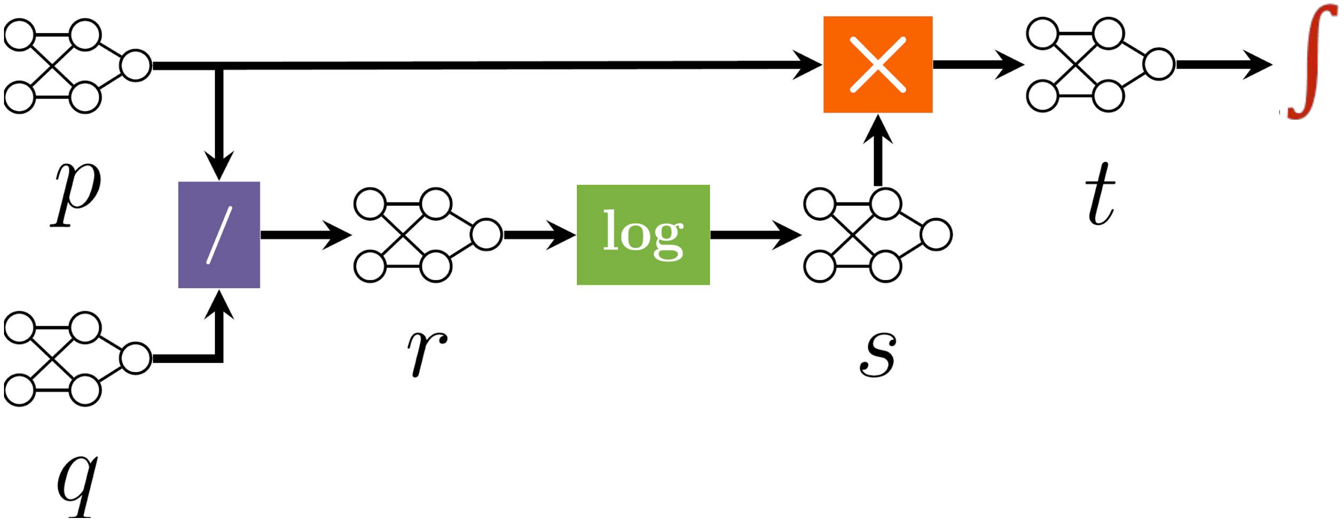


Black hole imaging

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso and Guy Van den Broeck. A Compositional Atlas of Tractable Circuit Operations for Probabilistic Inference, *NeurIPS*, 2021.

Queries as pipelines

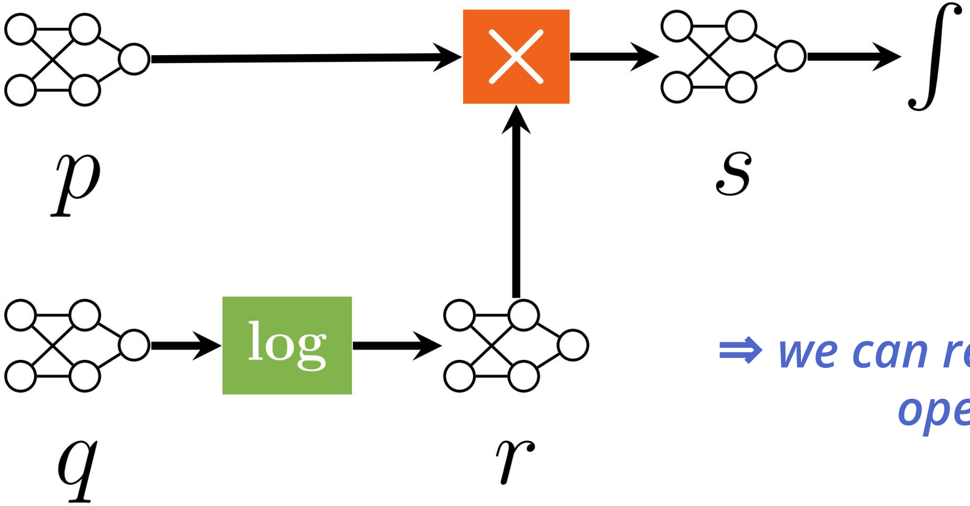
$$\mathbb{KLD}(p \parallel q) = \int p(\boldsymbol{x}) \times \log((p(\boldsymbol{x})/q(\boldsymbol{x}))d\boldsymbol{X}$$



Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso and Guy Van den Broeck. A Compositional Atlas of Tractable Circuit Operations for Probabilistic Inference, *NeurIPS*, 2021.

Queries as pipelines

$$H(p,q) = \int p(\boldsymbol{x}) \times \log(q(\boldsymbol{x}))d\boldsymbol{X}$$

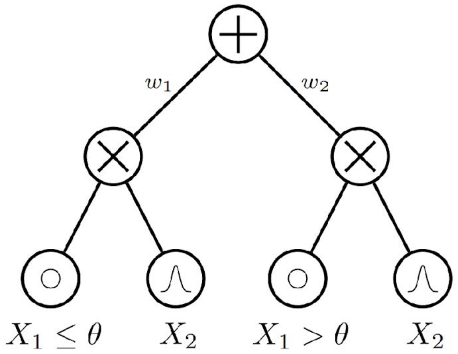


⇒ we can reuse the operations!

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso and Guy Van den Broeck. A Compositional Atlas of Tractable Circuit Operations for Probabilistic Inference, *NeurIPS*, 2021.

Determinism

A sum node is ***deterministic*** if only one of its children outputs non-zero for any input



deterministic circuit

⇒ *allows* ***tractable MAP*** *inference*

$$argmax_{\boldsymbol{x}}\, p(\boldsymbol{x})$$

Darwiche and Marquis, “A Knowledge Compilation Map”, 2002

Operation		Tractability	
		Input conditions	Output conditions
LOG	$\log(p)$	Sm, Dec, Det	Sm, Dec

smooth, decomposable, deterministic

smooth, decomposable

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso and Guy Van den Broeck. A Compositional Atlas of Tractable Circuit Operations for Probabilistic Inference, *NeurIPS*, 2021.

Tractable circuit operations

Operation		Tractability: Input properties	Tractability: Output properties	Hardness
SUM	$\theta_1 p + \theta_2 q$	(+Cmp)	(+SD)	NP-hard for Det output
PRODUCT	$p \cdot q$	Cmp (+Det, +SD)	Dec (+Det, +SD)	#P-hard w/o Cmp
POWER	$p^n, n \in \mathbb{N}$	SD (+Det)	SD (+Det)	#P-hard w/o SD
	$p^\alpha, \alpha \in \mathbb{R}$	Sm, Dec, Det (+SD)	Sm, Dec, Det (+SD)	#P-hard w/o Det
QUOTIENT	p/q	Cmp; q Det (+p Det,+SD)	Dec (+Det,+SD)	#P-hard w/o Det
LOG	$\log(p)$	Sm, Dec, Det	Sm, Dec	#P-hard w/o Det
EXP	$\exp(p)$	linear	SD	#P-hard

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso and Guy Van den Broeck. A Compositional Atlas of Tractable Circuit Operations for Probabilistic Inference, *NeurIPS*, 2021.

Inference by tractable operations

systematically derive tractable inference algorithm of complex queries

	Query	Tract. Conditions	Hardness
CROSS ENTROPY	$-\int p(\boldsymbol{x}) \log q(\boldsymbol{x})\, \mathrm{d}\mathbf{X}$	Cmp, q Det	#P-hard w/o Det
SHANNON ENTROPY	$-\sum p(\boldsymbol{x}) \log p(\boldsymbol{x})$	Sm, Dec, Det	coNP-hard w/o Det
RÉNYI ENTROPY	$(1-\alpha)^{-1} \log \int p^{\alpha}(\boldsymbol{x})\, d\mathbf{X}, \alpha \in \mathbb{N}$	SD	#P-hard w/o SD
	$(1-\alpha)^{-1} \log \int p^{\alpha}(\boldsymbol{x})\, d\mathbf{X}, \alpha \in \mathbb{R}_{+}$	Sm, Dec, Det	#P-hard w/o Det
MUTUAL INFORMATION	$\int p(\boldsymbol{x}, \boldsymbol{y}) \log(p(\boldsymbol{x}, \boldsymbol{y})/(p(\boldsymbol{x})p(\boldsymbol{y})))$	Sm, SD, Det*	coNP-hard w/o SD
KULLBACK-LEIBLER DIV.	$\int p(\boldsymbol{x}) \log(p(\boldsymbol{x})/q(\boldsymbol{x})) d\mathbf{X}$	Cmp, Det	#P-hard w/o Det
RÉNYI'S ALPHA DIV.	$(1-\alpha)^{-1} \log \int p^{\alpha}(\boldsymbol{x}) q^{1-\alpha}(\boldsymbol{x})\, d\mathbf{X}, \alpha \in \mathbb{N}$	Cmp, q Det	#P-hard w/o Det
	$(1-\alpha)^{-1} \log \int p^{\alpha}(\boldsymbol{x}) q^{1-\alpha}(\boldsymbol{x})\, d\mathbf{X}, \alpha \in \mathbb{R}$	Cmp, Det	#P-hard w/o Det
ITAKURA-SAITO DIV.	$\int [p(\boldsymbol{x})/q(\boldsymbol{x}) - \log(p(\boldsymbol{x})/q(\boldsymbol{x})) - 1] d\,\mathbf{X}$	Cmp, Det	#P-hard w/o Det
CAUCHY-SCHWARZ DIV.	$-\log \frac{\int p(\boldsymbol{x}) q(\boldsymbol{x}) d\mathbf{X}}{\sqrt{\int p^2(\boldsymbol{x}) d\mathbf{X} \int q^2(\boldsymbol{x}) d\mathbf{X}}}$	Cmp	#P-hard w/o Cmp
SQUARED LOSS	$\int (p(\boldsymbol{x}) - q(\boldsymbol{x}))^2 d\,\mathbf{X}$	Cmp	#P-hard w/o Cmp

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso and Guy Van den Broeck. A Compositional Atlas of Tractable Circuit Operations for Probabilistic Inference, *NeurIPS*, 2021.

Even harder queries

Marginal MAP

Given a set of query variables $\boldsymbol{Q} \subset \boldsymbol{X}$ and evidence $\boldsymbol{e}$,

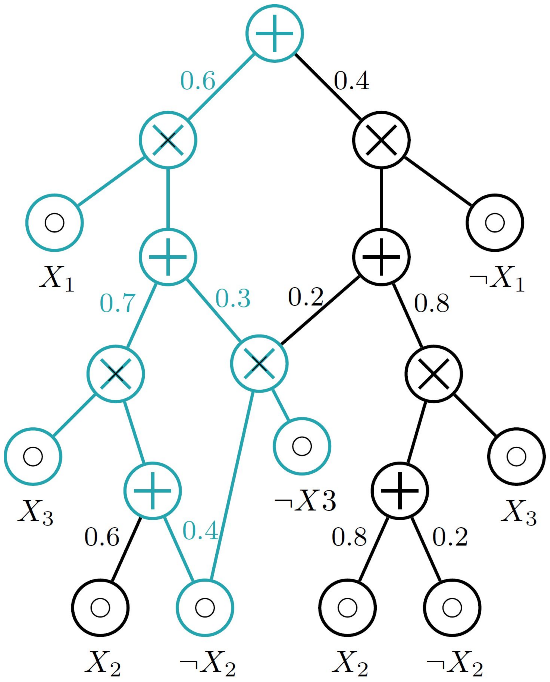
find: $argmax_{\boldsymbol{q}}\, p(\boldsymbol{q}|\boldsymbol{e})$

$\Rightarrow$ *i.e. MAP of a marginal distribution on **Q***

! ***NP^{PP}-complete*** *for PGMs*

! ***NP-hard*** *even for PCs tractable for marginals, MAP & entropy*

Pruning circuits



Any parts of circuit not relevant for MMAP state can be pruned away

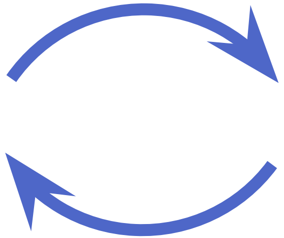
e.g. $p(X_1 = 1, X_2 = 0)$

We can find such edges in *linear time*

YooJung Choi, Tal Friedman and Guy Van den Broeck. Solving Marginal MAP Exactly by Probabilistic Circuit Transformations, 2021.

Iterative MMAP solver

Prune edges



Tighten bounds

Dataset	runtime (# solved) search	runtime (# solved) pruning
NLTCS	**0.01** (10)	0.63 (10)
MSNBC	**0.03** (10)	0.73 (10)
KDD	**0.04** (10)	0.68 (10)
Plants	2.95 (10)	**2.72** (10)
Audio	2041.33 (6)	**13.70 (10)**
Jester	2913.04 (2)	**14.74 (10)**
Netflix	– (0)	**47.18 (10)**
Accidents	109.56 (10)	**15.86** (10)
Retail	**0.06** (10)	0.81 (10)
Pumsb-star	2208.27 (7)	**20.88 (10)**
DNA	– (0)	**505.75 (9)**
Kosarek	48.74 (10)	**3.41** (10)
MSWeb	1543.49 (10)	**1.28** (10)
Book	– (0)	**46.50 (10)**
EachMovie	– (0)	**1216.89 (8)**
WebKB	– (0)	**575.68 (10)**
Reuters-52	– (0)	**120.58 (10)**
20 NewsGrp.	– (0)	**504.52 (9)**
BBC	– (0)	**2757.18 (3)**
Ad	– (0)	**1254.37 (8)**

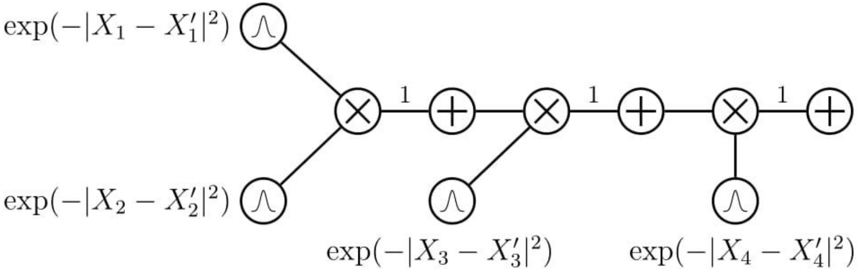
YooJung Choi, Tal Friedman and Guy Van den Broeck. Solving Marginal MAP Exactly by Probabilistic Circuit Transformations, 2021.

Tractable Computation of Expected Kernels

- How to compute the expected kernel given two distributions **p**, **q**?

$$\mathbb{E}_{\mathbf{x}\sim\mathbf{p},\mathbf{x}'\sim\mathbf{q}}[\mathbf{k}(\mathbf{x},\mathbf{x}')]$$

- Circuit representation for kernel functions, e.g., $\mathbf{k}(\mathbf{x},\mathbf{x}') = \exp\left(-\sum_{i=1}^{4} \mid X_i - X_i' \mid^2\right)$



Wenzhe Li, Zhe Zeng, Antonio Vergari and Guy Van den Broeck. Tractable Computation of Expected Kernels, *UAI*, 2021.

Tractable Computation of Expected Kernels: Applications

- Reasoning about support vector regression (SVR) with missing features

$$\underbrace{\mathbb{E}_{\mathbf{x}_m \sim \mathbf{p}(\mathbf{X}_m|\mathbf{x}_o)}}_{\text{missing features}} \underbrace{\left[\sum_{i=1}^{m} w_i \mathbf{k}(\mathbf{x}_i, \mathbf{x}) + b\right]}_{\text{SVR model}}$$

- Collapsed Black-box Importance Sampling: minimize kernelized Stein discrepancy

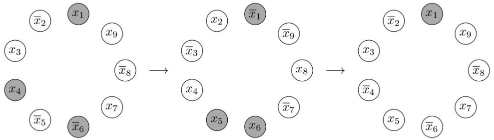
importance weights $\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left\{ \boldsymbol{w}^\top \underbrace{\boldsymbol{K}_{p,\mathbf{s}}}_{\text{expected kernel matrix}} \boldsymbol{w} \,\middle|\, \sum_{i=1}^{n} w_i = 1,\ w_i \geq 0 \right\}$

Wenzhe Li, Zhe Zeng, Antonio Vergari and Guy Van den Broeck. Tractable Computation of Expected Kernels, *UAI*, 2021.

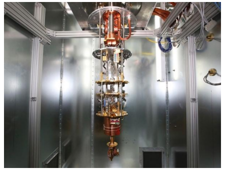
Dice Probabilistic Programming Language

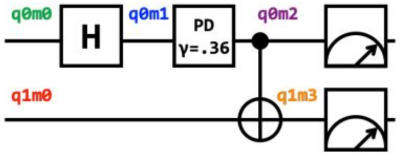
Probabilistic **Dice** Program → Symbolic Compilation → Probabilistic Circuit

As soon as *dice* was put online people started using it in surprising ways we had not foreseen



Probabilistic Model Checking
(verify randomized algorithms)



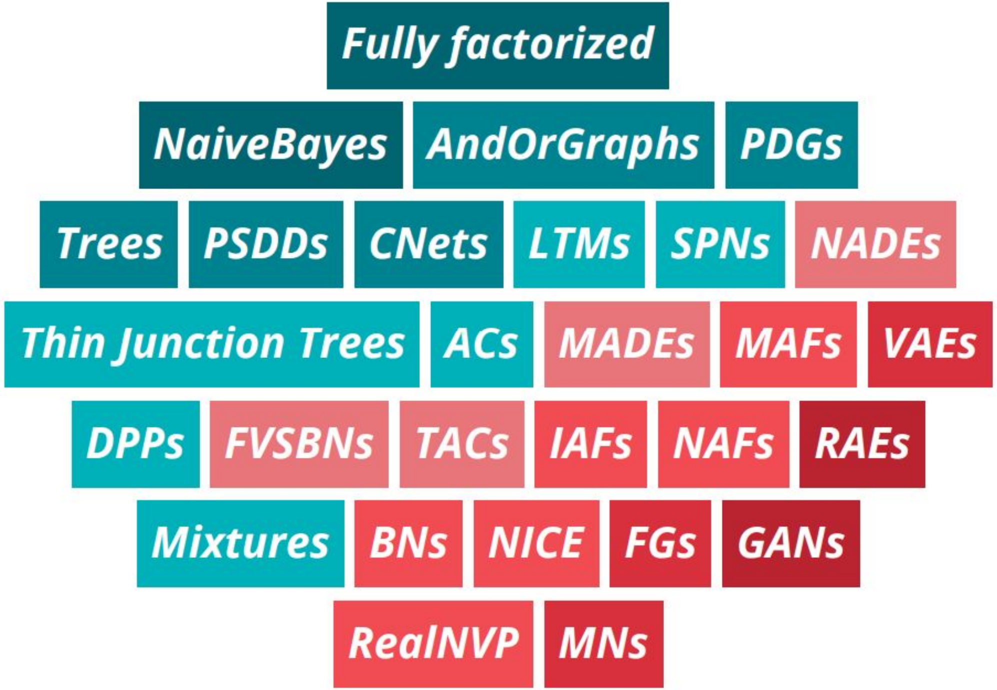


Quantum Simulation

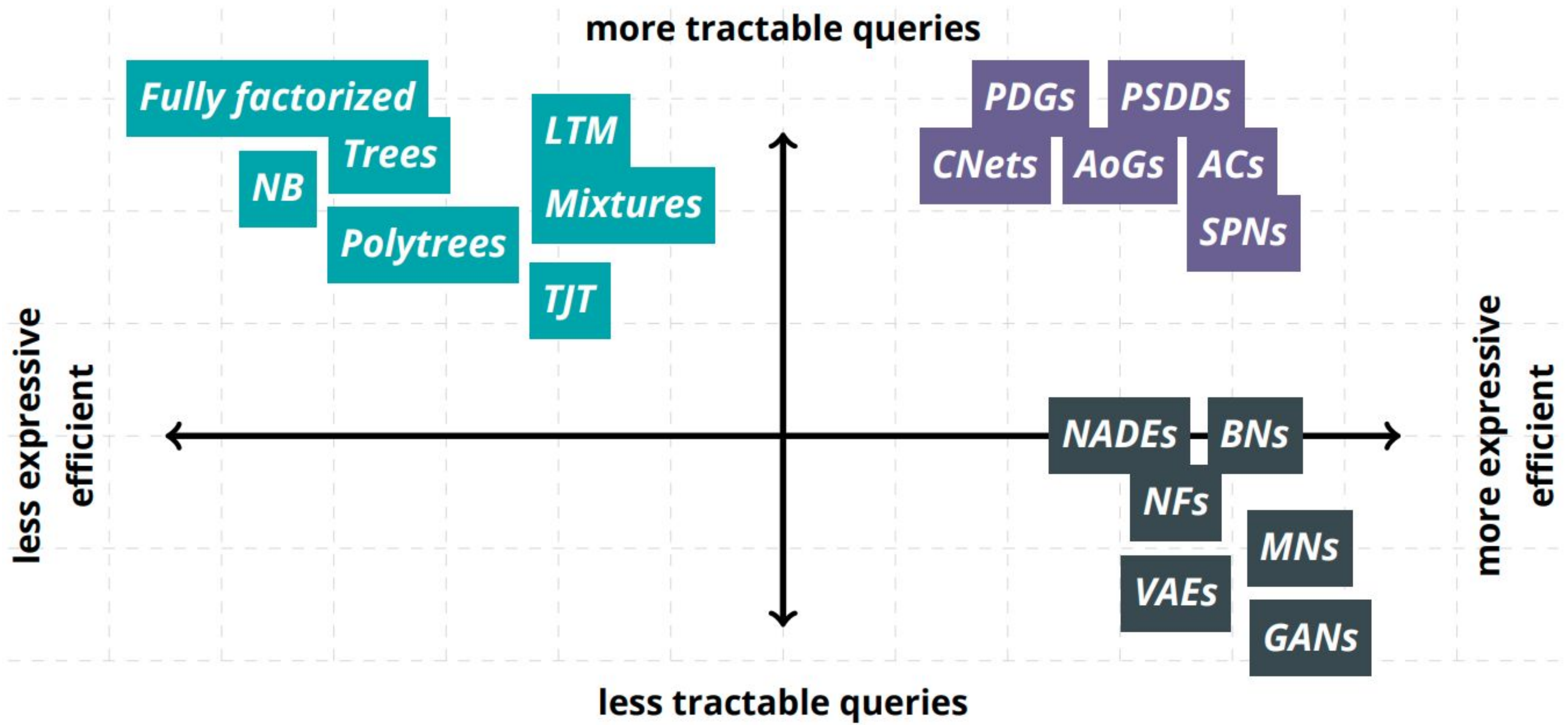
Steven Holtzen, Guy Van den Broeck and Todd Millstein. Scaling Exact Inference for Discrete Probabilistic Programs, *OOPSLA*, ACM, 2020.

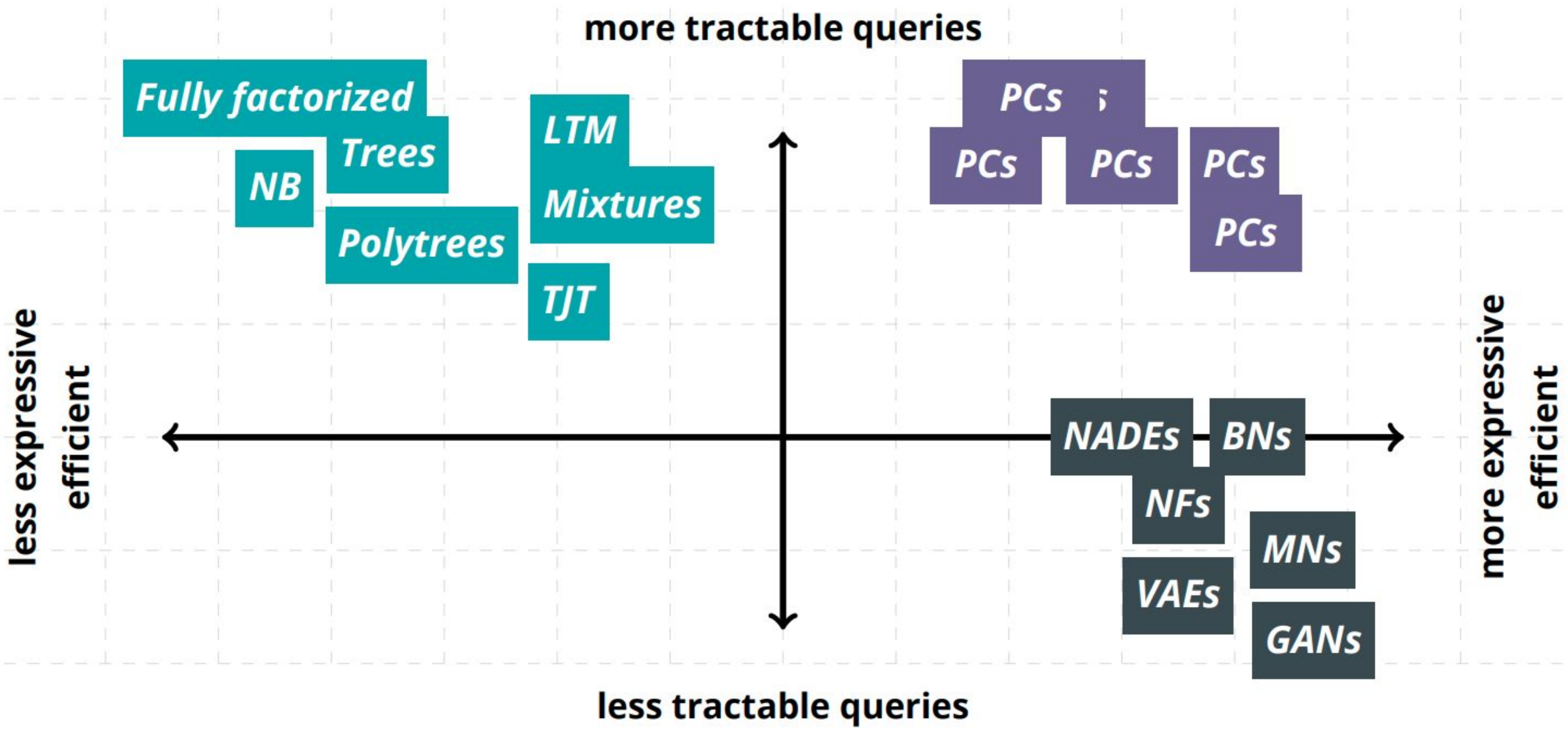
Conclusion

1. What are tractable probabilistic circuits?
2. Are these models any good?
3. What is their expressive power?
4. How far can we push tractable inference?



tractability is a spectrum


more tractable queries
less tractable queries
less expressive
efficient
more expressive
efficient
Fully factorized
NB
Trees
Polytrees
LTM
Mixtures
TJT
PDGs
PSDDs
CNets
AoGs
ACs
SPNs
NADEs
BNs
NFs
MNs
VAEs
GANs



more tractable queries
Fully factorized
Trees
NB
Polytrees
LTM
Mixtures
TJT
PCs
PCs
PCs
PCs
PCs
less expressive
efficient
more expressive
efficient
NADEs
BNs
NFs
MNs
VAEs
GANs
less tractable queries

Thanks

This was the work of many wonderful students/postdoc/collaborators!

References: http://starai.cs.ucla.edu/publications/