What's the difference between PGMs & Neural Networks?

antonio vergari (he/him)

@tetraduzione

16th Feb 2024 - **Probabilistic Modeling and Reasoning**

april

april is
probably a
recursive
identifier of a
lab

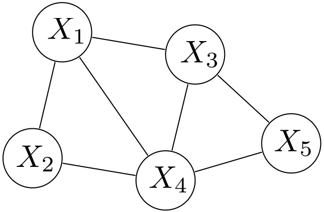
april

***a**bout*
***pr**obabilities*
***i**ntegrals &*
***l**ogic*

Probabilistic Graphical Models (PGMs)

Declarative semantics: a clean separation of modeling assumptions from inference

Nodes: random variables
Edges: dependencies



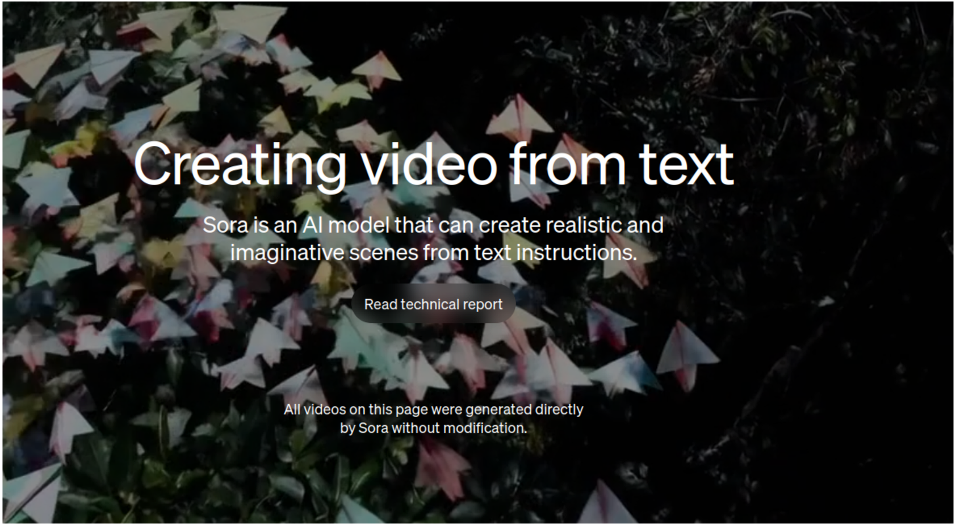
$+$

Inference:

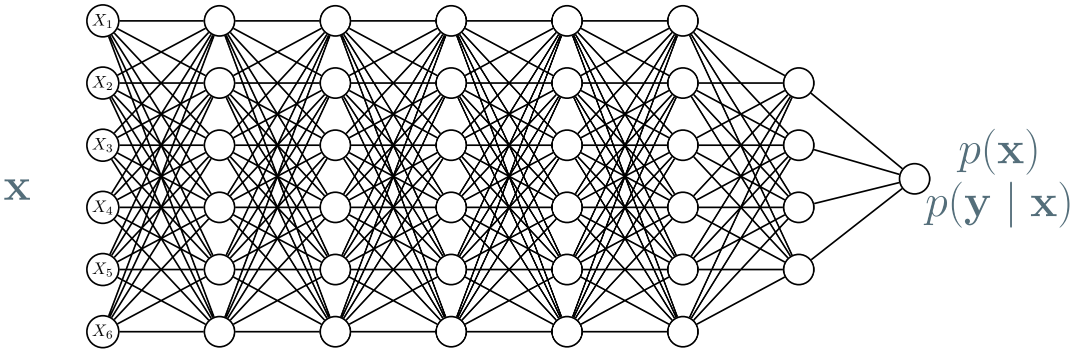
conditioning *[Darwiche 2001; Sang, Beame, and Kautz 2005]*

elimination *[Zhang and Poole 1994; Dechter 1998]*

message passing *[Yedidia, Freeman, and Weiss 2001; Dechter, Kask, and Mateescu 2002; Choi and Darwiche 2010; Sontag, Globerson, and Jaakkola 2011]*


Creating video from text
Sora is an AI model that can create realistic and imaginative scenes from text instructions.
Read technical report
All videos on this page were generated directly by Sora without modification.


MOGWAI
HA PGMs WILL NEVER DIE,
BUT YOU WILL.



...but what about neural networks?

...so what's the difference?

NNs are ***graphs*** (and ***can encode joint/conditional distributions***), but ...

	PGMs	***Neural Networks***
Nodes:	random variables	unit of computations
Edges:	dependencies	order of execution
Inference:	conditioning	feedforward pass
	elimination	backward pass
	message passing	

...so what's the difference?

NNs are ***graphs*** (and ***can encode joint/conditional distributions***), but ...

	PGMs	***Neural Networks***
Nodes:	random variables	unit of computations
Edges:	dependencies	order of execution
Inference:	conditioning	feedforward pass
	elimination	backward pass
	message passing	

$\Longrightarrow$ *they are **computational graphs***

Goal

“Can we find a middle ground between these two representations?”

Goal

"Can we design computational graphs that efficiently encode inference procedures in PGMs?"

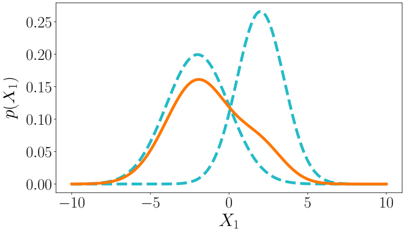
Goal

"Can we design computational graphs that efficiently encode inference procedures in PGMs?"

$\Rightarrow$ ***yes! with circuits!***

GMMs

as computational graphs

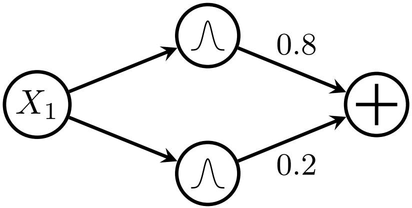


$$p(X) = w_1 \cdot p_1(X_1) + w_2 \cdot p_2(X_1)$$

$\Longrightarrow$ *translating inference to data structures...*

GMMs

as computational graphs

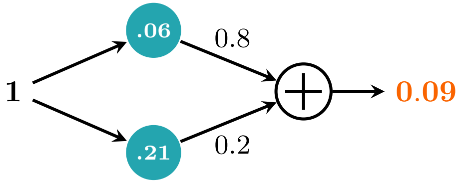


$$p(X_1) = 0.2 \cdot p_1(X_1) + 0.8 \cdot p_2(X_1)$$

$\Rightarrow$ *...e.g., as a weighted sum unit over Gaussian input distributions*

GMMs

as computational graphs

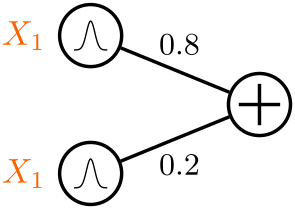


$$p(X=1) = 0.2 \cdot p_1(X_1 = 1) + 0.8 \cdot p_2(X_1 = 1)$$

$\Longrightarrow$ *inference = feedforward evaluation*

GMMs

as computational graphs



A simplified notation:

$\Rightarrow$ ***scopes*** *attached to inputs*

$\Rightarrow$ *edge directions omitted*

GMMs

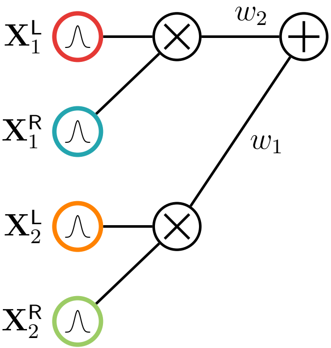
as computational graphs

$$p(\mathbf{X}) = w_1 \cdot p_1(\mathbf{X}_1^{\mathsf{L}}) \cdot p_1(\mathbf{X}_1^{\mathsf{R}}) + w_2 \cdot p_2(\mathbf{X}_2^{\mathsf{L}}) \cdot p_2(\mathbf{X}_2^{\mathsf{R}})$$

$\Longrightarrow$ *local factorizations...*

GMMs

as computational graphs



$$p(\mathbf{X}) = w_1 \cdot p_1(\mathbf{X}_1^{\mathsf{L}}) \cdot p_1(\mathbf{X}_1^{\mathsf{R}}) + w_2 \cdot p_2(\mathbf{X}_2^{\mathsf{L}}) \cdot p_2(\mathbf{X}_2^{\mathsf{R}})$$

$\Rightarrow$ *...are product units*

Probabilistic Circuits (PCs)

A grammar for tractable computational graphs

I. *A simple tractable function is a circuit*

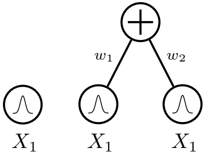
X_1

Probabilistic Circuits (PCs)

A grammar for tractable computational graphs

I. *A simple tractable function is a circuit*

II. *A weighted combination of circuits is a circuit*



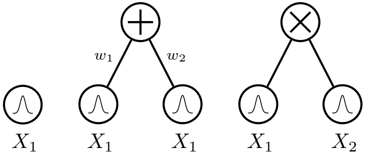
Probabilistic Circuits (PCs)

A grammar for tractable computational graphs

I. *A simple tractable function is a circuit*

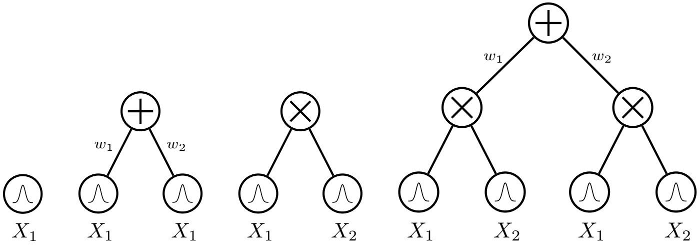
II. *A weighted combination of circuits is a circuit*

III. *A product of circuits is a circuit*



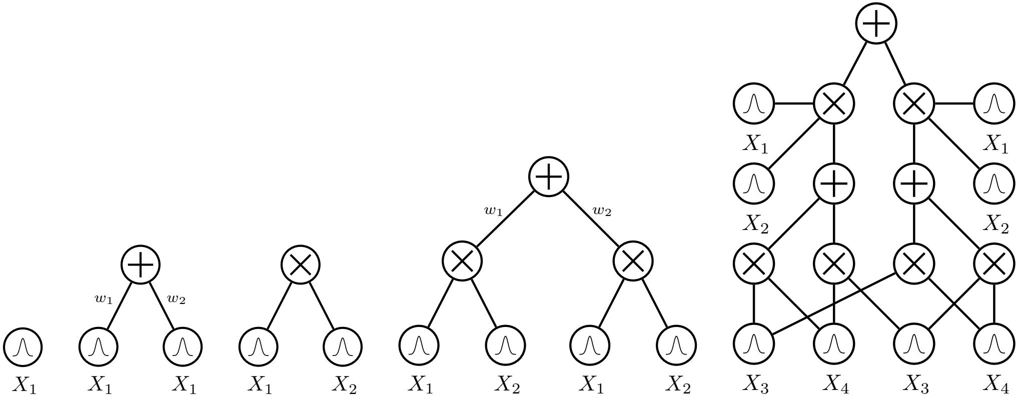
Probabilistic Circuits (PCs)

A grammar for tractable computational graphs



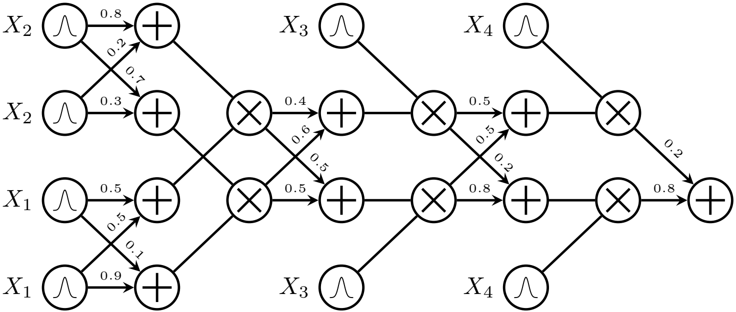
Probabilistic Circuits (PCs)

A grammar for tractable computational graphs



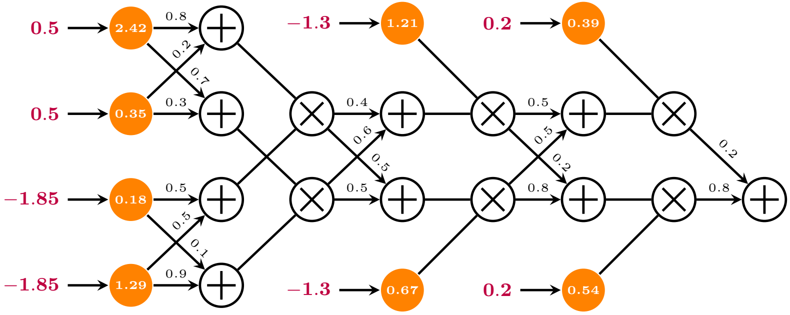
Probabilistic queries = feedforward evaluation

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



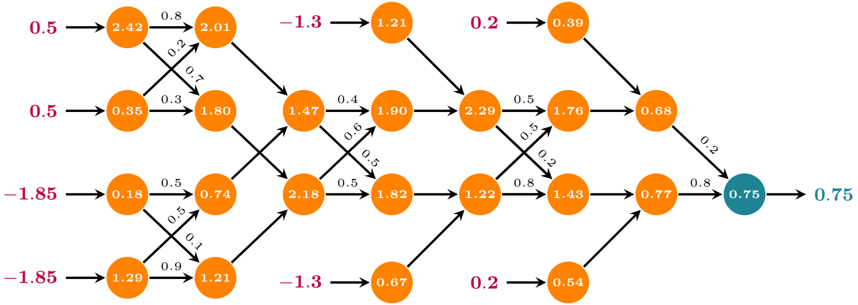
Probabilistic queries = feedforward evaluation

$$p(X_1 = -1.85, X_2 = 0.5, X_3 = -1.3, X_4 = 0.2)$$



Probabilistic queries = feedforward evaluation

$$p(X_1 = -1.85, X_2 = 0.5, X_3 = -1.3, X_4 = 0.2) = 0.75$$



...why PCs?

1. A grammar for tractable models

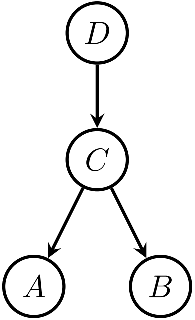
One formalism to represent many probabilistic and logical models

$\Longrightarrow$ *#HMMs #Trees #XGBoost, Tensor Networks, ...*

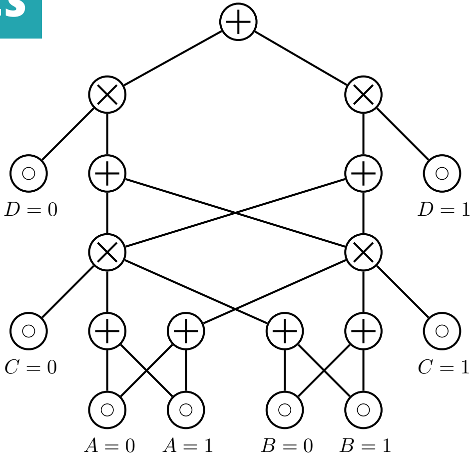
and other *PGMs*...

From PGMs to circuits

via compilation

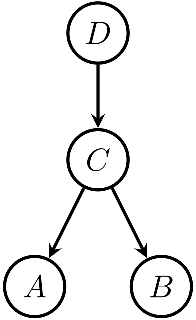


$\rightarrow$



From PGMs to circuits

via compilation

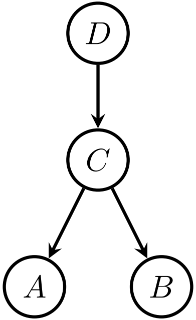


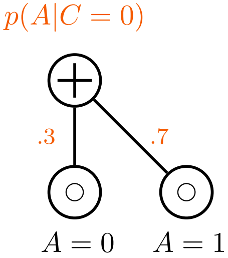
Bottom-up **compilation**: starting from leaves...

From PGMs to circuits

via compilation

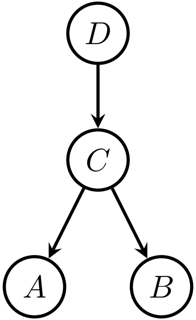
…compile a leaf CPT





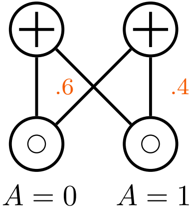
From PGMs to circuits

via compilation



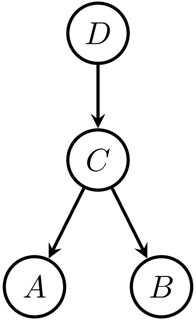
...compile a leaf CPT

$p(A|C = 1)$

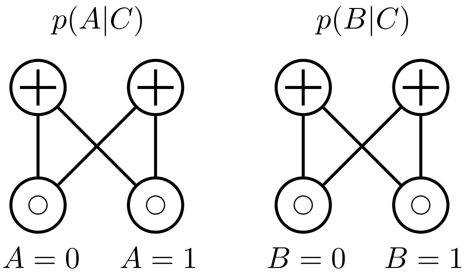


From PGMs to circuits

via compilation

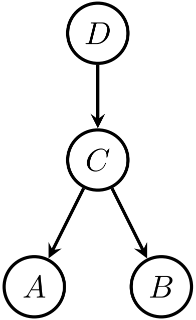


...compile a leaf CPT...for all leaves...

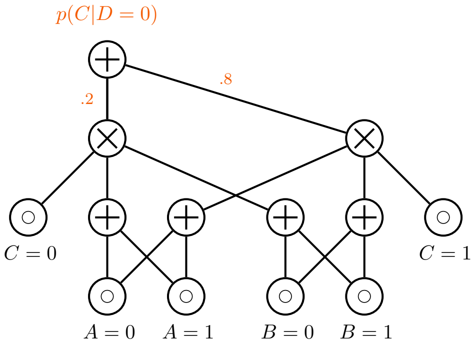


From PGMs to circuits

via compilation



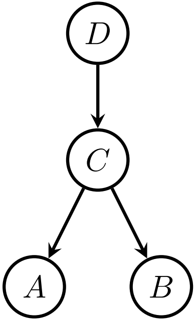
...and recurse over parents...

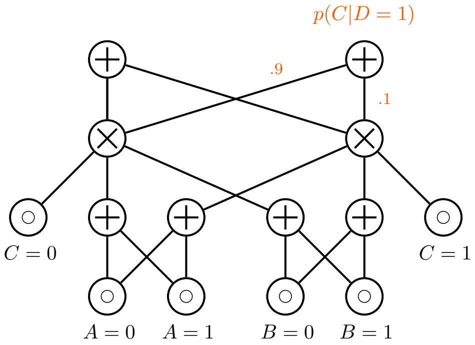


From PGMs to circuits

via compilation

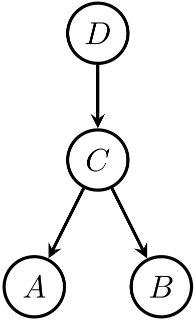
...while reusing previously compiled nodes!...

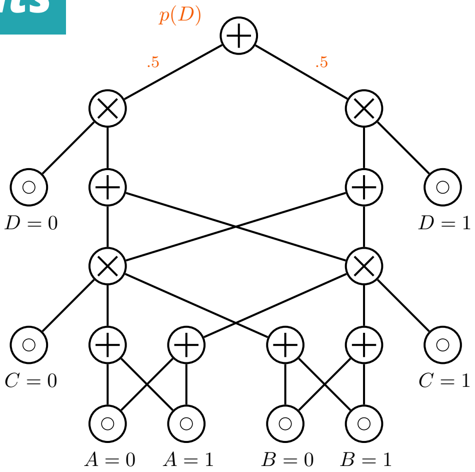




From PGMs to circuits

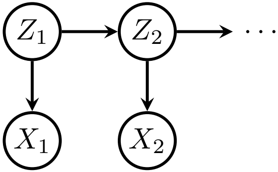
via compilation

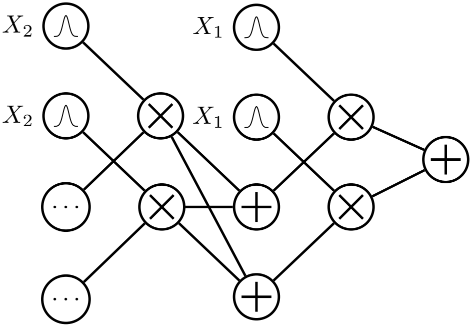


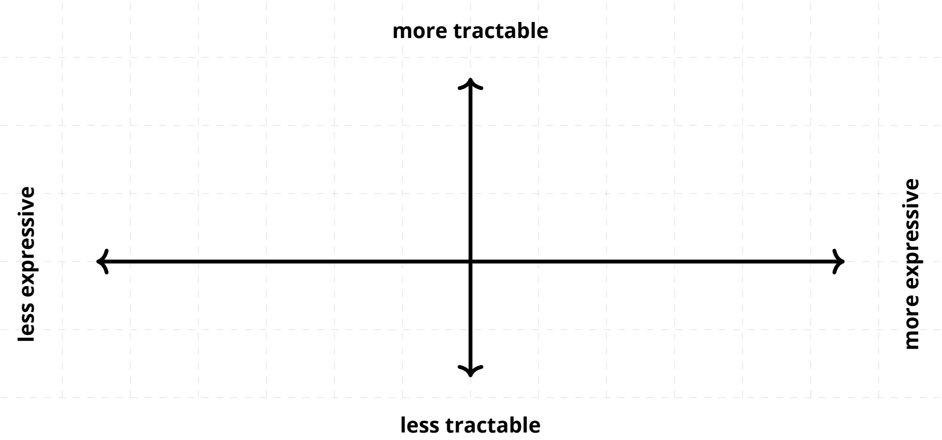


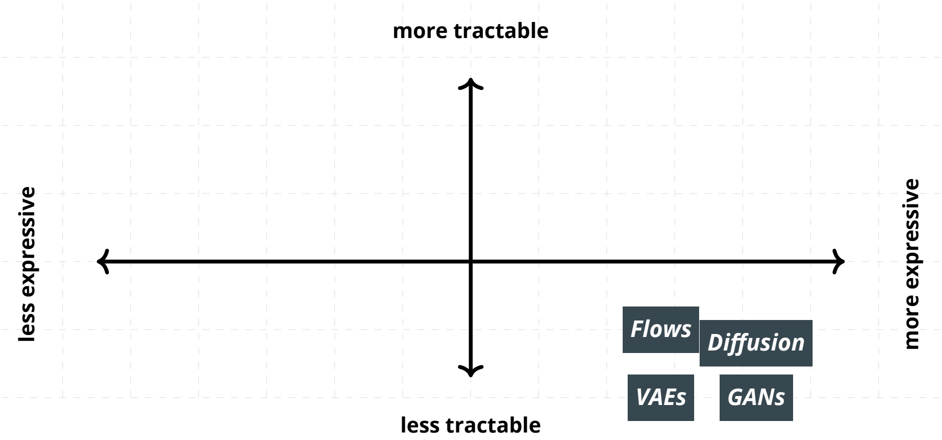
HMMs

as computational graphs

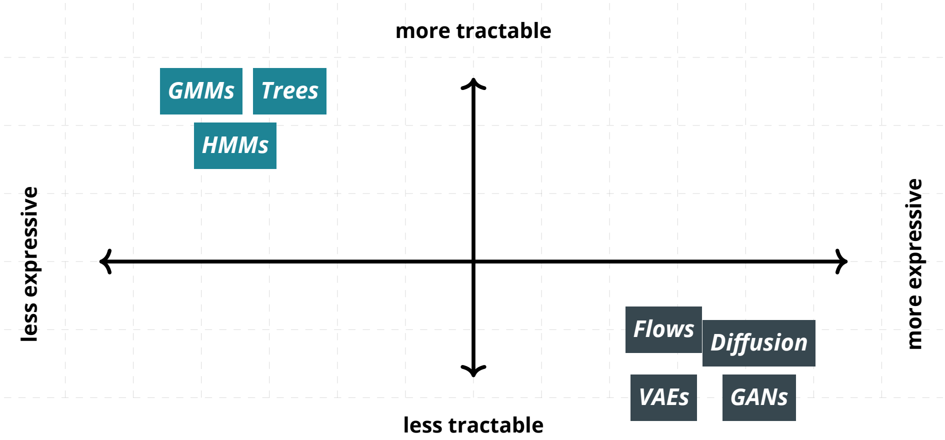




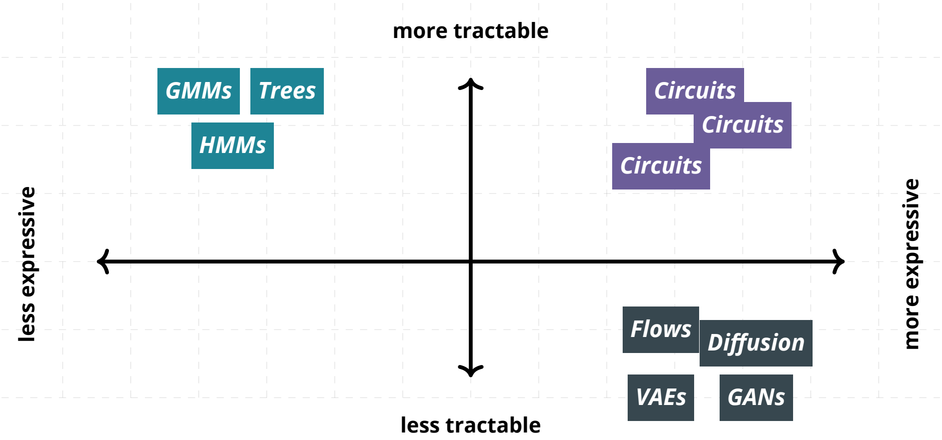

more tractable
less expressive
more expressive
less tractable



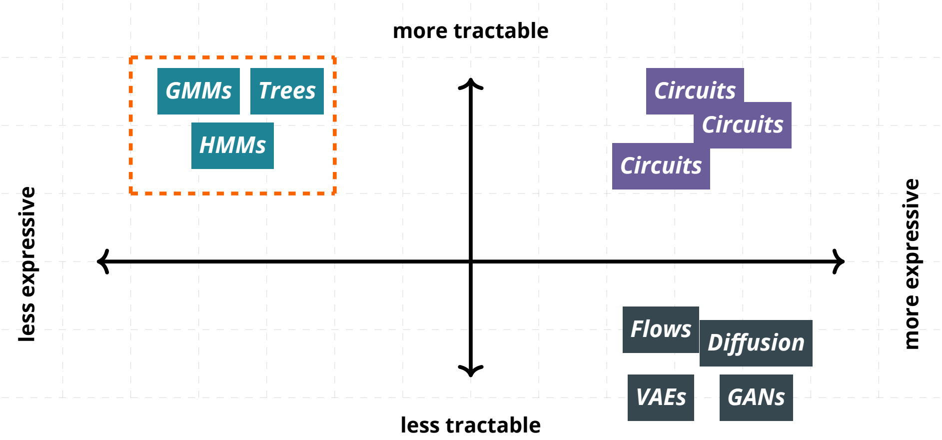
Expressive models are not much tractable...



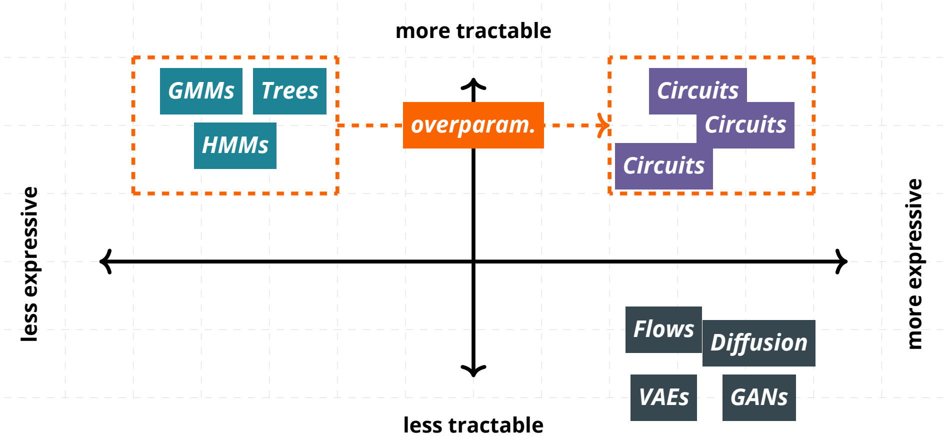
Tractable models are not that expressive...



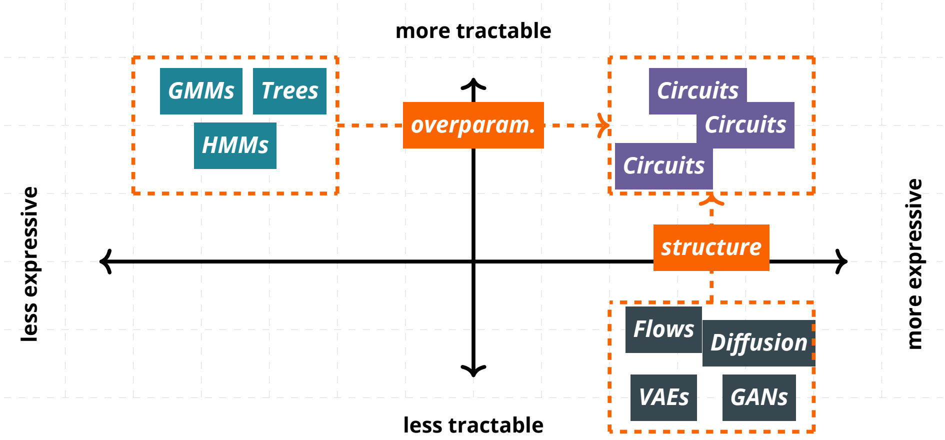
Circuits can be both expressive and tractable!



Start simple...



then make it more expressive!



impose structure!

...why PCs?

1. A grammar for tractable models

One formalism to represent many probabilistic and logical models

$\Longrightarrow$ *#HMMs #Trees #XGBoost, Tensor Networks, ...*

and other *PGMs...*

...why PCs?

1. A grammar for tractable models

One formalism to represent many probabilistic and logical models

$\Longrightarrow$ *#HMMs #Trees #XGBoost, Tensor Networks, ...*

and other *PGMs*...

2. Expressiveness

Competitive with intractable models, VAEs, Flow...*#hierachical #mixtures #polynomials*

How expressive?

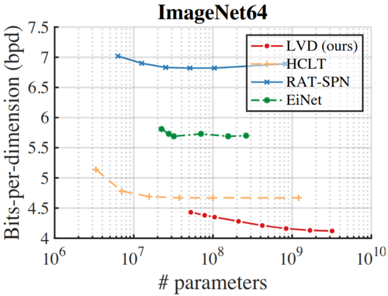
Dataset	Sparse PC (ours)	HCLT	RatSPN	IDF	BitSwap	BB-ANS	McBits
MNIST	**1.14**	1.20	1.67	1.90	1.27	1.39	1.98
EMNIST(MNIST)	**1.52**	1.77	2.56	2.07	1.88	2.04	2.19
EMNIST(Letters)	**1.58**	1.80	2.73	1.95	1.84	2.26	3.12
EMNIST(Balanced)	**1.60**	1.82	2.78	2.15	1.96	2.23	2.88
EMNIST(ByClass)	**1.54**	1.85	2.72	1.98	1.87	2.23	3.14
FashionMNIST	**3.27**	3.34	4.29	3.47	3.28	3.66	3.72

competitive with Flows and VAEs!

Dang, Liu, and Broeck, "Sparse Probabilistic Circuits via Pruning and Growing", NeurIPS, 2022

How scalable?

Dataset	TPMs				DGMs		
	LVD (ours)	HCLT	EiNet	RAT-SPN	Glow	RealNVP	BIVA
ImageNet32	**4.39**±0.01	4.82	5.63	6.90	4.09	4.28	3.96
ImageNet64	**4.12**±0.00	4.67	5.69	6.82	3.81	3.98	-
CIFAR	**4.38**±0.02	4.61	5.81	6.95	3.35	3.49	3.08



up to billions of parameters

Liu, Zhang, and Broeck, "Scaling Up Probabilistic Circuits by Latent Variable Distillation", arXiv preprint, 2022

...why PCs?

1. A grammar for tractable models

One formalism to represent many probabilistic and logical models

$\Longrightarrow$ *#HMMs #Trees #XGBoost, Tensor Networks, ...*

and other *PGMs*...

2. Expressiveness

Competitive with intractable models, VAEs, Flow...*#hierachical #mixtures #polynomials*

3. Tractability == Structural Properties!!!

Exact computations of reasoning tasks are certified by guaranteeing certain structural properties. *#marginals #expectations #MAP, #product ...*

Structural properties

smoothness

decomposability

determinism

compatibility

Vergari et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021

Structural properties

property A

property B

property C

property D

Vergari *et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021*

Structural properties

property A

property B

property C

property D

tractable computation of ***arbitrary integrals***

$$p(\mathbf{y}) = \sum_{\mathsf{val}(\mathbf{Z})} p(\mathbf{z}, \mathbf{y}), \quad \forall \mathbf{Y} \subseteq \mathbf{X}, \quad \mathbf{Z} = \mathbf{X} \setminus \mathbf{Y}$$

$\Rightarrow$ ***sufficient*** *and* ***necessary*** *conditions for a single feedforward evaluation*

$\Rightarrow$ *tractable partition function*

Vergari *et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021*

Structural properties

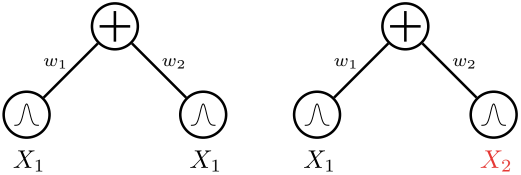
smoothness

decomposability

compatibility

determinism

the inputs of sum units are defined over the same variables



smooth* circuit** ***non-smooth circuit

__Vergari__ et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021

Structural properties

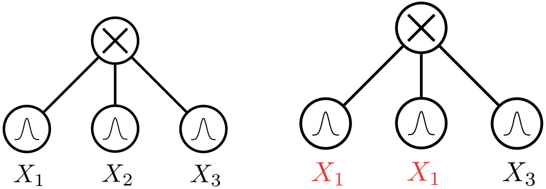
smoothness

decomposability

compatibility

determinism

the inputs of prod units are defined over disjoint variable sets

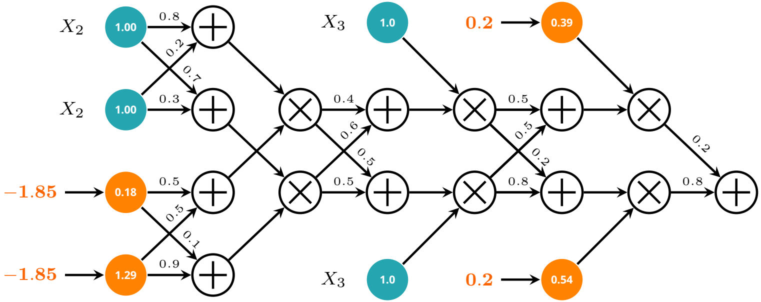


decomposable* circuit** ***non-decomposable circuit

__Vergari__ et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021

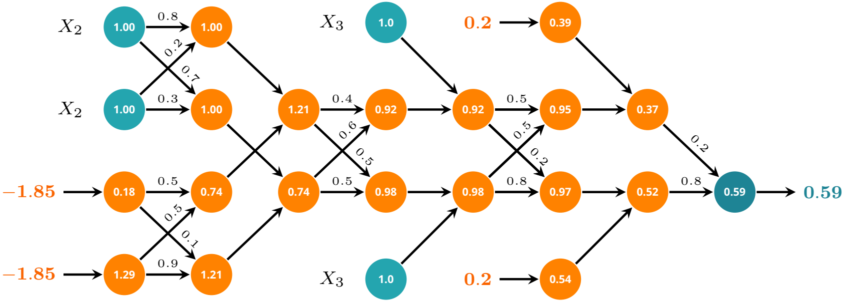
Probabilistic queries = feedforward evaluation

$$p(X_1 = -1.85, X_4 = 0.2)$$



Probabilistic queries = feedforward evaluation

$$p(X_1 = -\mathbf{1.85}, X_4 = \mathbf{0.2})$$



smooth + decomposable circuits = ...

Computing arbitrary integrations (or summations)

$\Longrightarrow$ *linear in circuit size!*

E.g., suppose we want to compute Z:

$$\int \boldsymbol{p}(\mathbf{x}) d\mathbf{x}$$

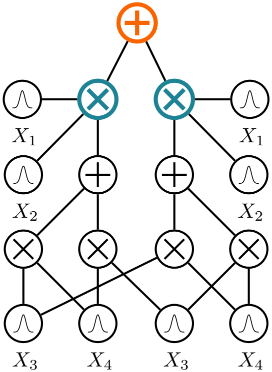
smooth + decomposable circuits = ...

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

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

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

$\Longrightarrow$ *integrals are "pushed down" to inputs*



smooth + decomposable circuits = ...

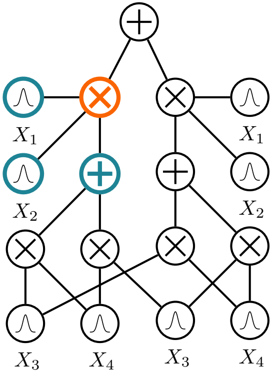
If $p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{x})p(\mathbf{y})p(\mathbf{z})$, (***decomposability***):

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

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

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

$\Longrightarrow$ *integrals decompose into easier ones*



...wait!

"Are all PGMs circuits?"

and/or

"Are all circuits PGMs?"

...not so easy!

1. Marginal inference in PGMs is exponential in the treewidth!

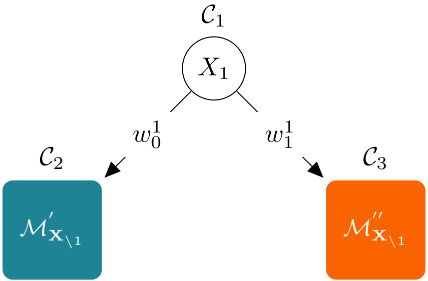
but PCs can exploit context specific independence

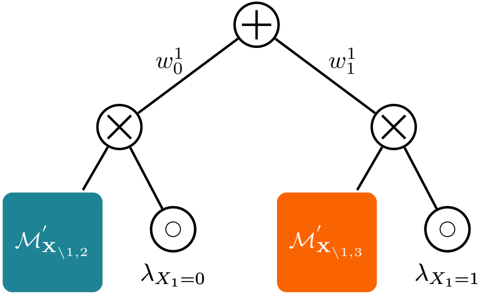
$$\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid Z = z_1$$

but

$$\mathbf{X} \not\!\perp\!\!\!\perp \mathbf{Y} \mid Z = z_2$$

Decision trees as PCs...





...not so easy!

1. Marginal inference in PGMs is exponential in the treewidth!
but PCs can exploit **context specific independence**

2. We do not know how to compile exactly an arbitrary PGM over continuous vars!
we need to extend the language of PCs to **integral units**

Probabilistic Integral Circuits

Gennaro Gala[1] Cassio de Campos[1] Robert Peharz[1,2] Antonio Vergari[3] Erik Quaeghebeur[1]

PCs

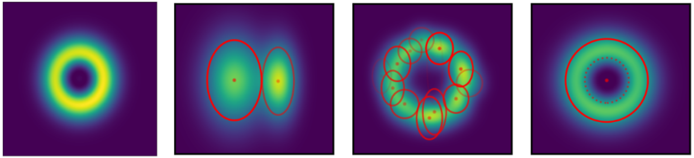
(and more circuits)

everywhere

Subtractive Mixture Models via Squaring: Representation and Learning

Lorenzo Loconte[1] Aleksanteri M. Sladek[2] Stefan Mengel[3]

Martin Trapp[2] Arno Solin[2] Nicolas Gillis[4] Antonio Vergari[1]



GMM ($K = 2$) GMM ($K = 16$) nGMM^2 ($K = 2$)

PGMs with negative parameters!
spotlight at ICLR 2024 (top 5% papers)

How to Turn Your Knowledge Graph Embeddings into Generative Models via Probabilistic Circuits

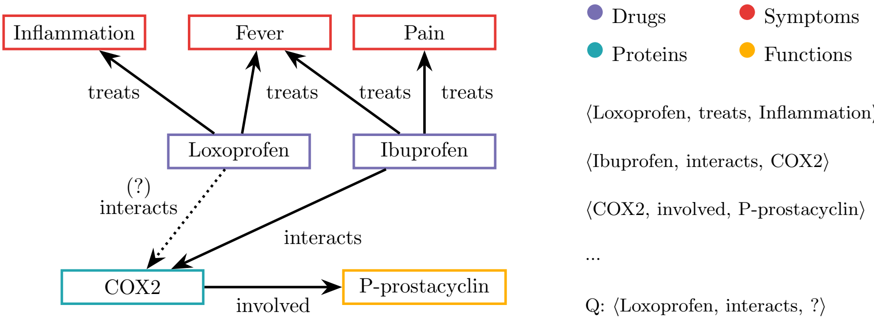
Lorenzo Loconte
University of Edinburgh, UK
l.loconte@sms.ed.ac.uk

Nicola Di Mauro
University of Bari, Italy
nicola.dimauro@uniba.it

Robert Peharz
TU Graz, Austria
robert.peharz@tugraz.at

Antonio Vergari
University of Edinburgh, UK
avergari@ed.ac.uk

PCs meet knowledge graph embedding models
oral at NeurIPS 2023 (top 0.6% papers)



⟨Loxoprofen, treats, Inflammation⟩

⟨Ibuprofen, interacts, COX2⟩

⟨COX2, involved, P-prostacyclin⟩

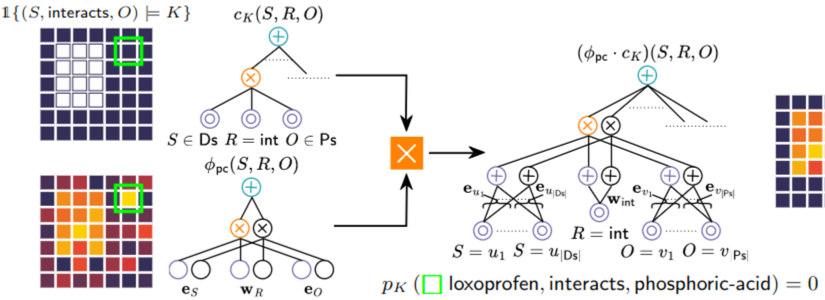
...

Q: ⟨Loxoprofen, interacts, ?⟩

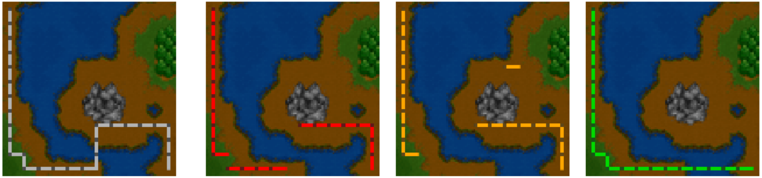
A: ⟨Loxoprofen, interacts, **phosphoric-acid**⟩ !!!

neural link predictors can violate domain constraints

PCs+KGE


$\mathbb{1}\{(S, \text{interacts}, O) \models K\}$
$c_K(S,R,O)$
$S \in \text{Ds}$ $R = \text{int}$ $O \in \text{Ps}$
$\phi_{\text{pc}}(S,R,O)$
$\mathbf{e}_S$ $\mathbf{w}_R$ $\mathbf{e}_O$
$(\phi_{\text{pc}} \cdot c_K)(S,R,O)$
$\mathbf{e}_{u_1}$ $\mathbf{e}_{u_{|\text{Ds}|}}$ $\mathbf{w}_{\text{int}}$ $\mathbf{e}_{v_1}$ $\mathbf{e}_{v_{|\text{Ps}|}}$
$R = \text{int}$
$S = u_1$ $S = u_{|\text{Ds}|}$ $O = v_1$ $O = v_{|\text{Ps}|}$
p_K (loxoprofen, interacts, phosphoric-acid) $= 0$

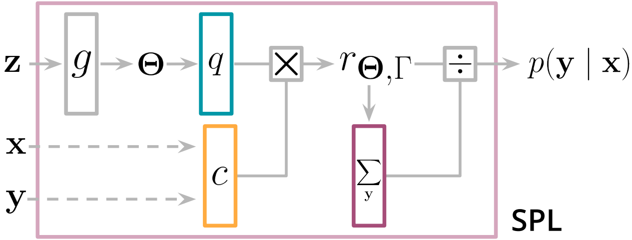
Semantic Probabilistic Layers for Neuro-Symbolic Learning



GROUND TRUTH FIL $\mathcal{L}_{\mathsf{SL}}$ SPL

injecting constraints in deep learning (NeurIPS 2022)

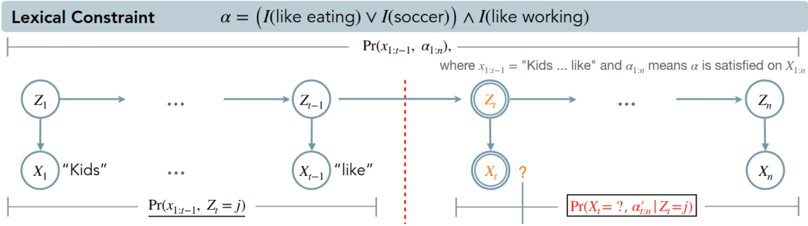
$$p(\mathbf{y} \mid \mathbf{x}) = q_{\boldsymbol{\Theta}}(\mathbf{y} \mid g(\mathbf{z})) \cdot c_{\mathsf{K}}(\mathbf{x}, \mathbf{y}) / \mathcal{Z}(\mathbf{x})$$



efficient and reliable reasoning over constraints

Tractable Control for Autoregressive Language Generation

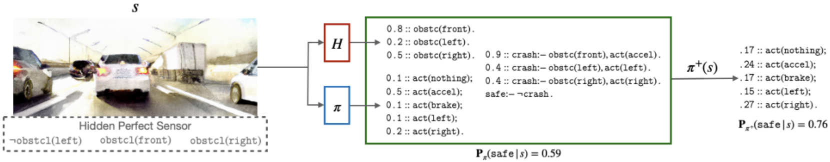
Honghua Zhang [*1] Meihua Dang [*1] Nanyun Peng [1] Guy Van den Broeck [1]



constrained text generation with LLMs (ICML 2023)

Safe Reinforcement Learning via Probabilistic Logic Shields

Wen-Chi Yang[1], Giuseppe Marra[1], Gavin Rens and Luc De Raedt[1,2]



reliable reinforcement learning (AAAI 23)

more reasoning scenarios

q_1 *"What is the **expected prediction** for a patient with unavailable records?"*

q_2 *"How **fair** is the prediction is a certain protected attribute changes?"*

q_3 *"Can we certify no **adversarial examples** exist?"*

...asking queries to a ML model

more reasoning scenarios

q1 $\mathbb{E}_{\mathbf{x}_m \sim p(\mathbf{X}_m|\mathbf{x}_o)}\left[f(\mathbf{x}_o, \mathbf{x}_m)\right]$

(expected prediction)

q2 $\mathbb{E}_{\mathbf{x}_c \sim p(\mathbf{X}_c|X_s=0)}\left[f_0(\mathbf{x}_c)\right] - \mathbb{E}_{\mathbf{x}_c \sim p(\mathbf{X}_c|X_s=1)}\left[f_1(\mathbf{x}_c)\right]$

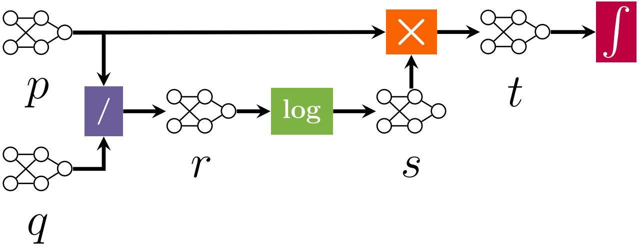
(fairness)

q3 $\mathbb{E}_{\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_D)}\left[f(\mathbf{x} + \mathbf{e})\right]$

(adversarial robust.)

...into math expressions over circuits

$$\int p(\mathbf{x}) \times \log \left(p(\mathbf{x}) \, / \, q(\mathbf{x}) \right) \, d\mathbf{X}$$



build a LEGO-like query calculus!

Vergari *et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021*

	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

compositionally *derive the tractability of many more queries*

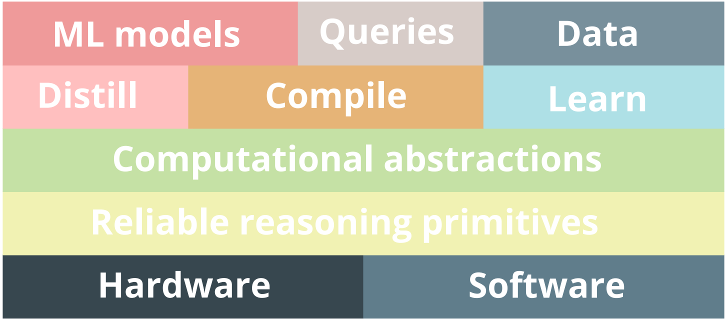
Vergari *et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021*

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

*and **prove hardness** when some input properties are not satisfied*

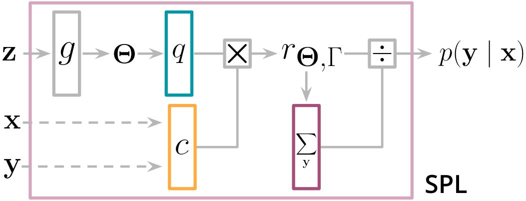
Vergari *et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021*

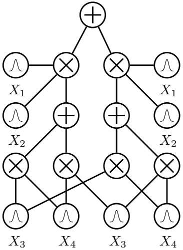
UNREAL



realizing a full “virtual machine” for reasoning


MOGWAI
PGMs
WILL NEVER DIE,
BUT YOU WILL.
and you can run and learn them
as neural networks





Ask me anything!