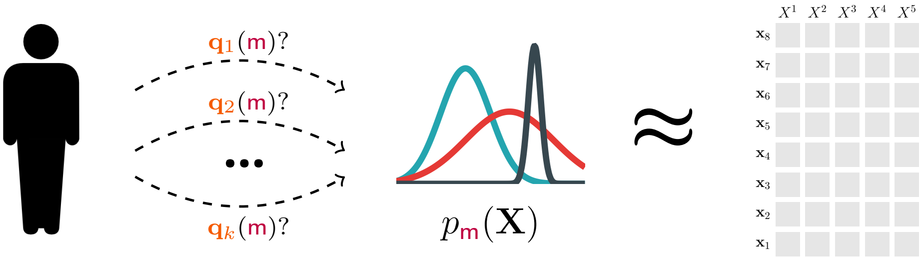
Integrating Logic and Prob ML with probabilistic circuits

antonio vergari (he/him)

@tetraduzione

29th June 2024 - **Nordic Prob AI Summer School** - *Copenhagen*

in the previous episodes…

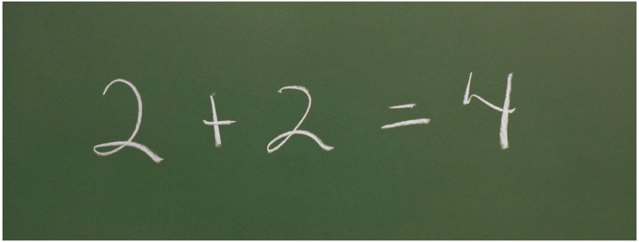


(generative) models that can reason probabilistically

...but some events are certain!

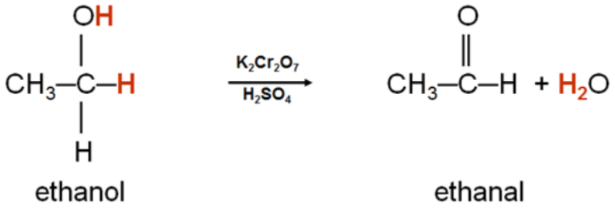
math reasoning

and logical deduction



Constraints: carrying out arithmetic tasks, but also ***proving theorems***

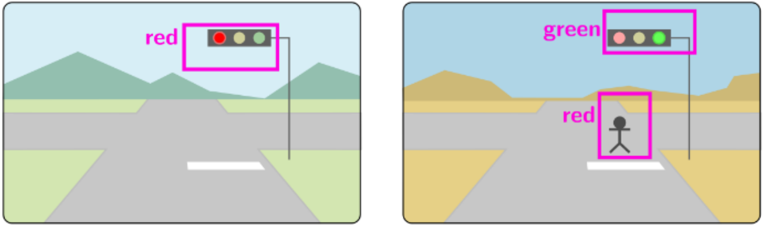
physics laws



Constraints: preserving #atoms, #electrons (RedOx), ...in chemical reactions

AI safety

$$K_1 = (\mathsf{pedestrian} \vee \mathsf{red} \Rightarrow \mathsf{stop})$$



Constraints: traffic rules, scene understanding ...

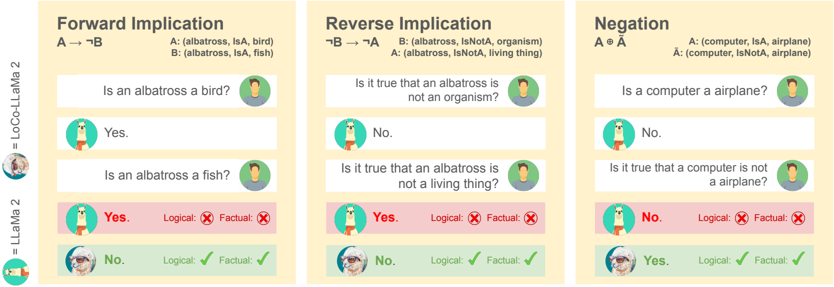
hard vs soft constraints

logic vs probabilities

logic	**prob logic**
"If X is a bird, X flies"	"If X is a bird, X might fly"
$A(X) \implies B(X)$	$\mathsf{p}(A(X) \implies B(X))$

***“but how bad
are purely neural models
when dealing with
hard constraints
in the real world?”***

logical inconsistency



LLMs confabulate and contraddict themselves [1]

[1] https://github.com/SuperBruceJia/Awesome-LLM-Self-Consistency

Calanzone, Teso, and Vergari, Towards Logically Consistent Language Models via Probabilistic Reasoning, , 2024

planning

Can Large Language Models Reason and Plan?

Subbarao Kambhampati
School of Computing & Augmented Intelligence
Arizona State University
email: rao@asu.edu

Spoiler: *"To summarize, nothing that I have read, verified, or done gives me any compelling reason to believe that LLMs do reasoning/planning, as normally understood.."*

Kambhampati, "Can large language models reason and plan?", , 2024

what about valid molecules?

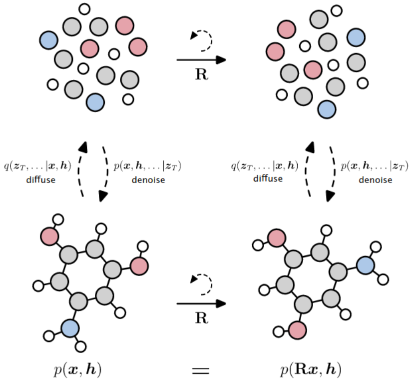
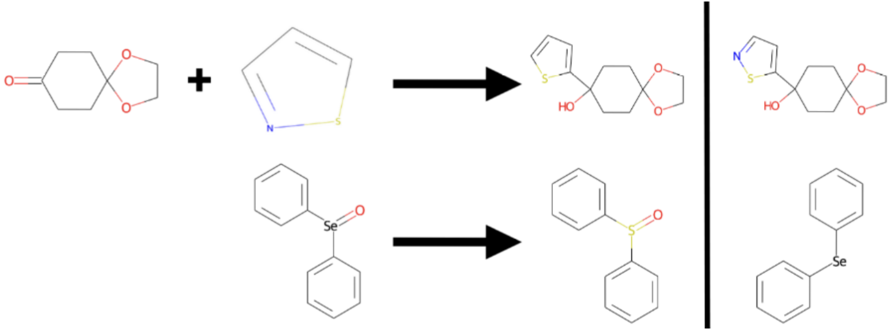


Table 2. Validity and uniqueness over 10000 molecules with standard deviation across 3 runs. Results marked (*) are not directly comparable, as they do not use 3D coordinates to derive bonds. H: model hydrogens explicitly

Method	H	Valid (%)	Valid and Unique (%)
Graph VAE (*)		55.7	42.3
GTVAE (*)		74.6	16.8
Set2GraphVAE (*)		59.9±1.7	56.2±1.4
EDM (ours)		**97.5**±0.2	**94.3**±0.2
E-NF	✓	40.2	39.4
G-Schnet	✓	85.5	80.3
GDM-aug	✓	90.4	89.5
EDM (ours)	✓	**91.9**±0.5	**90.7**±0.6
Data	✓	97.7	97.7

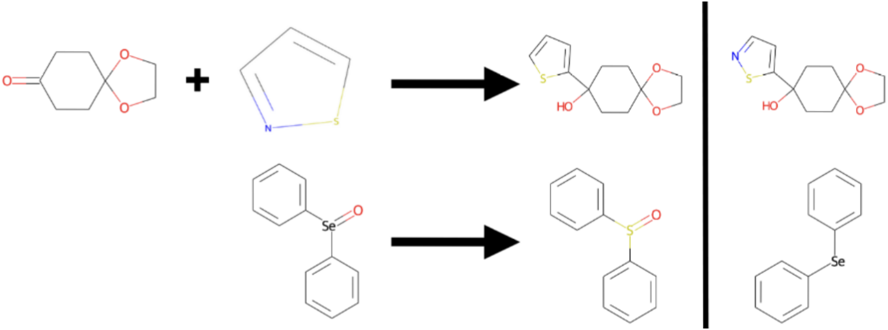
Hoogeboom et al., "Equivariant diffusion for molecule generation in 3d", International Conference on Machine Learning, 2022

and valid reactions?



"deep learning is doing alchemy"

and valid reactions?



CHEMALGEBRA: ALGEBRAIC REASONING BY PREDICTING CHEMICAL REACTIONS

the issues!

I) Logical constraints can be hard to represent in a unified way

$\Longrightarrow$ ***a single framework*** *for matching, paths, hierarchies, plans ...*

II) How to integrate logic and probabilities in a single architecture

$\Longrightarrow$ ***combining soft and hard constraints***

III) Logical constraints are piecewise constant functions!

$\Longrightarrow$ *differentiable almost everywhere but **gradient is zero**!*

the issues!

I) Logical constraints can be hard to represent in a unified way

$\Longrightarrow$ ***a single framework*** *for matching, paths, hierarchies, plans ...*

II) How to integrate logic and probabilities in a single architecture

$\Longrightarrow$ ***combining soft and hard constraints***

III) Logical constraints are piecewise constant functions!

$\Longrightarrow$ *differentiable almost everywhere but* ***gradient is zero****!*

the issues!

I) Logical constraints can be hard to represent in a unified way

$\Longrightarrow$ ***a single framework*** *for matching, paths, hierarchies, plans ...*

II) How to integrate logic and probabilities in a single architecture

$\Longrightarrow$ ***combining soft and hard constraints***

III) Logical constraints are piecewise constant functions!

$\Longrightarrow$ *differentiable almost everywhere but **gradient is zero**!*

solution

probabilistic neuro-symbolic AI

solution

probabilistic ***neuro-symbolic AI***

integrate probabilistic reasoning

solution

probabilistic neuro-symbolic AI

with deep neural nets

solution

probabilistic neuro-symbolic AI

and hard constraints

Goal

"How can neural nets reason and learn with symbolic constraints reliably and efficiently?"

Goal

"How can neural nets reason and learn with symbolic constraints reliably and efficiently?"

guarantee that predictions ***always satisfy*** constraints

Goal

"How can neural nets reason and learn with symbolic constraints reliably and efficiently?"

fast and ***exact*** gradients

hard vs soft constraints

logic vs probabilities

logic

"If X is a bird, X flies"

$$A(X) \implies B(X)$$

prob logic

"If X is a bird, X might fly"

$$\mathsf{p}(A(X) \implies B(X))$$

which logic?

or which kind of constraints to represent?

propositional logic (zeroth-order)

$$(a \wedge b) \vee d \implies c$$

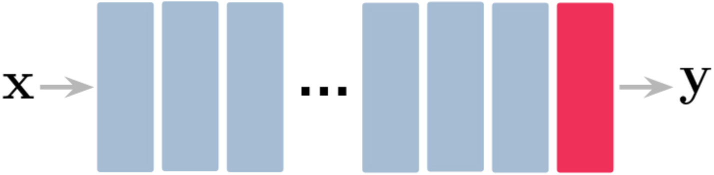
first-order logic (FOL)

$$\forall a \exists b : R(a, b) \vee Q(d) \implies C(x)$$

satisfiability modulo theory (SMT)

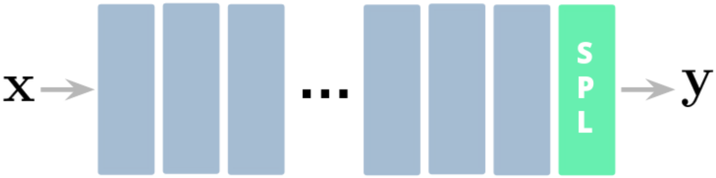
$$(\alpha X_i - \beta X_j \leq 100) \vee (X_j + X_k \geq 0) \implies (X_j X_k \leq X_i)$$

how to



make any neural network architecture...

how to



...guarantee all predictions to conform to constraints?

When?

Ground Truth

e.g. predict shortest path in a map

When?

Ground Truth

given $\mathbf{x}$ *// e.g. a tile map*

nesy structured output prediction (SOP) tasks

Vlastelica et al., "Differentiation of blackbox combinatorial solvers", , 2020

When?

Ground Truth

given $\mathbf{x}$ *// e.g. a tile map*

find $\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} p_\theta(\mathbf{y} \mid \mathbf{x})$ *// e.g. a configurations of edges in a grid*

nesy structured output prediction (SOP) tasks

Vlastelica et al., "Differentiation of blackbox combinatorial solvers", , 2020

When?

Ground Truth

given $\mathbf{x}$ *// e.g. a tile map*

find $\mathbf{y}^* = \text{argmax}_{\mathbf{y}}\, p_\theta(\mathbf{y} \mid \mathbf{x})$ *// e.g. a configurations of edges in a grid*

s.t. $\mathbf{y} \models \mathsf{K}$ *// e.g., that form a valid path*

nesy structured output prediction (SOP) tasks

Vlastelica et al., "Differentiation of blackbox combinatorial solvers", , 2020

When?

Ground Truth

given $\mathbf{x}$ *// e.g. a tile map*

find $\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} p_\theta(\mathbf{y} \mid \mathbf{x})$ *// e.g. a configurations of edges in a grid*

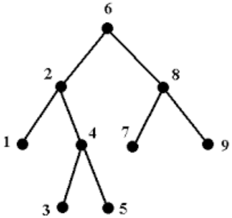
s.t. $\mathbf{y} \models \mathsf{K}$ *// e.g., that form a valid path*

// for a 12×12 grid, 2^{144} states but only 10^{10} valid ones!

nesy structured output prediction (SOP) tasks

Vlastelica et al., "Differentiation of blackbox combinatorial solvers", , 2020

When?



given $\mathbf{x}$ *// e.g. a feature map*

find $\mathbf{y}^* = \mathrm{argmax}_{\mathbf{y}}\, p_\theta(\mathbf{y} \mid \mathbf{x})$ *// e.g. labels of classes*

s.t. $\mathbf{y} \models \mathsf{K}$ *// e.g., constraints over superclasses*

$$\mathsf{K} : (Y_{\text{cat}} \implies Y_{\text{animal}}) \wedge (Y_{\text{dog}} \implies Y_{\text{animal}})$$

hierarchical multi-label classification

Giunchiglia and Lukasiewicz, "Coherent hierarchical multi-label classification networks", , 2020

When?

given $\mathbf{x}$ *// e.g. a user preference over $K - N$ sushi types*

find $\mathbf{y}^* = \mathrm{argmax}_{\mathbf{y}}\, p_\theta(\mathbf{y} \mid \mathbf{x})$ *// e.g. prefs over N more types*

s.t. $\mathbf{y} \models \mathsf{K}$ *// e.g., output valid rankings*

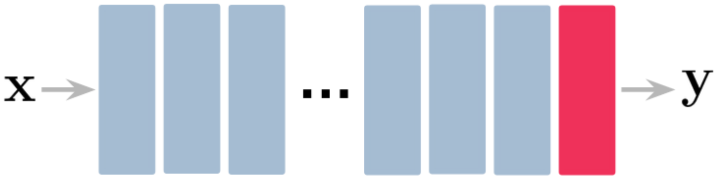
user preference learning

Choi, Van den Broeck, and Darwiche, "Tractable learning for structured probability spaces: A case study in learning preference distributions",
Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), 2015

How?

"which neural network
architecture
to use?"

e.g.,

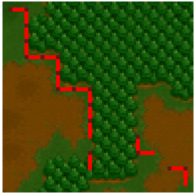


sigmoid linear layers

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{N} p(y_i \mid \mathbf{x})$$

When?

Ground Truth



ResNet-18

neural nets struggle to satisfy validity constraints!

Constraint losses

$$\mathcal{L}(\theta; \mathbf{x}, \mathbf{y}) + \lambda \mathcal{L}_{\mathsf{K}}(\mathbf{x}, \mathbf{y})$$

losses improve consistency during training...

Xu et al., "A Semantic Loss Function for Deep Learning with Symbolic Knowledge", Proceedings of the 35th International Conference on Machine Learning (ICML), 2018

Constraint losses

$$\mathcal{L}(\theta; \mathbf{x}, \mathbf{y}) + \lambda \mathcal{L}_{\mathsf{K}}(\mathbf{x}, \mathbf{y})$$

losses improve consistency during training...

e.g., the ***semantic loss***: $\mathcal{L}_{\mathsf{SL}} := -\log \sum_{\mathbf{y} \models \mathsf{K}} \prod_i p(Y_i \mid \mathbf{x})$

Xu et al., "A Semantic Loss Function for Deep Learning with Symbolic Knowledge", Proceedings of the 35th International Conference on Machine Learning (ICML), 2018

WMC

computing the probability of logical formulas

$$\sum_{\mathbf{y} \models \mathsf{K}} p(\mathbf{y}) = \sum_{\mathbf{y}} p(\mathbf{y}) \mathbb{1}\{\mathbf{y} \models \mathsf{K}\} = \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})}[\mathbb{1}\{\mathbf{y} \models \mathsf{K}\}]$$

*computing the **weighted model count** (WMC) of* K

Xu et al., "A Semantic Loss Function for Deep Learning with Symbolic Knowledge", Proceedings of the 35th International Conference on Machine Learning (ICML), 2018

WMC

computing the probability of logical formulas

$$\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathbb{1}\{\mathbf{y} \models \mathsf{K}\}] = \mathbb{P}(\mathsf{K}(\mathbf{y}))$$

computing the probability of K

Xu et al., "A Semantic Loss Function for Deep Learning with Symbolic Knowledge", Proceedings of the 35th International Conference on Machine Learning (ICML), 2018

WMC

computing the probability of logical formulas

$$\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathbb{1}\{\mathbf{y} \models \mathsf{K}\}] = \sum_{\mathbf{y}\models\mathsf{K}} \prod_{i:\mathbf{y}\models Y_i} w(Y_i) \prod_{i:\mathbf{y}\models\neg Y_i} (1 - w(Y_i))$$

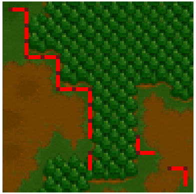
assuming independence of $\mathbf{y}$ (but be careful!)[2]

[2] *van Krieken et al., "On the Independence Assumption in Neurosymbolic Learning", 2024*

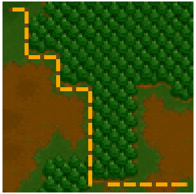
Xu et al., "A Semantic Loss Function for Deep Learning with Symbolic Knowledge", Proceedings of the 35th International Conference on Machine Learning (ICML), 2018

Constraint losses

Ground Truth



ResNet-18

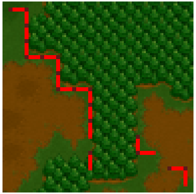


Semantic Loss

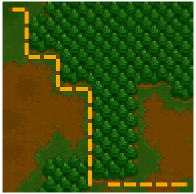
...but cannot guarantee consistency at test time!

SPL

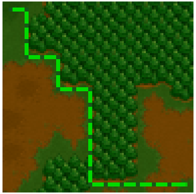
Ground Truth



ResNet-18



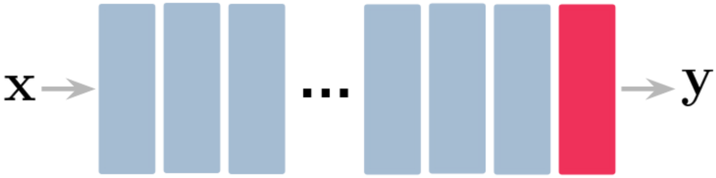
Semantic Loss



SPL (ours)

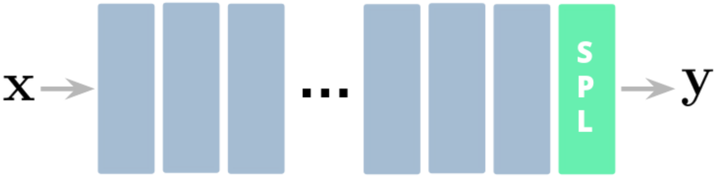
you can predict valid paths 100% of the time!

How?



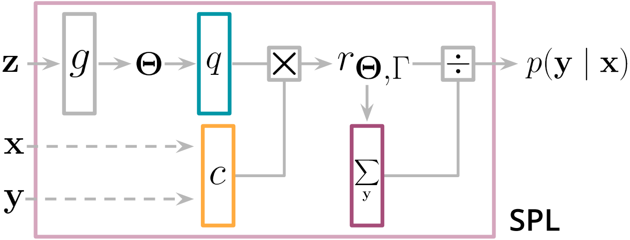
take an unreliable neural network architecture...

How?

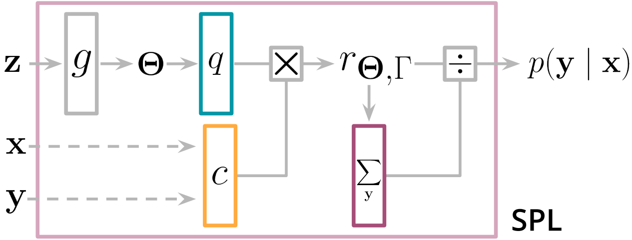


......and replace the last layer with a semantic probabilistic layer

SPL


$\mathbf{z}$
g
$\boldsymbol{\Theta}$
q
$r_{\boldsymbol{\Theta},\Gamma}$
$p(\mathbf{y} \mid \mathbf{x})$
$\mathbf{x}$
$\mathbf{y}$
c
$\sum_{\mathbf{y}}$
SPL

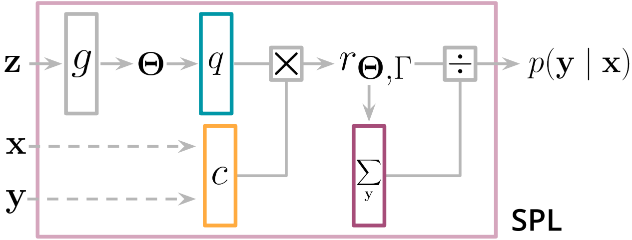
SPL



$$p(\mathbf{y} \mid \mathbf{x}) = q_{\Theta}(\mathbf{y} \mid g(\mathbf{z}))$$

$q_{\Theta}(\mathbf{y} \mid g(\mathbf{z}))$ is an expressive distribution over labels

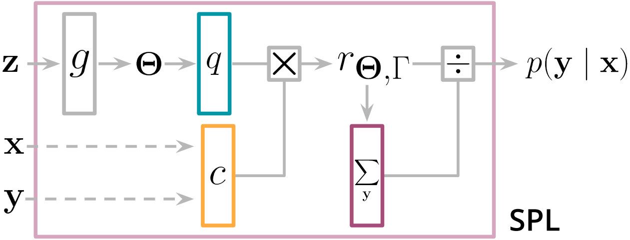
SPL



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

$c_{\mathsf{K}}(\mathbf{x}, \mathbf{y})$ encodes the constraint $\mathbb{1}\{\mathbf{x}, \mathbf{y} \models \mathsf{K}\}$

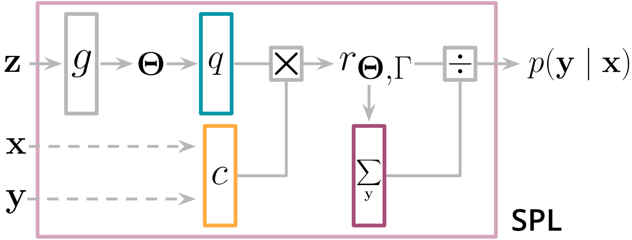
SPL



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

a product of experts : (

SPL



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

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

Goal

Can we design q and c
to be expressive models
yet yielding a tractable product?

Goal

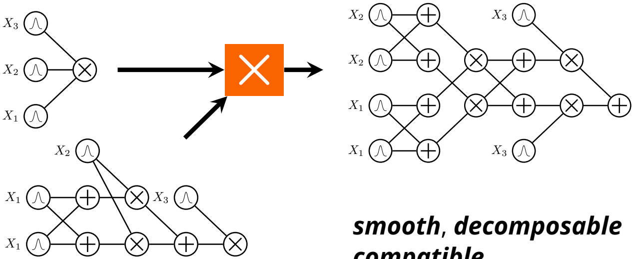
Can we design q and c
to be deep computational graphs
yet yielding a tractable product?

Goal

Can we design q and c
to be deep computational graphs
yet yielding a tractable product?

yes! as *circuits*!

Tractable products

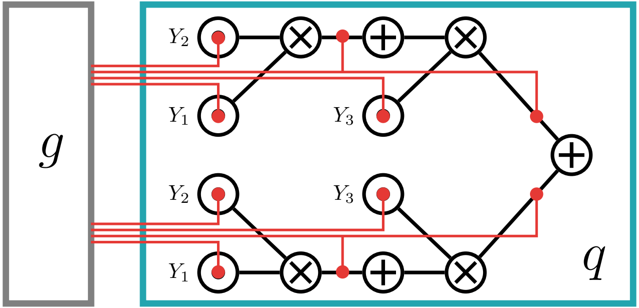


smooth, ***decomposable***
compatible

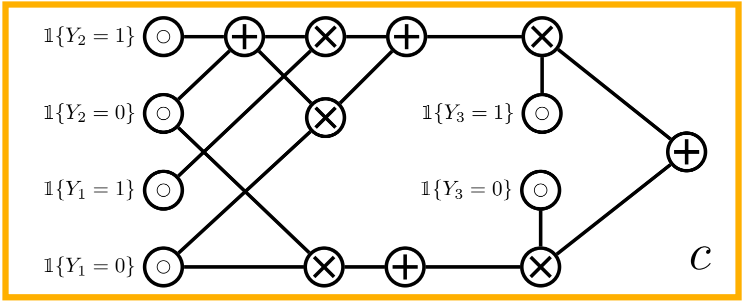
exactly compute $\mathcal{Z}$ in time $O(|\mathbf{q}||\mathbf{c}|)$

__Vergari__ et al., "A Compositional Atlas of Tractable Circuit Operations for Probabilistic Inference", NeurIPS, 2021

SPL

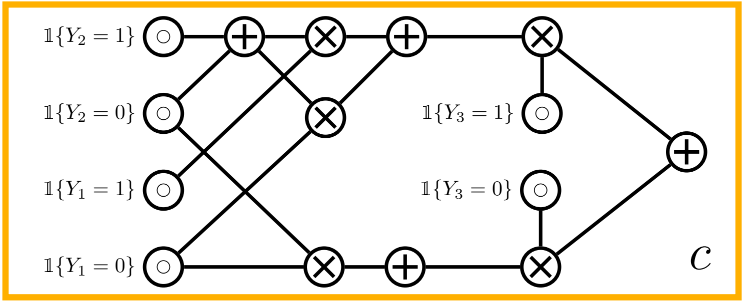


a conditional circuit $q(\mathrm{y}; \Theta = g(\mathrm{z}))$



and a logical circuit $c(\mathrm{y}, \mathrm{x})$ **encoding** K

SPL



compiling logical formulas into circuits

Knowledge compilation

$$\mathsf{K}:\ (Y_1 = 1 \implies Y_3 = 1)$$

$$\land\ \ (Y_2 = 1 \implies Y_3 = 1)$$

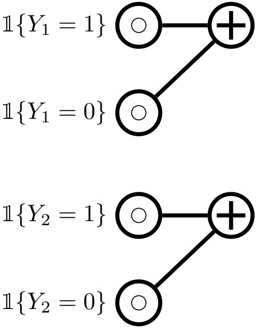
$\mathbb{1}\{Y_1 = 0\} \odot$ $\mathbb{1}\{Y_1 = 1\} \odot$ $\mathbb{1}\{Y_2 = 0\} \odot$

$\mathbb{1}\{Y_2 = 1\} \odot$ $\mathbb{1}\{Y_3 = 0\} \odot$ $\mathbb{1}\{Y_3 = 1\} \odot$

Pipatsrisawat and Darwiche, "New Compilation Languages Based on Structured Decomposability.", AAAI, 2008

Knowledge compilation

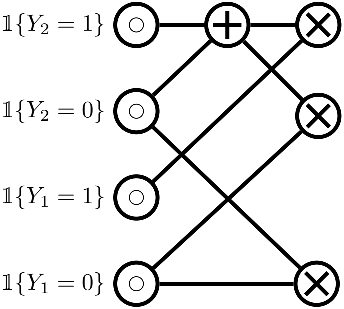
$$\mathsf{K}: \ (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \ (Y_2 = 1 \implies Y_3 = 1)$$



Pipatsrisawat and Darwiche, "New Compilation Languages Based on Structured Decomposability.", AAAI, 2008

Knowledge compilation

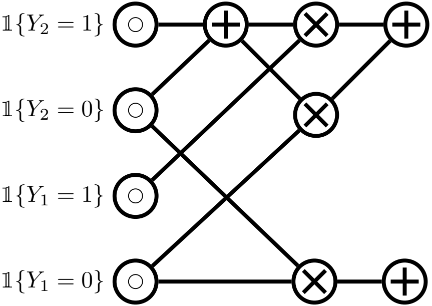
$$\mathsf{K}: \ (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \quad (Y_2 = 1 \implies Y_3 = 1)$$



Pipatsrisawat and Darwiche, "New Compilation Languages Based on Structured Decomposability.", AAAI, 2008

Knowledge compilation

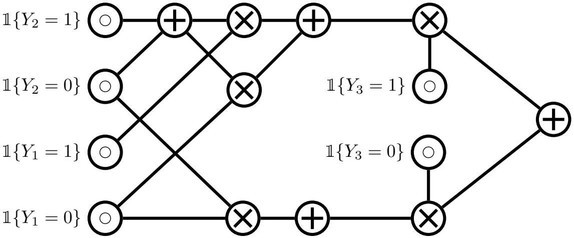
$$\mathsf{K}: \; (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \; (Y_2 = 1 \implies Y_3 = 1)$$



Pipatsrisawat and Darwiche, "New Compilation Languages Based on Structured Decomposability.", AAAI, 2008

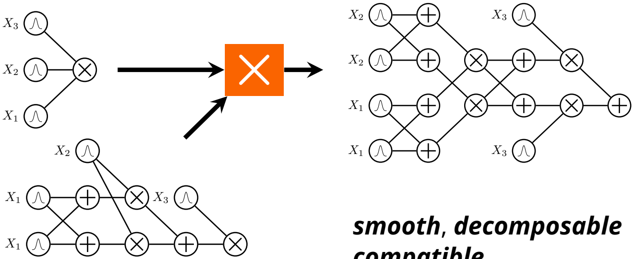
Knowledge compilation

$$\mathsf{K}: \; (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \; (Y_2 = 1 \implies Y_3 = 1)$$



Pipatsrisawat and Darwiche, "New Compilation Languages Based on Structured Decomposability.", AAAI, 2008

Tractable products



smooth, ***decomposable***
compatible

exactly compute $\mathcal{Z}$ in time $O(|\boldsymbol{q}||\boldsymbol{c}|)$

__Vergari__ et al., "A Compositional Atlas of Tractable Circuit Operations for Probabilistic Inference", NeurIPS, 2021

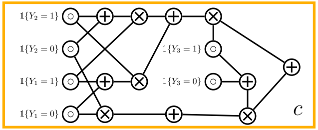
SPL recipe

$$\mathsf{K}: (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \quad (Y_2 = 1 \implies Y_3 = 1)$$

1) Take a
logical constraint

SPL recipe

$$\mathsf{K}: (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \quad (Y_2 = 1 \implies Y_3 = 1)$$



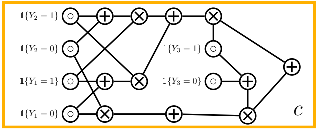
1) Take a logical constraint

2) Compile it into a constraint circuit

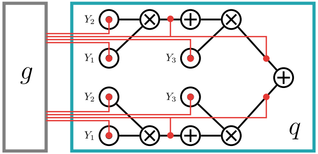
SPL recipe

$$K : (Y_1 = 1 \implies Y_3 = 1)$$
$$\land \quad (Y_2 = 1 \implies Y_3 = 1)$$

1) Take a logical constraint



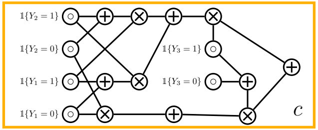
2) Compile it into a constraint circuit

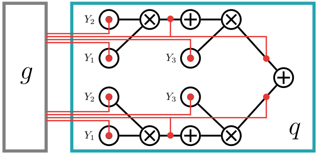


3) Multiply it by a circuit distribution

SPL recipe

$$\mathsf{K} : (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \quad (Y_2 = 1 \implies Y_3 = 1)$$





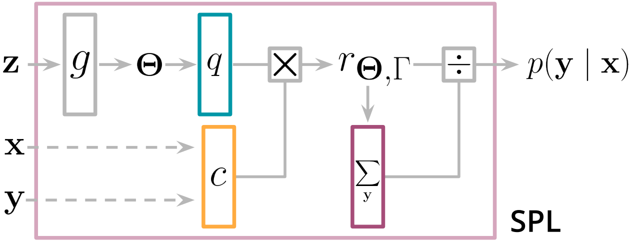
1) Take a
logical constraint

2) Compile it into
a constraint circuit

3) Multiply it
by a circuit distribution

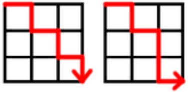
4) train end-to-end by sgd!

Experiments



how good are SPLs?

Experiments

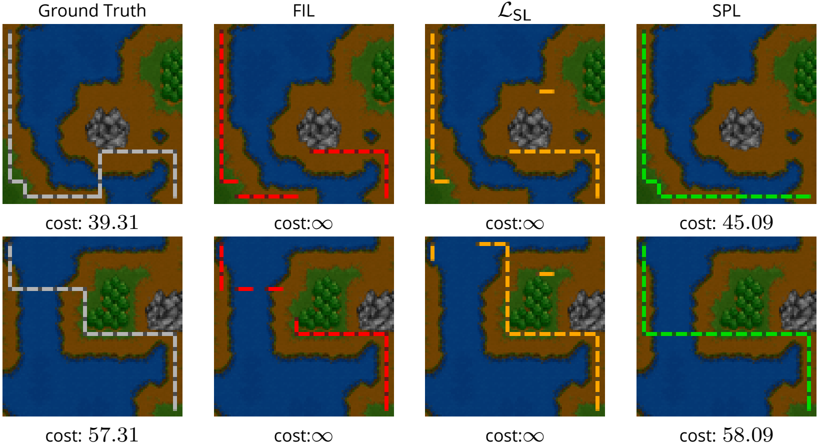


	Simple Path			Preference Learning		
Architecture	Exact	Hamming	Consistent	Exact	Hamming	Consistent
MLP+FIL	5.6	85.9	7.0	1.0	**75.8**	2.7
MLP+$\mathcal{L}_{\text{SL}}$	28.5	83.1	75.2	15.0	72.4	69.8
MLP+NeSyEnt	30.1	83.0	91.6	18.2	71.5	96.0
MLP+SPL	**37.6**	**88.5**	**100.0**	**20.8**	72.4	**100.0**

Experiments

Architecture	Exact	Hamming	Consistent
ResNet-18+FIL	55.0	**97.7**	56.9
ResNet-18+$\mathcal{L}_{\text{SL}}$	59.4	**97.7**	61.2
ResNet-18+SPL	**78.2**	96.3	**100.0**

Experiments



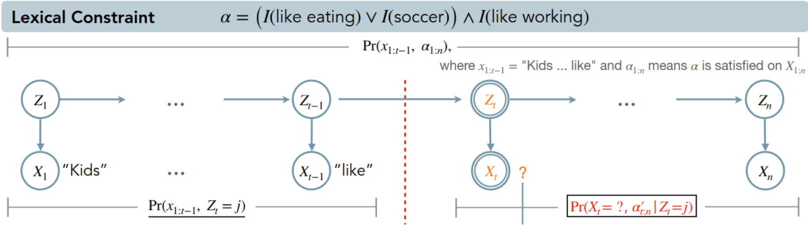
SPLs

(and more circuits)

everywhere

Tractable Control for Autoregressive Language Generation

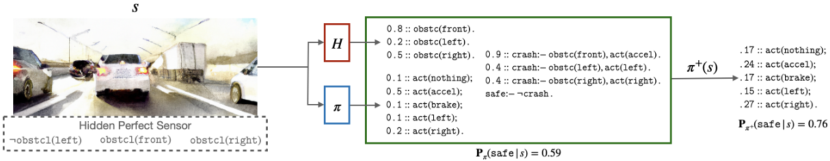
Honghua Zhang * [1] **Meihua Dang** * [1] **Nanyun Peng** [1] **Guy Van den Broeck** [1]



constrained text generation with LLMs (ICML 2023)

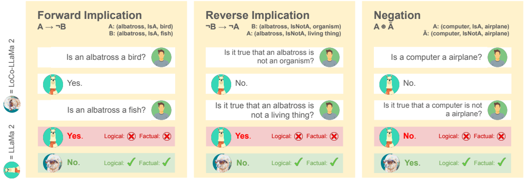
Safe Reinforcement Learning via Probabilistic Logic Shields

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



reliable reinforcement learning (AAAI 23)

Logically Consistent Language Models via Neuro-Symbolic Integration



improving logical (self-)consistency in LLMs (under submission)

How to Turn Your Knowledge Graph Embeddings into Generative Models

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

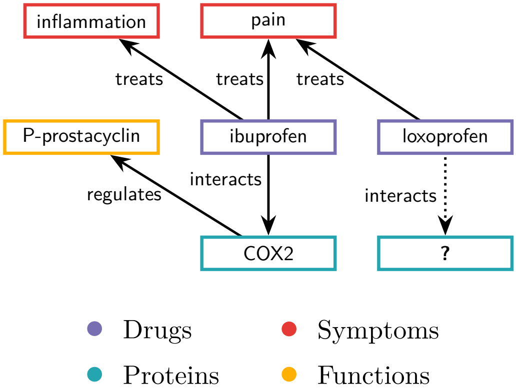
Nicola Di Mauro
University of Bari, Italy
nicola.dimauro@uniba.it

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

Knowledge Graphs



⟨loxoprofen, treats, pain⟩

⟨ibuprofen, treats, pain⟩

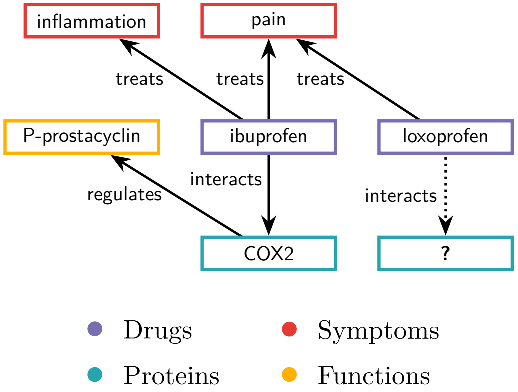
⋮

⟨COX2, regulates, P-prostacyclin⟩

⟨ibuprofen, interacts, COX2⟩

$\mathcal{Q}$: ⟨loxoprofen, interacts, **?**⟩

Knowledge Graphs



$\langle$loxoprofen, treats, pain$\rangle$

$\langle$ibuprofen, treats, pain$\rangle$

$\vdots$

$\langle$COX2, regulates, P-prostacyclin$\rangle$

$\langle$ibuprofen, interacts, COX2$\rangle$

$\mathcal{Q}$: $\langle$loxoprofen, interacts, **?**$\rangle$

KGE Models

SOTA ***knowledge graph embeddings*** (**KGE**) models

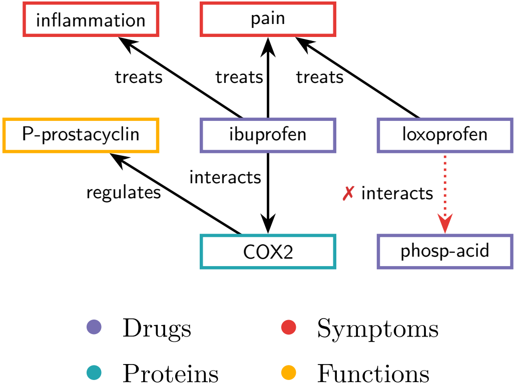
CP, RESCAL, TuckER, ComplEx

define a ***score function*** $\phi(s, r, o) \in \mathbb{R}$

E.g., for ComplEx:

$$\phi_{\text{ComplEx}}(s, r, o) = \Re(\langle \mathbf{e}_s, \mathbf{w}_r, \overline{\mathbf{e}_o} \rangle)$$

e.g., ComplEx



$\boldsymbol{K}$: only drugs and proteins interact

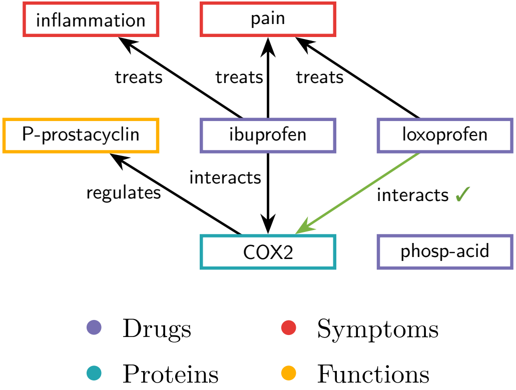
$\mathcal{A}$: $\langle$loxoprofen, interacts, **phosp-acid**$\rangle$

✘

$\mathcal{A}$: $\langle$loxoprofen, interacts, **COX2**$\rangle$

✔

e.g., ComplEx



$\boldsymbol{K}$: only drugs and proteins interact

$\boldsymbol{\mathcal{A}}$: $\langle$loxoprofen, interacts, **phosp-acid**$\rangle$

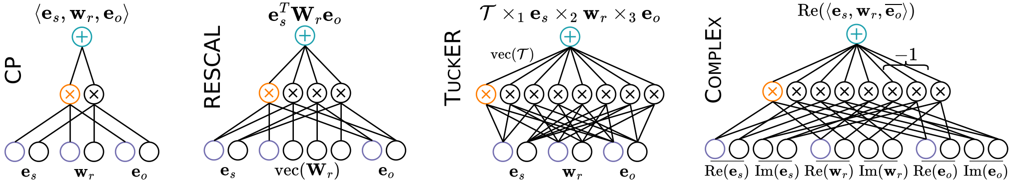
✘

$\boldsymbol{\mathcal{A}}$: $\langle$loxoprofen, interacts, **COX2**$\rangle$

✔

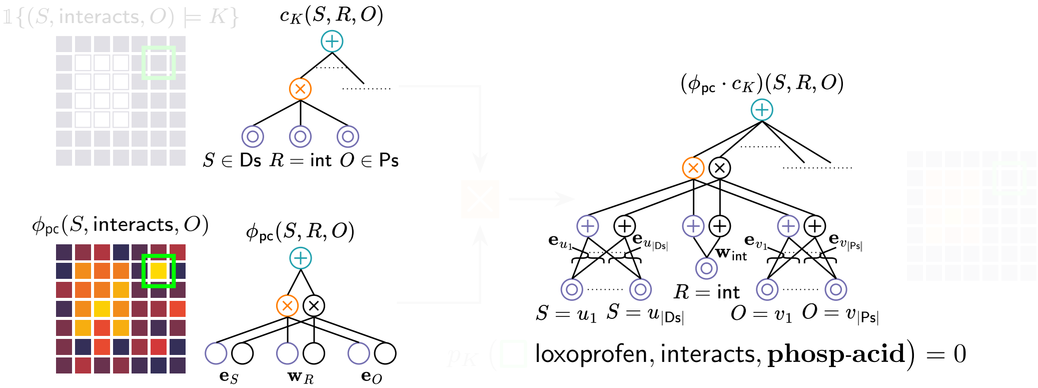
from KGE Models ...

to probabilistic circuits



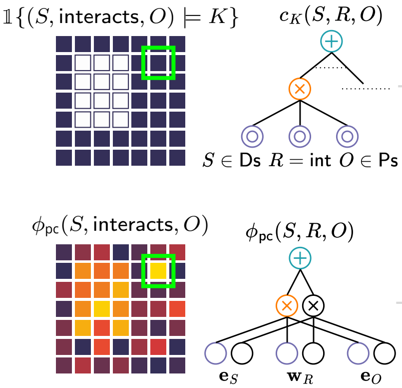
ϕ_{CP} ϕ_{RESCAL} ϕ_{TuckER} ϕ_{ComplEx}

Guaranteed satisfaction of constraints

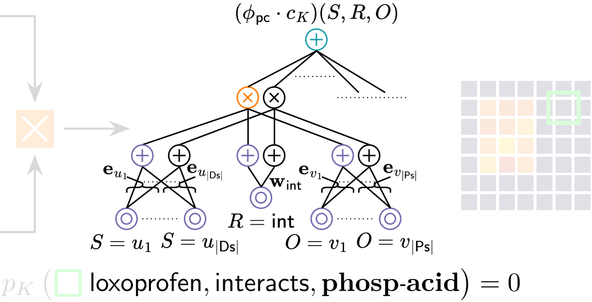


Ahmed et al., "Semantic probabilistic layers for neuro-symbolic learning", Advances in Neural Information Processing Systems 35 (NeurIPS), 2022

Guaranteed satisfaction of constraints

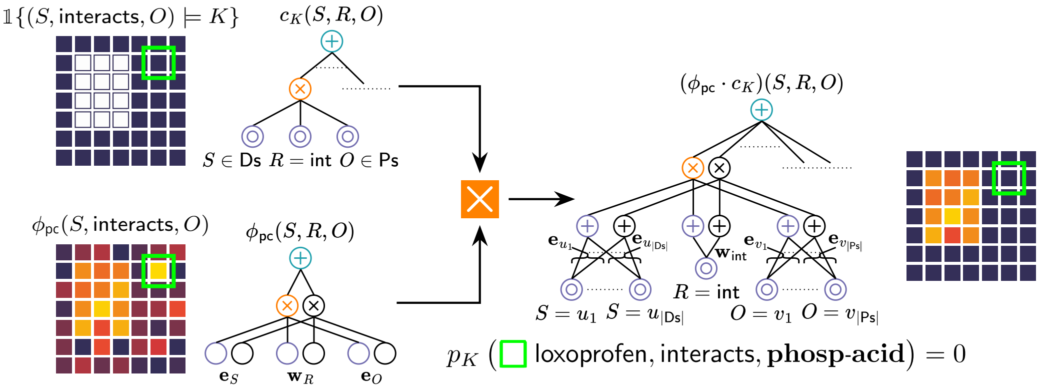


$\boldsymbol{K}$: only drugs and proteins interact



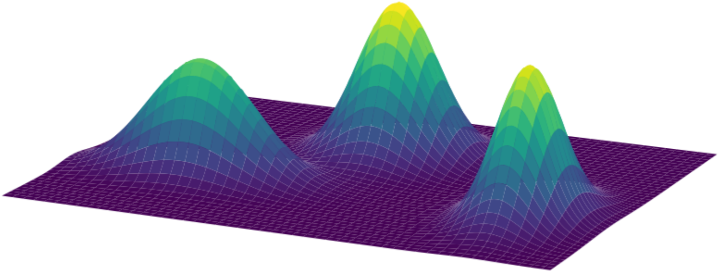
Ahmed et al., *"Semantic probabilistic layers for neuro-symbolic learning"*, *Advances in Neural Information Processing Systems 35 (NeurIPS), 2022*

Guaranteed satisfaction of constraints



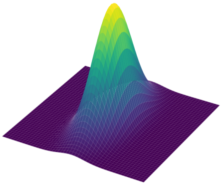
$p_K\left(\square\ \text{loxoprofen}, \text{interacts}, \textbf{phosp-acid}\right) = 0$

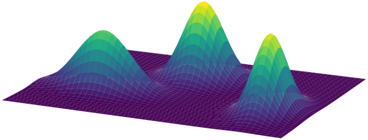
Ahmed et al., "Semantic probabilistic layers for neuro-symbolic learning", Advances in Neural Information Processing Systems 35 (NeurIPS), 2022



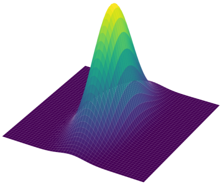
oh mixtures, you're so fine you blow my mind!

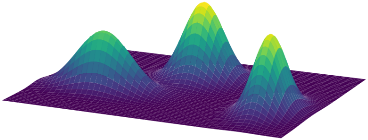
image taken from Hao Tang's course on ASR





$$p(\mathbf{X}) \longrightarrow \sum_{i=1}^{K} p_i(\mathbf{X})$$

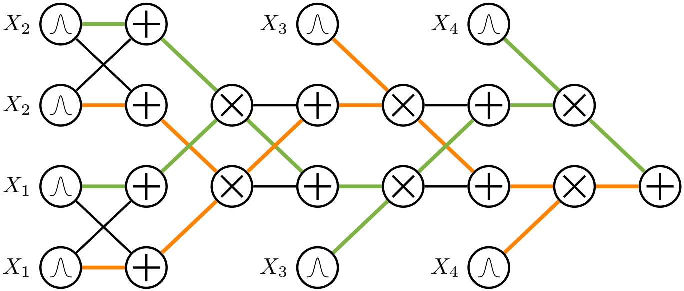




$$p(\mathbf{X}) \longrightarrow \sum_{i=1}^{K} p_i(\mathbf{X})$$

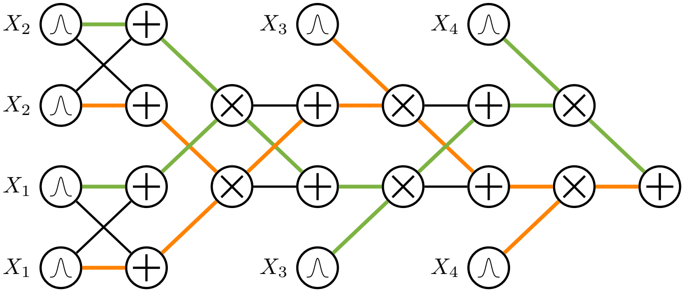
"if someone publishes a paper on model A, there will be a paper about mixtures of A soon with high probability" **A. Vergari**

Expressive efficiency

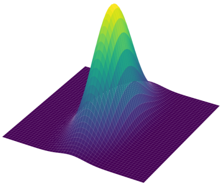


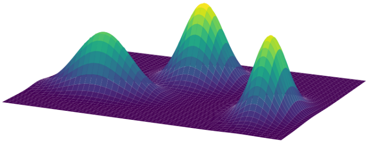
$$p(\mathbf{x}) = \sum_{\mathcal{T}} \left(\prod_{w_j \in \mathbf{w}_{\mathcal{T}}} w_j \right) \prod_{l \in \mathsf{leaves}(\mathcal{T})} p_l(\mathbf{x})$$

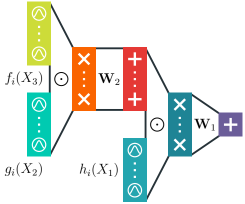
Expressive efficiency



an exponential number of mixture components!



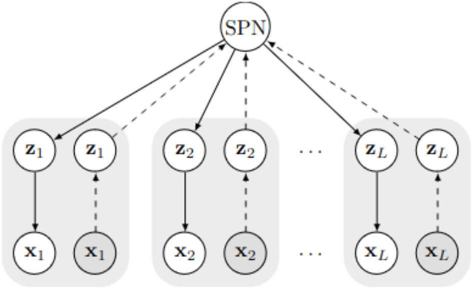


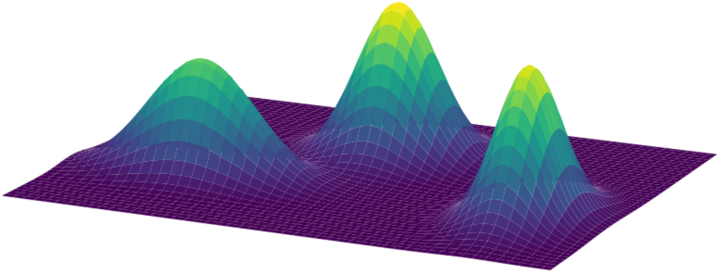


$$p(\mathbf{X}) \rightarrow \sum_{i=1}^{K} p_i(\mathbf{X}) \rightarrow \sum_{i=1}^{2^D} p_i(\mathbf{X}) = \mathsf{PC}(\mathbf{X})$$

Hierarchical Decompositional Mixtures of Variational Autoencoders

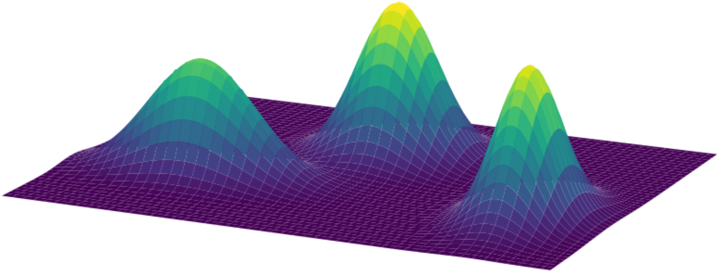
Ping Liang Tan [1 2] **Robert Peharz** [1]





$$c(\mathbf{X}) = \sum_{i=1}^{K} w_i c_i(\mathbf{X}), \quad \text{with} \quad w_i \geq 0, \quad \sum_{i=1}^{K} w_i = 1$$

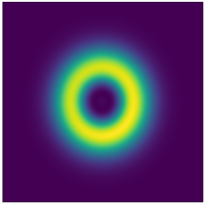
image taken from Hao Tang's course on ASR



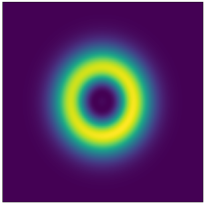
$$c(\mathbf{X}) = \sum_{i=1}^{K} w_i c_i(\mathbf{X}), \quad \text{with} \quad w_i \geq 0, \quad \sum_{i=1}^{K} w_i = 1$$

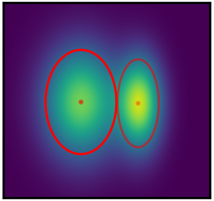
image taken from Hao Tang's course on ASR

however...



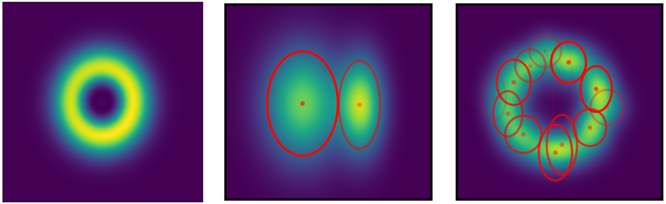
however...





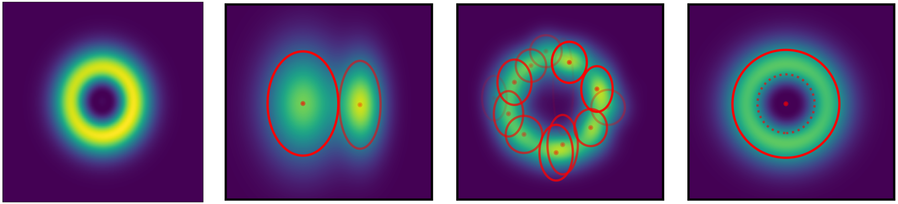
GMM ($K = 2$)

however...



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

however...



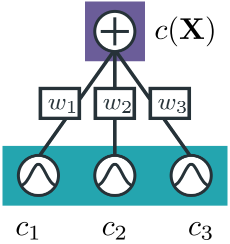
GMM ($K = 2$) GMM ($K = 16$) nGMM^2 ($K = 2$)

theorem

Shallow mixtures
with negative parameters
can be *exponentially more compact* than
deep ones with positive ones.

Loconte et al., "Subtractive Mixture Models via Squaring: Representation and Learning", ICLR, 2024

subtractive MMs as circuits

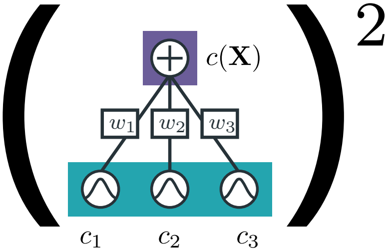


a **non-monotonic** smooth and (structured) decomposable circuit

$\Longrightarrow$ *possibly with negative outputs*

$$c(\mathbf{X}) = \sum_{i=1}^{K} w_i c_i(\mathbf{X}), \qquad w_i \in \mathbb{R},$$

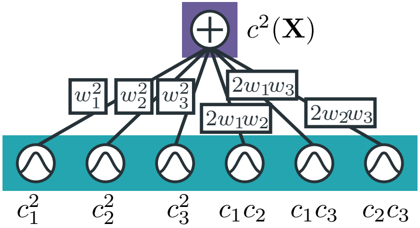
squaring shallow MMs



$$c^2(\mathbf{X}) = \left(\sum_{i=1}^{K} w_i c_i(\mathbf{X})\right)^2$$

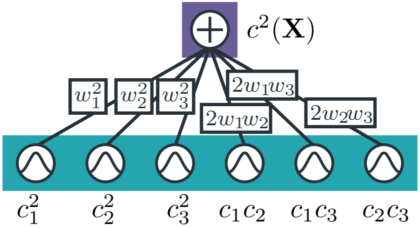
$\Longrightarrow$ *ensure non-negative output*

squaring shallow MMs



$$
\begin{aligned}
c^2(\mathbf{X}) &= \left(\sum\nolimits_{i=1}^{K} w_i c_i(\mathbf{X})\right)^2 \\
&= \sum\nolimits_{i=1}^{K} \sum\nolimits_{j=1}^{K} w_i w_j c_i(\mathbf{X}) c_j(\mathbf{X})
\end{aligned}
$$

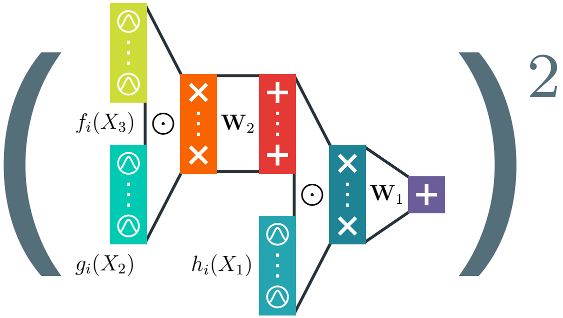
squaring shallow MMs



$$
\begin{aligned}
c^2(\mathbf{X}) &= \left(\sum\nolimits_{i=1}^{K} w_i c_i(\mathbf{X})\right)^2 \\
&= \sum\nolimits_{i=1}^{K} \sum\nolimits_{j=1}^{K} w_i w_j c_i(\mathbf{X}) c_j(\mathbf{X})
\end{aligned}
$$

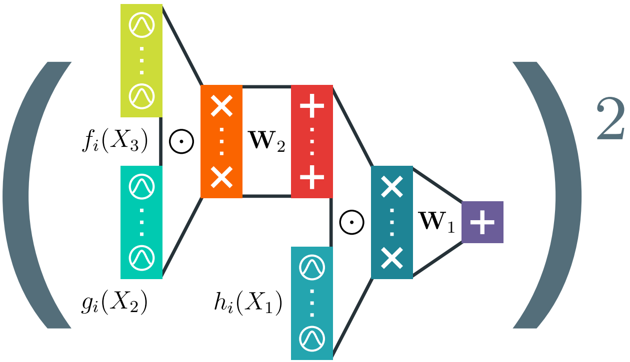
still a smooth and (str) decomposable PC with $\mathcal{O}(K^2)$ components!

$\Longrightarrow$ *but still* $\mathcal{O}(K)$ *parameters*



how to efficiently square (and renormalize) a deep PC?

Loconte et al., "Subtractive Mixture Models via Squaring: Representation and Learning", ICLR, 2024



questions?