

Integrating Logic and Prob ML with probabilistic circuits

antonio vergari (he/him)



@tetraduzione

robert peharz

18th July 2024 - ESSAI 2024 AI Summer School - Athens

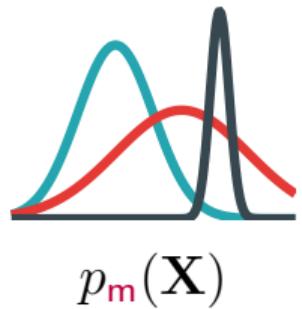
slides available at



in the previous episodes...



$q_1(m) ?$
 $q_2(m) ?$
...
 $q_k(m) ?$



\approx

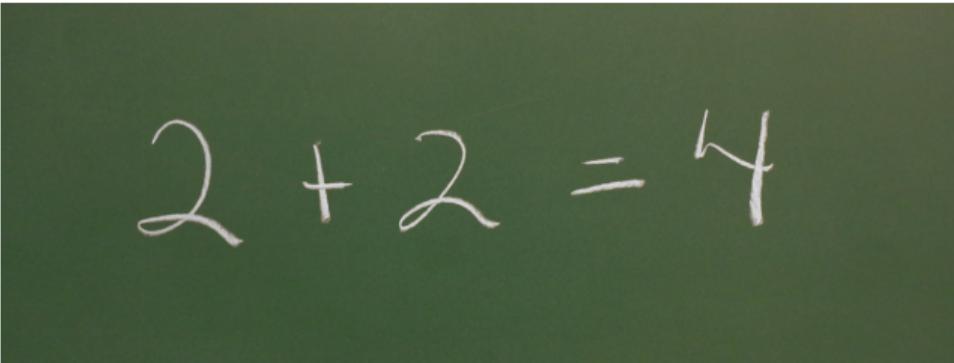
	X^1	X^2	X^3	X^4	X^5
x_8					
x_7					
x_6					
x_5					
x_4					
x_3					
x_2					
x_1					

(generative) models that can reason probabilistically

...but some events are certain!

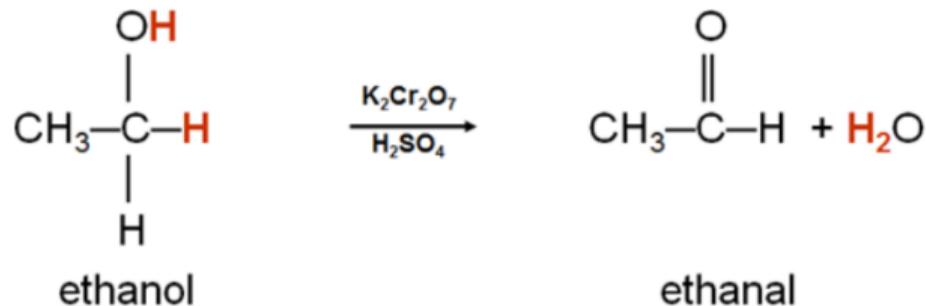
math reasoning

and logical deduction


$$2 + 2 = 4$$

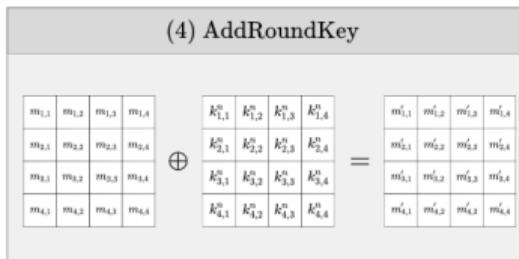
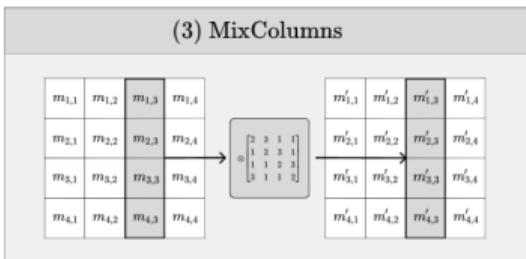
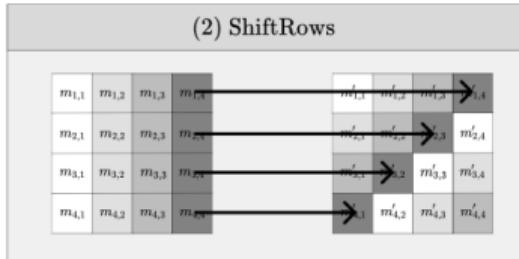
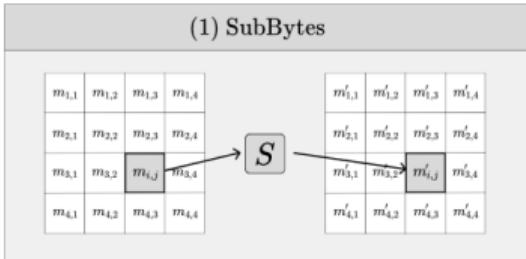
Constraints: carrying out arithmetic tasks, but also ***proving theorems***

physics laws



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

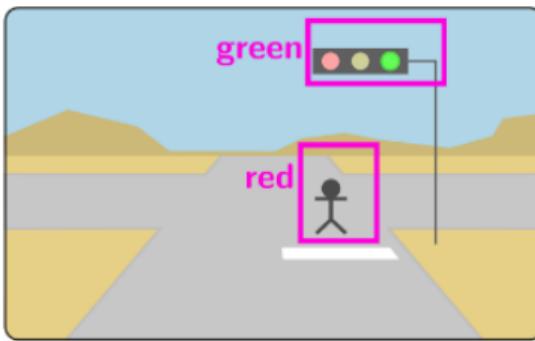
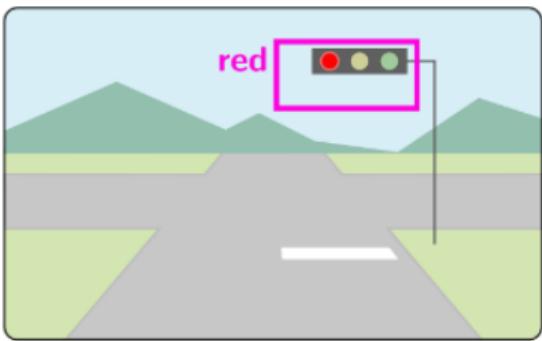
Cryptography



Constraints: logical ANDs, XORs, etc

AI safety

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



Constraints: traffic rules, scene understanding ...

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”

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

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

logical inconsistency

LLaMa 2 = LoCo-LLaMa 2

Forward Implication

$$A \rightarrow \neg B$$

A: (albatross, IsA, bird)
B: (albatross, IsA, fish)

Is an albatross a bird?



Is an albatross a fish?



Yes. Logical: ✘ Factual: ✘



No. Logical: ✓ Factual: ✓

Reverse Implication

$$\neg B \rightarrow \neg A$$

B: (albatross, IsNotA, organism)
A: (albatross, IsNotA, living thing)

Is it true that an albatross is not an organism?



Is it true that an albatross is not a living thing?



Yes. Logical: ✘ Factual: ✘



No. Logical: ✓ Factual: ✓

Negation

$$A \oplus \bar{A}$$

A: (computer, IsA, airplane)
 \bar{A} : (computer, IsNotA, airplane)

Is a computer a airplane?



No.



Is it true that a computer is not a airplane?



No. Logical: ✘ Factual: ✘



Yes. Logical: ✓ Factual: ✓

LLMs confabulate and contradict themselves¹

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..*”

what about valid molecules?

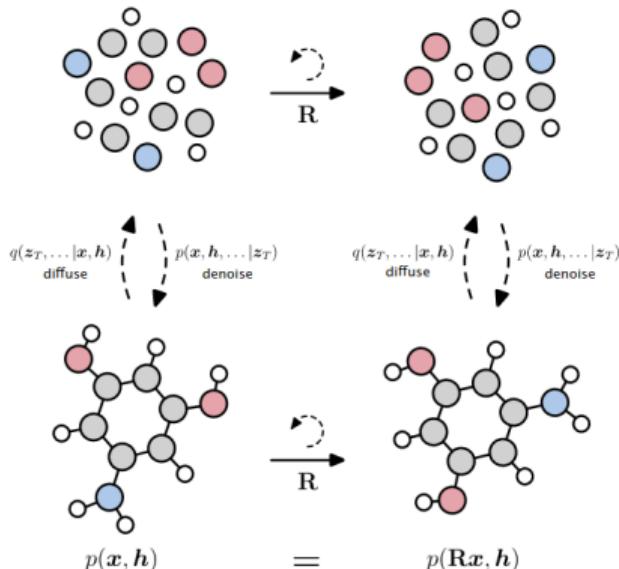
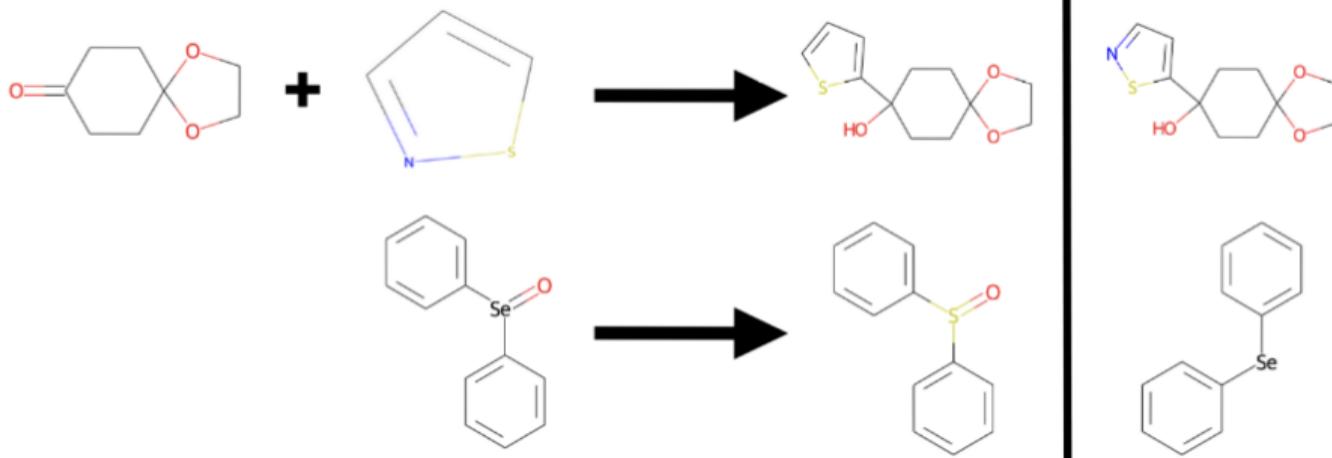


Table 2. Validity and uniqueness over 10000 molecules with standard deviation across 3 runs. Results marked (*) are not directly comparable, as they do not use 3D coordinates to derive bonds.
H: model hydrogens explicitly

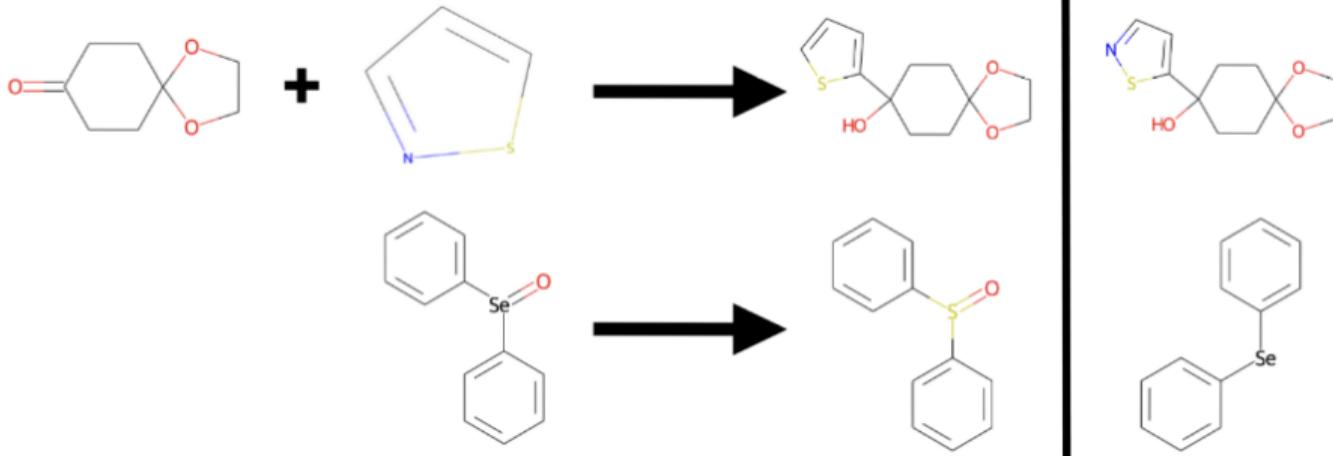
Method	H	Valid (%)	Valid and Unique (%)
Graph VAE (*)		55.7	42.3
GTVAE (*)		74.6	16.8
Set2GraphVAE (*)		59.9 ± 1.7	56.2 ± 1.4
EDM (ours)		97.5 ± 0.2	94.3 ± 0.2
E-NF	✓	40.2	39.4
G-Schnet	✓	85.5	80.3
GDM-aug	✓	90.4	89.5
EDM (ours)	✓	91.9 ± 0.5	90.7 ± 0.6
Data	✓	97.7	97.7

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
 ⇒ *a single framework for matching, paths, hierarchies, plans ...*
- II) How to integrate logic and probabilities in a single architecture
 ⇒ *combining soft and hard constraints*
- III) Logical constraints are piecewise constant functions!
 ⇒ *differentiable almost everywhere but gradient is zero!*

the issues!

- I) Logical constraints can be hard to represent in a unified way
 ⇒ *a single framework for matching, paths, hierarchies, plans ...*
- II) How to integrate logic and probabilities in a single architecture
 ⇒ *combining soft and hard constraints*
- III) Logical constraints are piecewise constant functions!
 ⇒ *differentiable almost everywhere but gradient is zero!*

the issues!

- I) Logical constraints can be hard to represent in a unified way
 ⇒ *a single framework* for matching, paths, hierarchies, plans ...
- II) How to integrate logic and probabilities in a single architecture
 ⇒ *combining soft and hard constraints*
- III) Logical constraints are piecewise constant functions!
 ⇒ *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

...and more!

	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5	Dimension 6	Dimension 7
	(D)irected (U)ndirected	(G)rounding (P)roofs	(L)ogic (P)robability (N)eural	(L)ogic (P)robability (F)uzzy	(P)arameter (S)tructure	(S)ymbols (Sub)symbols	(P)ropositional (R)elational (FOL) (LP)
øILP [Evans and Grefenstette, 2018]	D	P	L+N	L	P	S	R
DeepProbLog [Manhaeve et al., 2018]	D	P	L+P+N	P	P	S+Sub	LP
DiffLog [Si et al., 2019]	D	P	L+N	L	P+S	S	R
LRNN [Šourek et al., 2018]	D	P	L+N	F	P+S	S+Sub	LP
LTN [Donadello et al., 2017]	U	G	L+N	F	P	Sub	FOL
NeuralLP [Yang et al., 2017]	D	G	L+N	L	P	S	R
NLM [Dong et al., 2019]	D	G	L+N	L	P+S	S	R
NLProlog [Weber et al., 2019]	D	P	L+P+N	P	P+S	S+Sub	LP
NMLN [Marra and Kuželka, 2019]	U	G	L+P+N	P	P+S	S+Sub	FOL
NTP [Rocktäschel and Riedel, 2017]	D	P	L+N	L	P+S	S+Sub	R
RNM [Marra et al., 2020]	U	G	L+P+N	P	P	S+Sub	FOL
SL [Xu et al., 2018]	U	G	L+P+N	P	P	S+Sub	P
SBR [Diligenti et al., 2017]	U	G	L+N	F	P	Sub	FOL
Tensorlog [Cohen et al., 2017]	D	P	L+N	P	P	S+Sub	R

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)

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

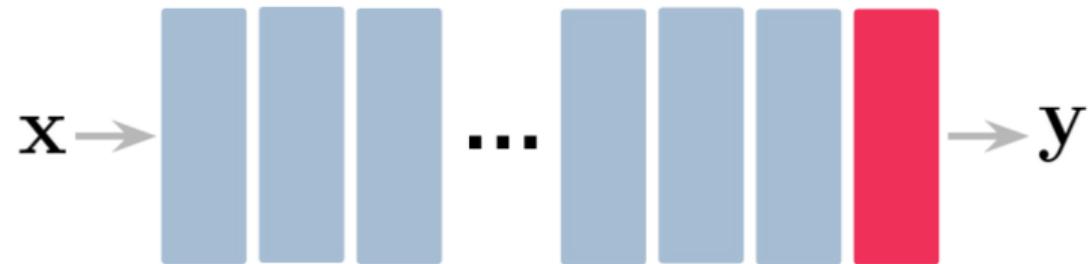
first-order logic (FOL)

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

satisfiability modulo theory (SMT)

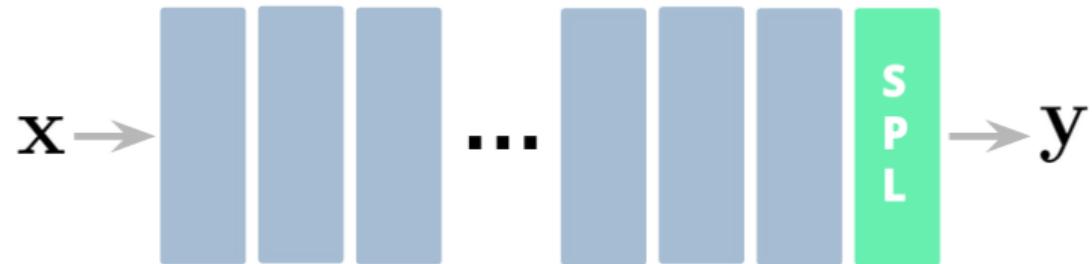
$$(\alpha X_i - \beta X_j \leq 100) \vee (X_j + X_k \geq 0) \implies (X_j X_k \leq X_i)$$

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?



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

Ground Truth

***n*esy structured output prediction (SOP) tasks**

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

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

nesy structured output prediction (SOP) tasks

When?



Ground Truth

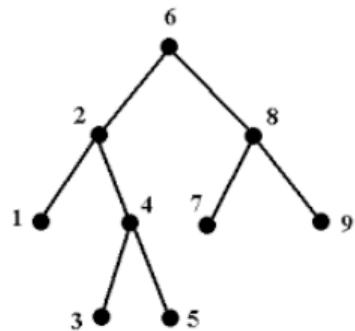
given \mathbf{x} // e.g. a tile map

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. $\mathbf{y} \models K$ // 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

When?



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

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

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

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

hierarchical multi-label classification

When?



given \mathbf{x} // e.g. a user preference over $K - N$ sushi types
find $\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} p_{\theta}(\mathbf{y} \mid \mathbf{x})$ // e.g. prefs over N more types
s.t. $\mathbf{y} \models K$ // e.g., output valid rankings

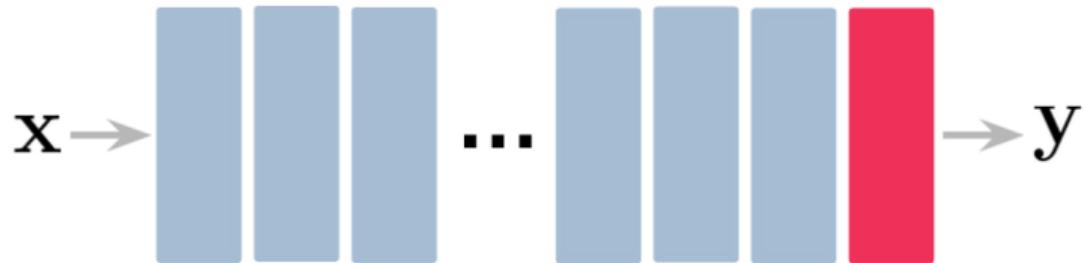
user preference learning

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

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

losses improve consistency during training...

Constraint losses

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

losses improve consistency during training...

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

WMC

computing the probability of logical formulas

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

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

WMC

computing the probability of logical formulas

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

computing the probability of K

computing the probability of logical formulas

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

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

²van 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

on the independence assumption

$$K : \neg r \vee \neg g$$

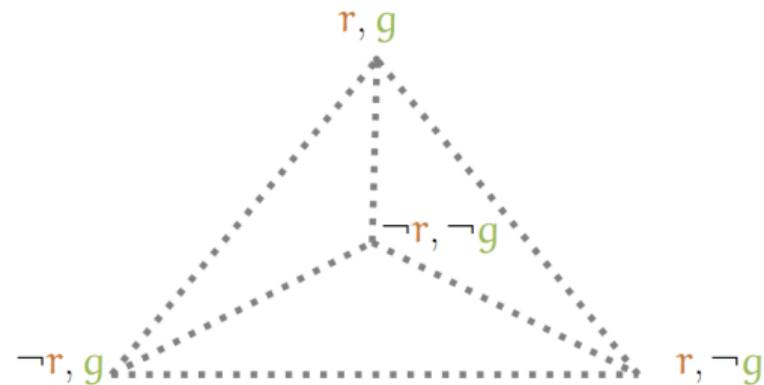
*a neural net should not output that a traffic light is both **red** and **green***



on the independence assumption

$$K : \neg r \vee \neg g$$

a neural net should not output that a traffic light is both **red** and **green**

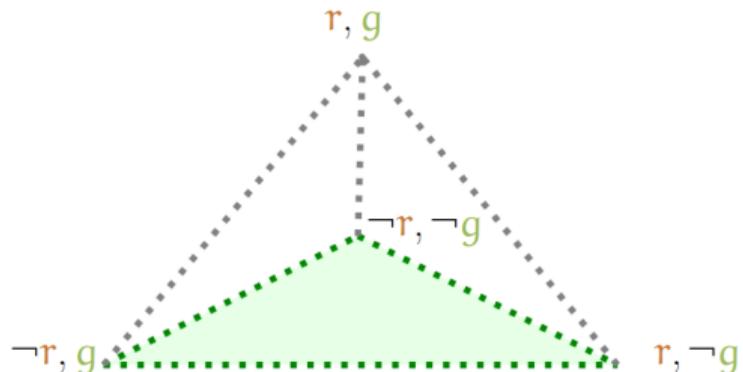


on the independence assumption

$$K : \neg r \vee \neg g$$

a neural net should not output that a traffic light is both **red** and **green**

only some probability assignments should be non-zero (lower triangle)

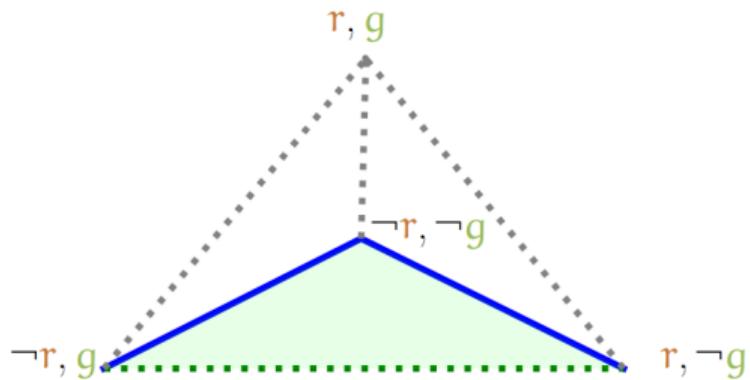


on the independence assumption

$$K : \neg r \vee \neg g$$

a neural net should not output that a traffic light is both **red** and **green**

but assuming $p(r, g) = p(r)p(g)$ restricts this even further (only blue lines)

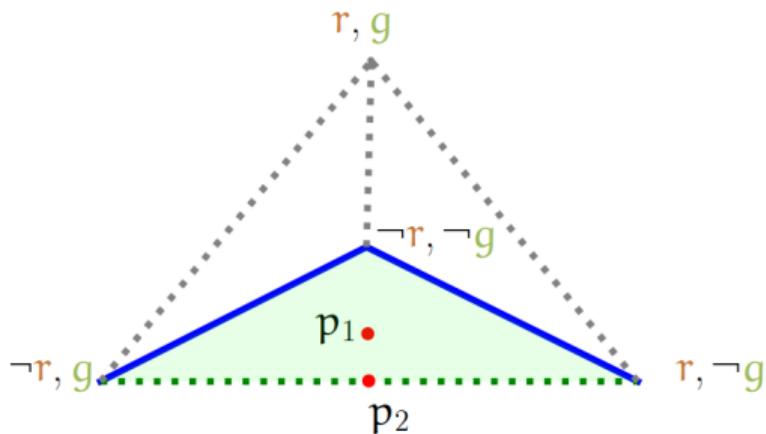


on the independence assumption

$$K : \neg r \vee \neg g$$

a neural net should not output that a traffic light is both **red** and **green**

but assuming $p(r, g) = p(r)p(g)$ restricts this even further (only blue lines)



Constraint losses



Ground Truth



ResNet-18



Semantic Loss

...but cannot guarantee consistency at test time!

SPL



Ground Truth



ResNet-18



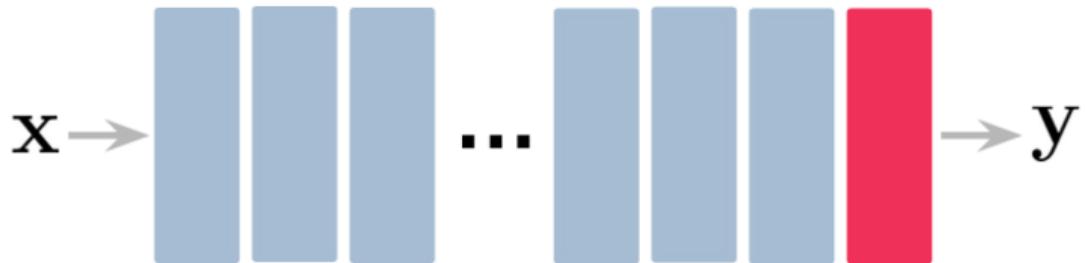
Semantic Loss



SPL (ours)

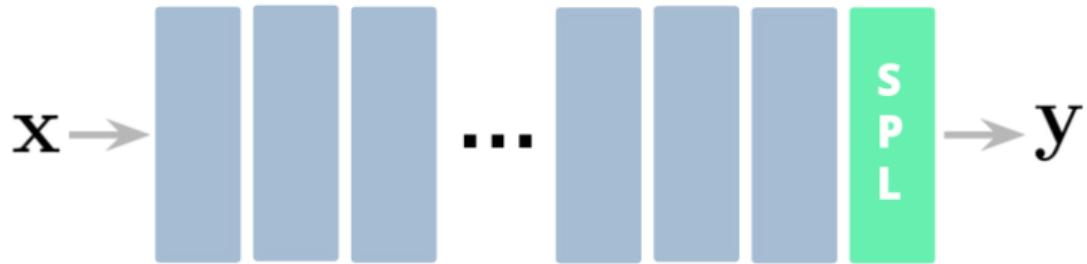
you can predict valid paths 100% of the time!

How?

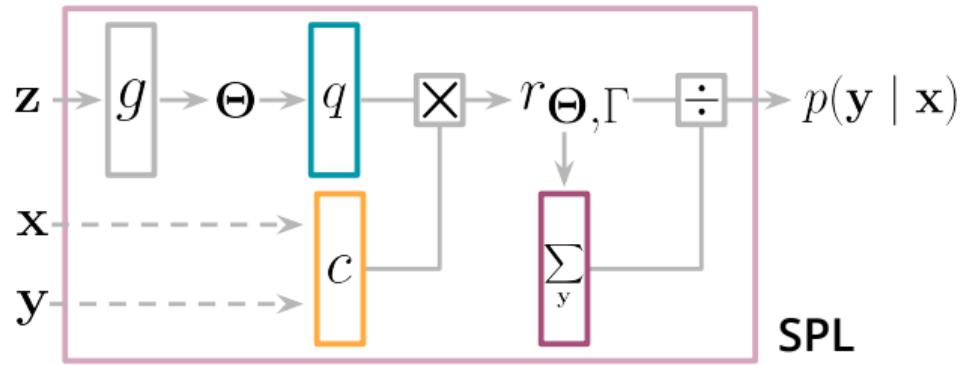


take an unreliable neural network architecture...

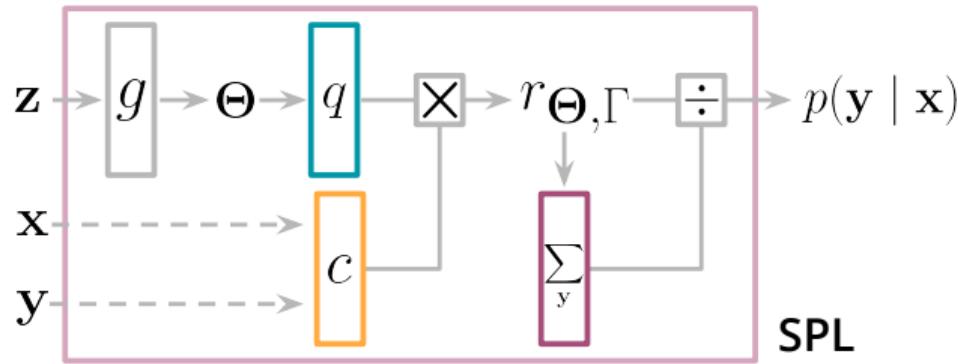
How?



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

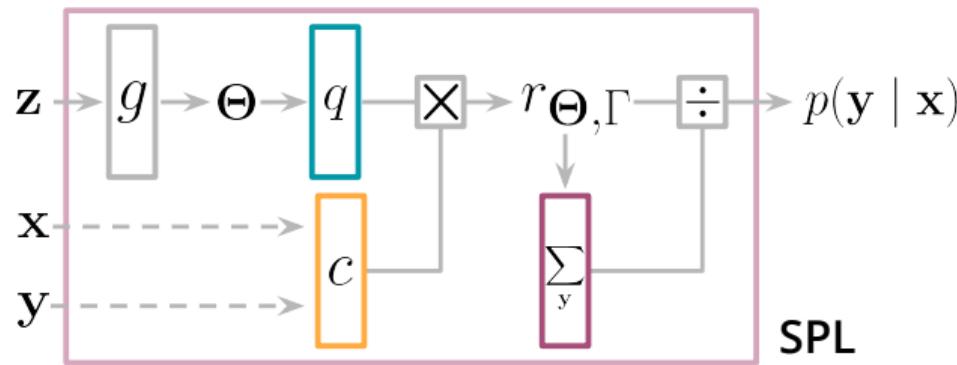


SPL



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

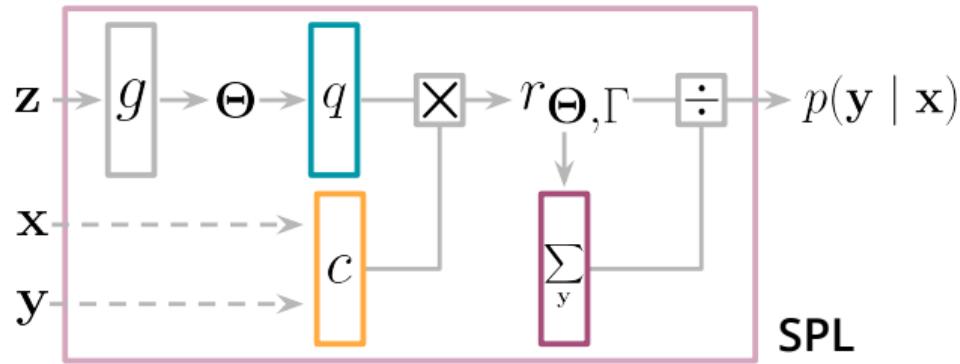
$\mathbf{q}_{\Theta}(\mathbf{y} \mid g(\mathbf{z}))$ is an expressive distribution over labels



$$p(\mathbf{y} \mid \mathbf{x}) = \mathbf{q}_{\Theta}(\mathbf{y} \mid g(\mathbf{z})) \cdot \mathbf{c}_K(\mathbf{x}, \mathbf{y})$$

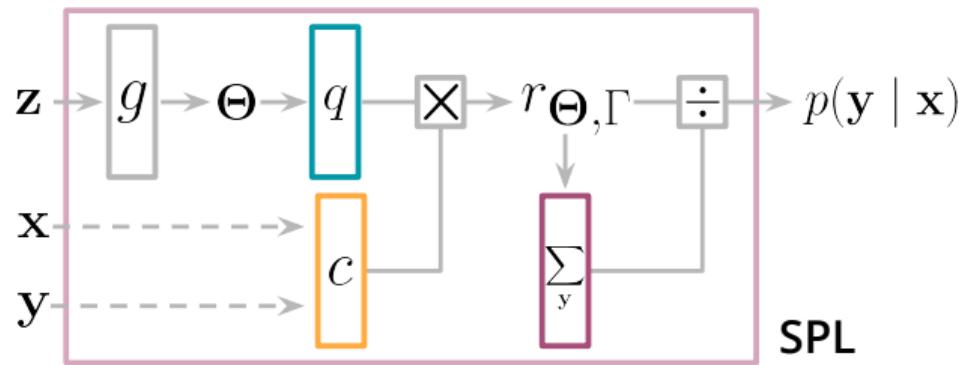
$\mathbf{c}_K(\mathbf{x}, \mathbf{y})$ encodes the constraint $\mathbb{1}\{\mathbf{x}, \mathbf{y} \models K\}$

SPL



$$p(\mathbf{y} \mid \mathbf{x}) = \mathbf{q}_{\Theta}(\mathbf{y} \mid g(\mathbf{z})) \cdot \mathbf{c}_{\mathbf{x}}(\mathbf{x}, \mathbf{y})$$

a product of experts : (



$$p(\mathbf{y} \mid \mathbf{x}) = \mathbf{q}_\Theta(\mathbf{y} \mid g(\mathbf{z})) \cdot \mathbf{c}_K(\mathbf{x}, \mathbf{y}) / \mathbf{Z}(\mathbf{x})$$

$$\mathbf{Z}(\mathbf{x}) = \sum_{\mathbf{y}} q_\Theta(\mathbf{y} \mid \mathbf{x}) \cdot c_K(\mathbf{x}, \mathbf{y})$$

Goal

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

Goal

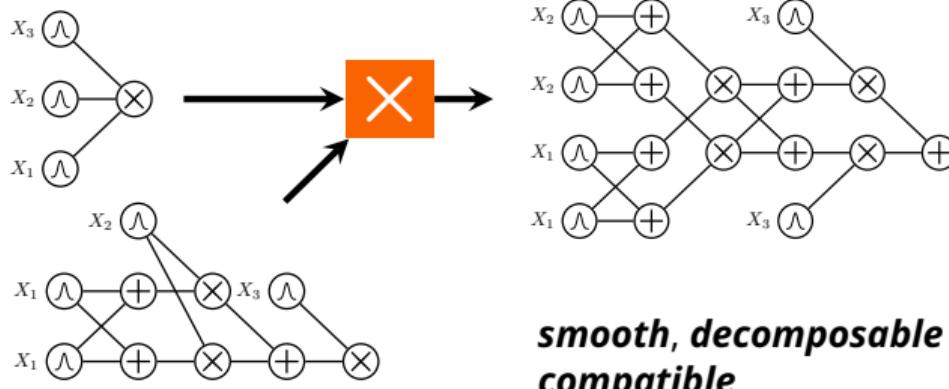
*Can we design q and c
to be **deep computational graphs**
yet yielding a tractable product?*

Goal

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

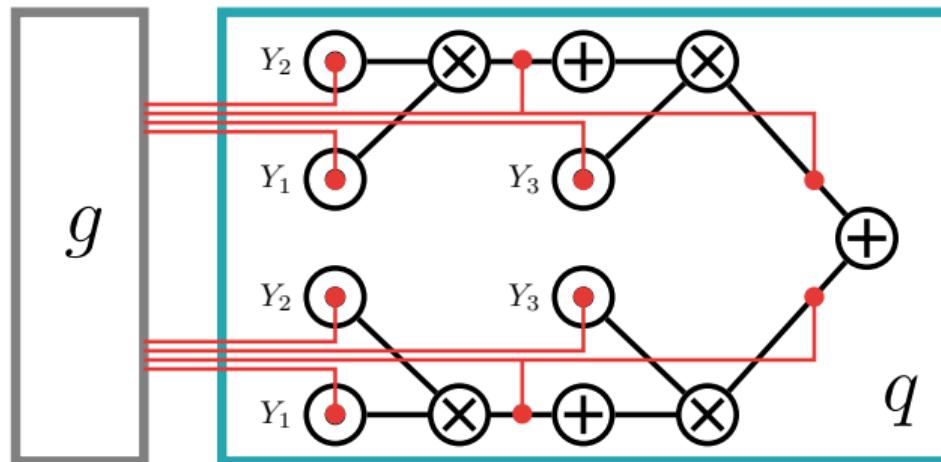
yes! as *circuits!*

Tractable products

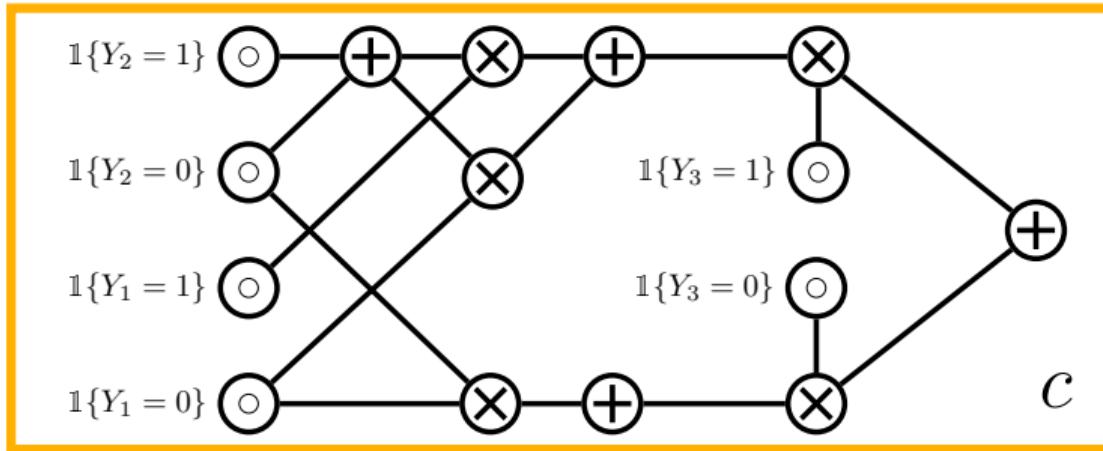


***smooth, decomposable
compatible***

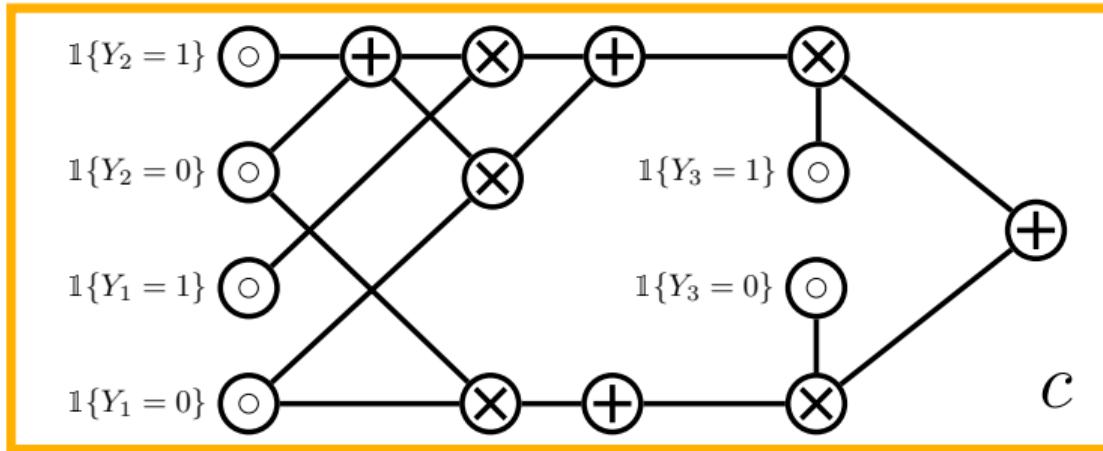
exactly compute \mathcal{Z} in time $O(|\mathbf{q}||\mathbf{c}|)$



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



and a logical circuit $c(y, x)$ encoding K



compiling logical formulas into circuits

Knowledge compilation

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

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

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

$\mathbb{1}\{Y_1 = 1\}$

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

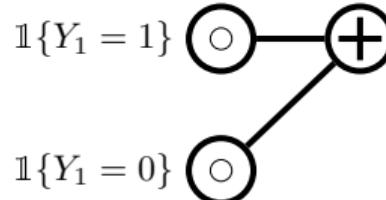
$\mathbb{1}\{Y_2 = 1\}$

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

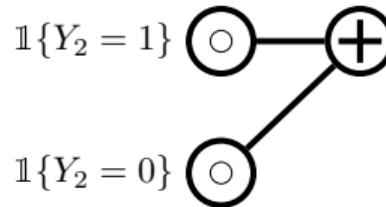
$\mathbb{1}\{Y_3 = 1\}$

Knowledge compilation

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

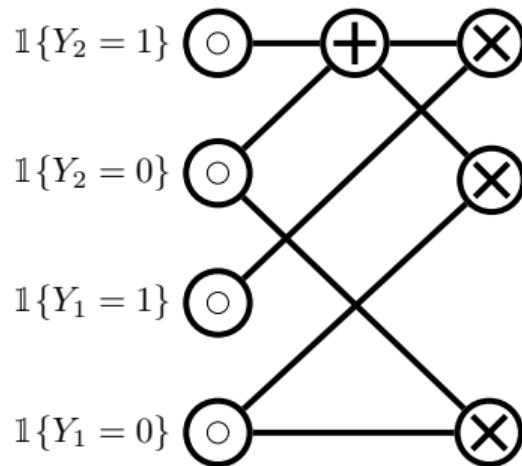


$\wedge (Y_2 = 1 \implies Y_3 = 1)$



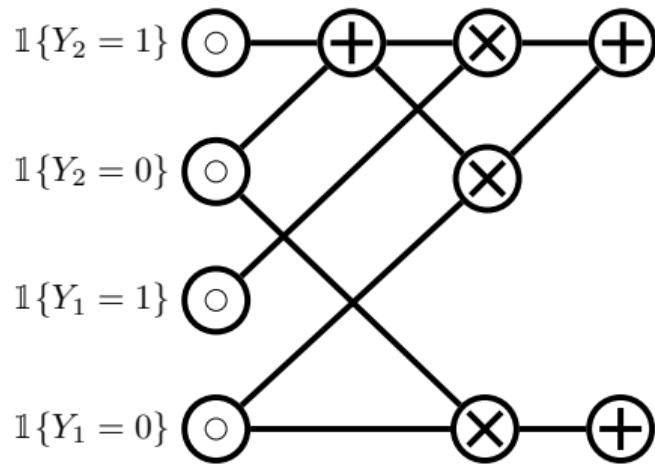
Knowledge compilation

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



Knowledge compilation

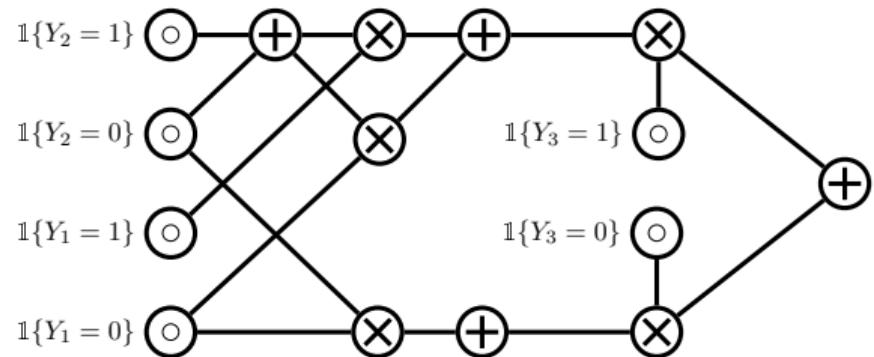
$\mathsf{K} : (Y_1 = 1 \implies Y_3 = 1)$
 $\wedge (Y_2 = 1 \implies Y_3 = 1)$



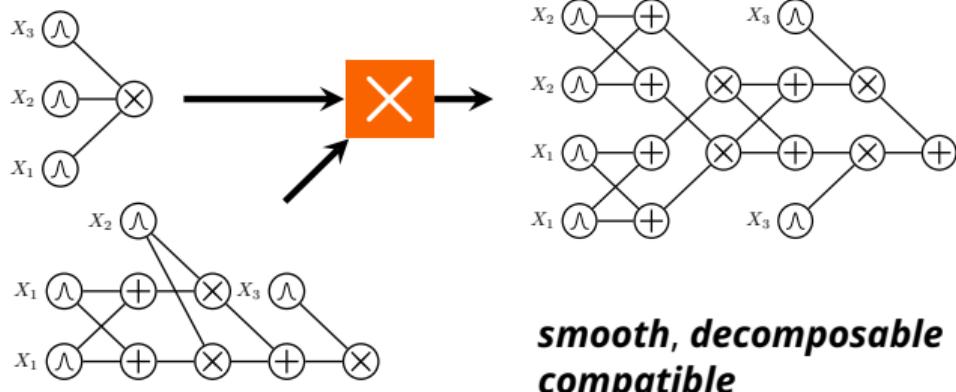
Knowledge compilation

$K : (Y_1 = 1 \implies Y_3 = 1)$

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



Tractable products



***smooth, decomposable
compatible***

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

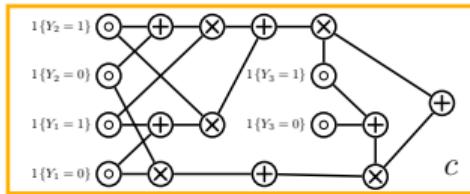
SPL recipe

$$\begin{aligned} K : & (Y_1 = 1 \implies Y_3 = 1) \\ \wedge & (Y_2 = 1 \implies Y_3 = 1) \end{aligned}$$

1) Take a
logical constraint

SPL recipe

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

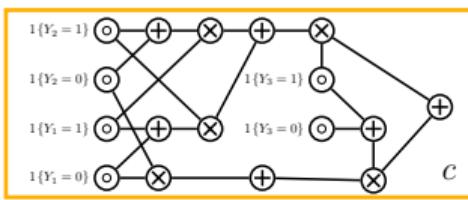


1) Take a
logical constraint

2) Compile it into
a constraint circuit

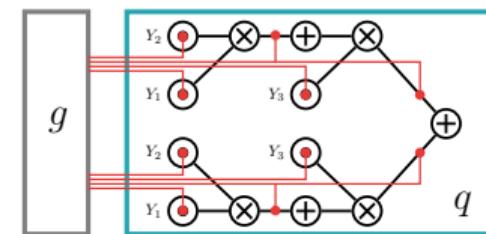
SPL recipe

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



1) Take a
logical constraint

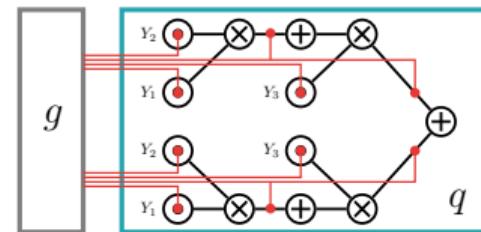
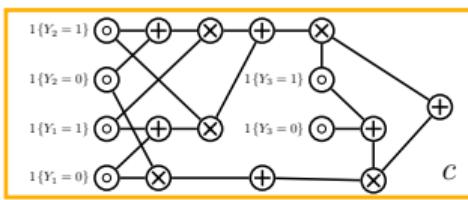
2) Compile it into
a constraint circuit



3) Multiply it
by a circuit distribution

SPL recipe

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



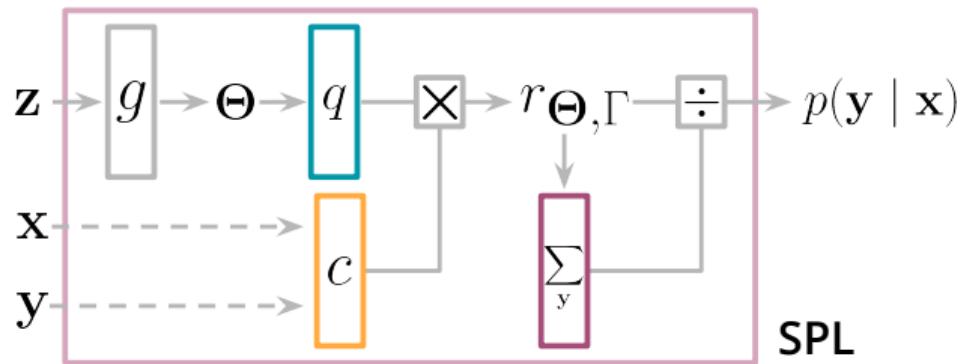
1) Take a
logical constraint

2) Compile it into
a constraint circuit

3) Multiply it
by a circuit distribution

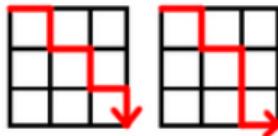
4) *train end-to-end by sgd!*

Experiments



how good are SPLs?

Experiments



Architecture	Simple Path			Preference Learning		
	Exact	Hamming	Consistent	Exact	Hamming	Consistent
MLP+FIL	5.6	85.9	7.0	1.0	75.8	2.7
MLP+ \mathcal{L}_{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}_{SL}	59.4	97.7	61.2
ResNet-18+SPL	78.2	96.3	100.0

Experiments



cost: 39.31



cost: ∞



cost: ∞



cost: 45.09



cost: 57.31



cost: ∞



cost: ∞



cost: 58.09

SPLs

(and more circuits)

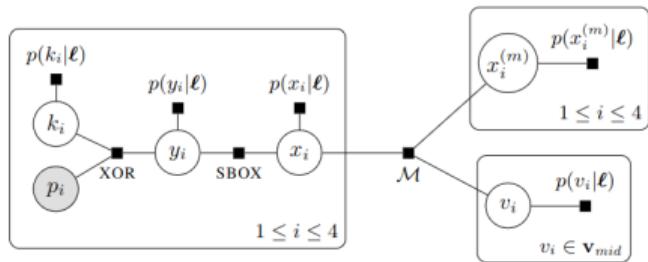
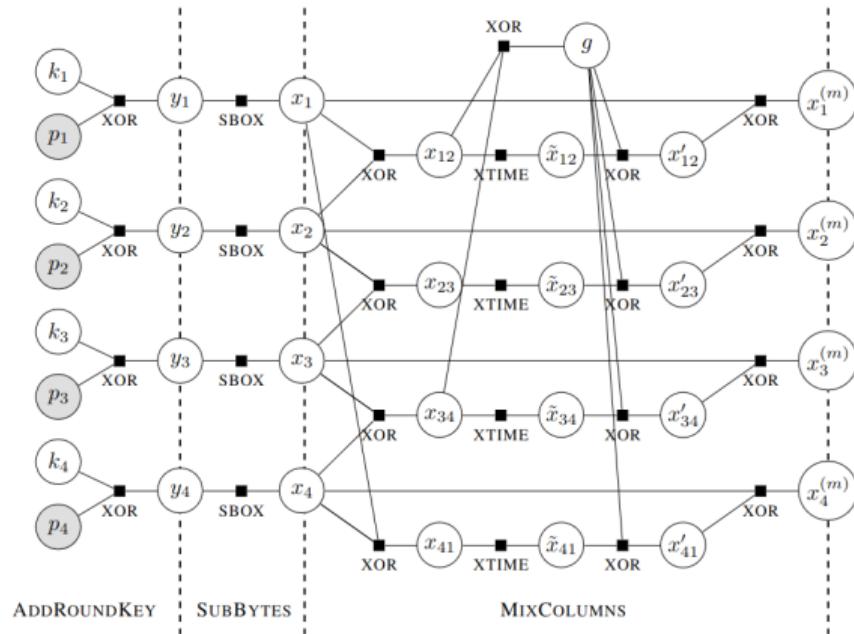
everywhere

Exact Soft Analytical Side-Channel Attacks using Tractable Circuits

Thomas Wedenig¹ Rishub Nagpal² Gaëtan Cassiers³ Stefan Mangard² Robert Peharz¹

cryptographic attacks via neuro-symbolic integration (ICML 2024, next week!)

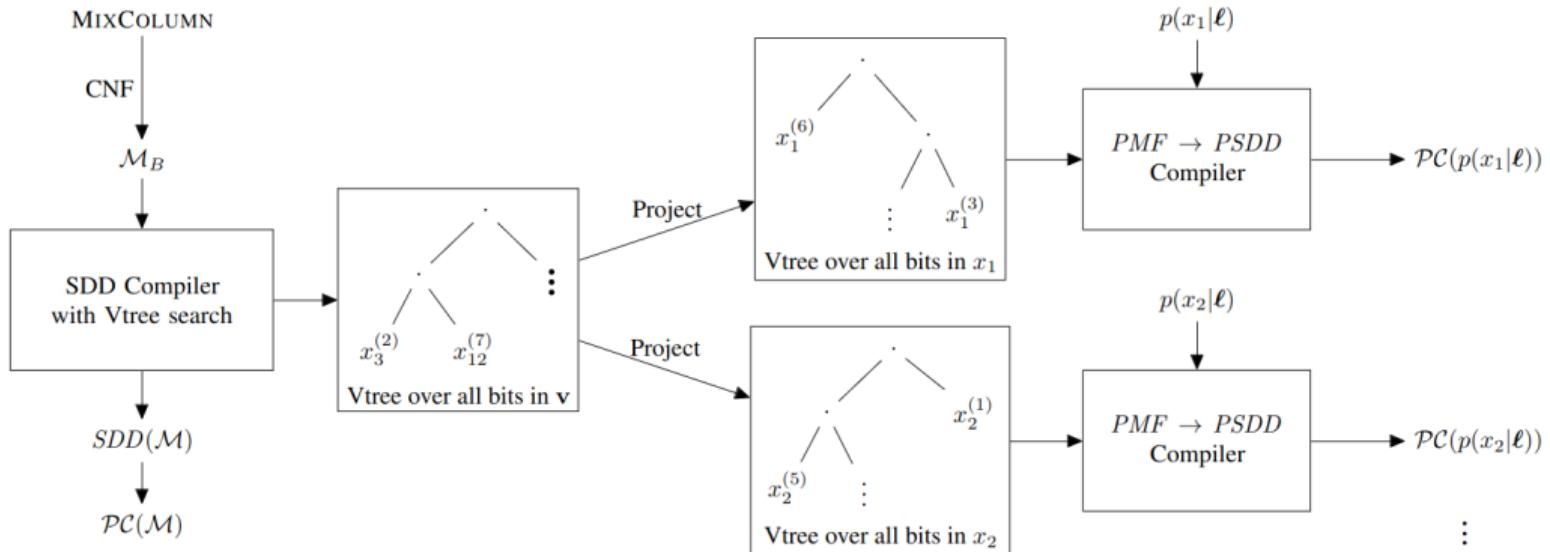
ExSASCA



$$\text{XOR}(a, b, c) = \begin{cases} 1 & \text{if } a \oplus b = c \\ 0 & \text{else} \end{cases} \quad \text{SBOX}(a, b) = \begin{cases} 1 & \text{if } S(a) = b \\ 0 & \text{else} \end{cases}$$

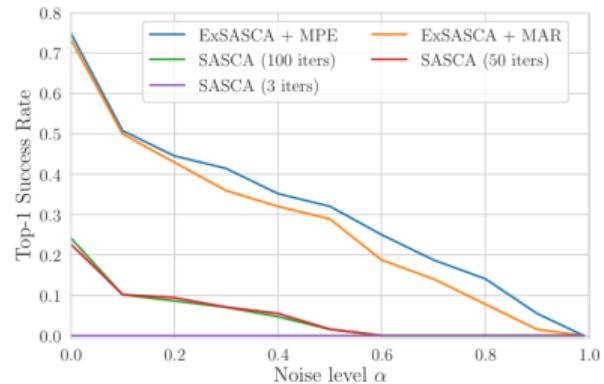
$$\text{XTIME}(a, b) = \begin{cases} 1 & \text{if } f_{\text{XTIME}}(a) = b \\ 0 & \text{else} \end{cases}$$

ExSASCA



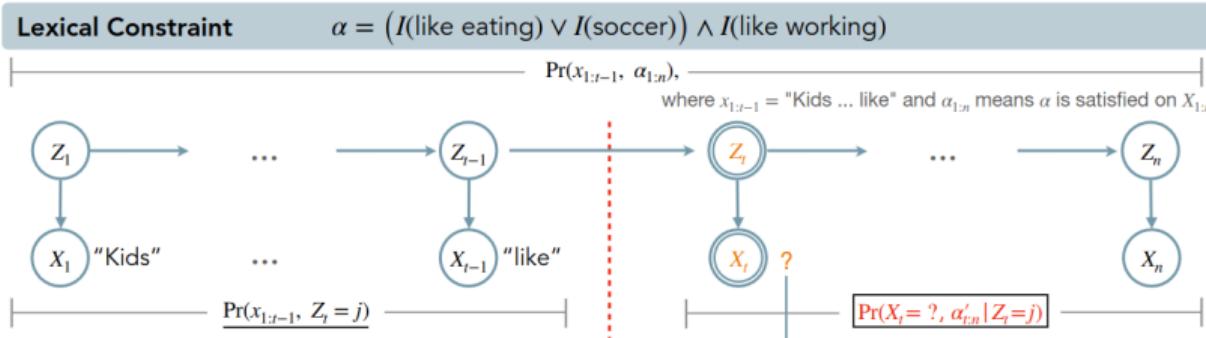
ExSASCA

Inference Method	Top-1 Success Rate		
	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-8}$	$\varepsilon = 0$
Baseline	0.10%	0.10%	0.10%
SASCA (3 iters)	0.52%	0.52%	0.52%
SASCA (50 iters)	32.34%	33.81%	33.76%
SASCA (100 iters)	34.25%	36.04%	35.84%
ExSASCA + MAR	57.36%	67.37%	67.37%
ExSASCA + MPE	57.51%	67.60%	67.61%



Tractable Control for Autoregressive Language Generation

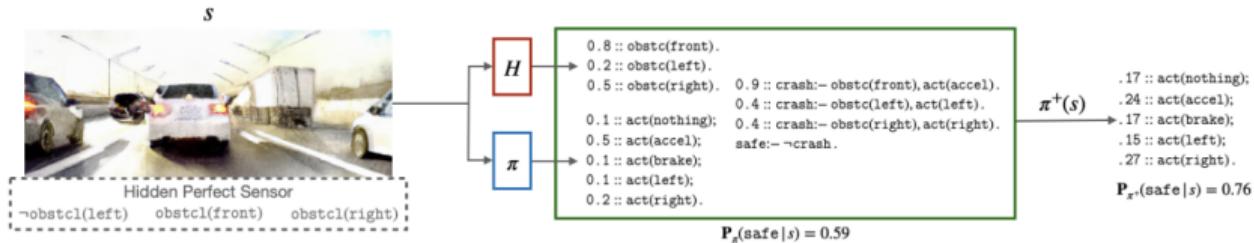
Honghua Zhang^{* 1} Meihua Dang^{* 1} Nanyun Peng¹ Guy Van den Broeck¹



constrained text generation with LLMs (ICML 2023)

Safe Reinforcement Learning via Probabilistic Logic Shields

Wen-Chi Yang¹, Giuseppe Marra¹, Gavin Rens and Luc De Raedt^{1,2}



reliable reinforcement learning (AAAI 23)

Logically Consistent Language Models via Neuro-Symbolic Integration

The figure displays three panels illustrating logical reasoning examples for different logical operations:

- Forward Implication**: $A \rightarrow \neg B$.
A: (albatross, isA, bird)
B: (albatross, isA, fish)
Is an albatross a bird? (Yes)
Is an albatross a fish? (Yes. Logical: ✕ Factual: ✕)
Is an albatross a fish? (No. Logical: ✓ Factual: ✓)
- Reverse Implication**: $\neg B \rightarrow \neg A$.
B: (albatross, isNotA, organism)
C: (albatross, isNotA, living thing)
Is it true that an albatross is not an organism? (Yes)
Is it true that an albatross is not a living thing? (Yes. Logical: ✕ Factual: ✕)
Is it true that an albatross is not a living thing? (No. Logical: ✓ Factual: ✓)
- Negation**: $A \bullet \neg A$.
A: (computer, isA, airplane)
B: (computer, isNotA, airplane)
Is a computer a airplane? (No)
Is it true that a computer is not a airplane? (No. Logical: ✕ Factual: ✕)
Is it true that a computer is not a airplane? (Yes. Logical: ✓ Factual: ✓)

Legend:
LLM = LLaMA 2
LoCo-LLM = LoCo-LLM 2

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

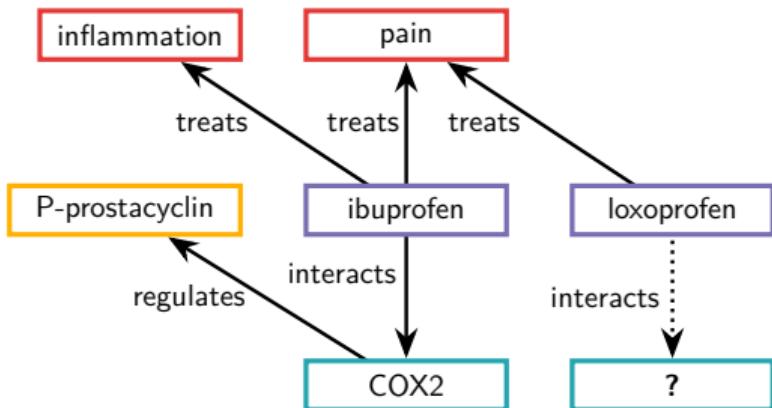
Nicola Di Mauro
University of Bari, Italy
nicola.dimauro@uniba.it

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

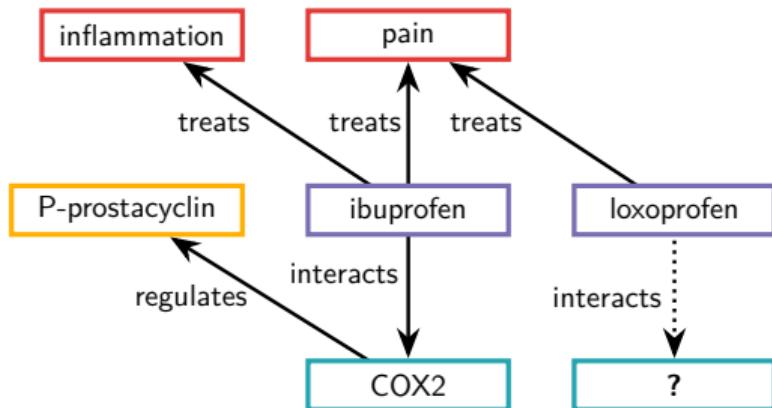


- Drugs
- Proteins
- Symptoms
- Functions

`<loxoprofen, treats, pain>`
`<ibuprofen, treats, pain>`
:
`<COX2, regulates, P-prostacyclin>`
`<ibuprofen, interacts, COX2>`

Q: `<loxoprofen, interacts, ?>`

Knowledge Graphs



- Drugs
- Proteins
- Symptoms
- Functions

`<loxoprofen, treats, pain>`
`<ibuprofen, treats, pain>`
:
`<COX2, regulates, P-prostacyclin>`
`<ibuprofen, interacts, COX2>`

\mathcal{Q} : `<loxoprofen, interacts, ?>`

KGE Models

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

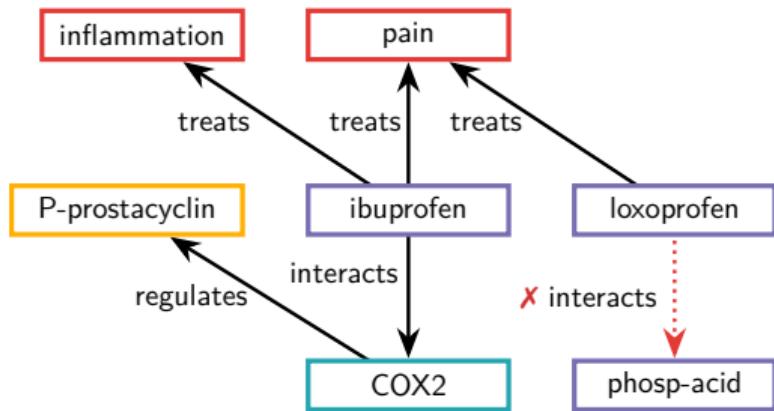
CP, RESCAL, TuckER, ComplEx

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



- Drugs
- Proteins
- Symptoms
- Functions

K: only drugs and proteins interact

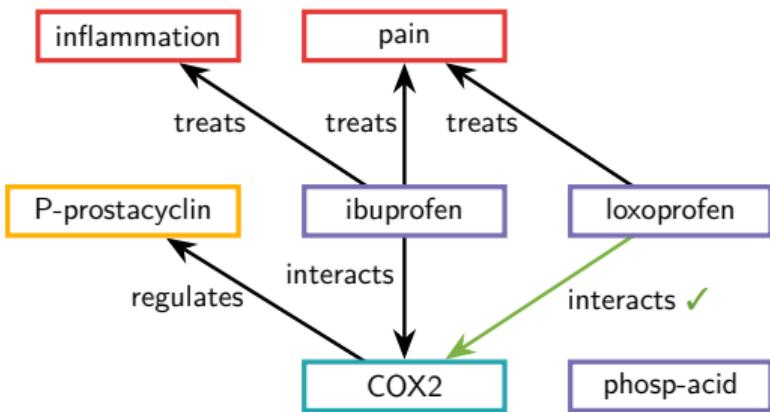
A: <loxoprofen, interacts, **phosp-acid**>

✗

A: <loxoprofen, interacts, **COX2**>

✓

e.g., ComplEx



- Drugs
- Proteins
- Symptoms
- Functions

K: only drugs and proteins interact

A: <loxoprofen, interacts, **phosp-acid**>

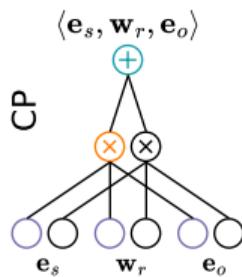
✗

A: <loxoprofen, interacts, **COX2**>

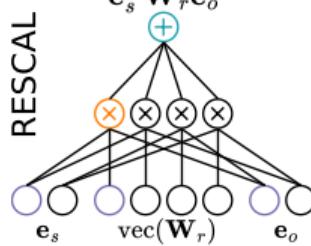
✓

from KGE Models ...

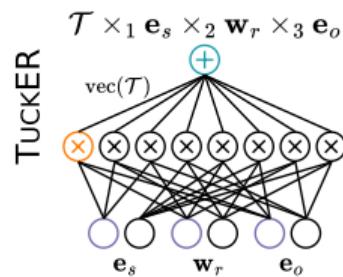
to probabilistic circuits



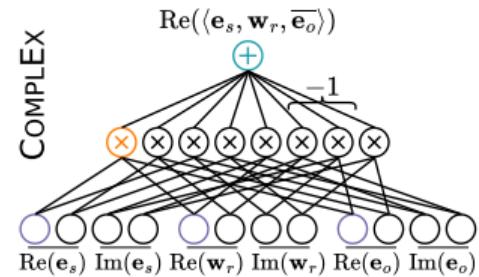
$$\phi_{\text{CP}}$$



$$\phi_{\text{RESCAL}}$$



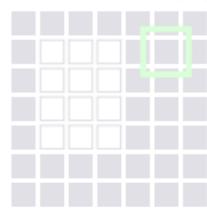
$$\phi_{\text{TuckER}}$$



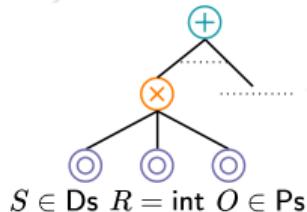
$$\phi_{\text{ComplEx}}$$

Guaranteed satisfaction of constraints

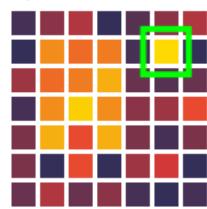
$$\models \{(S, \text{interacts}, O) \models K\}$$



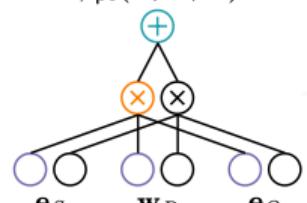
$$c_K(S, R, O)$$



$$\phi_{pc}(S, \text{interacts}, O)$$



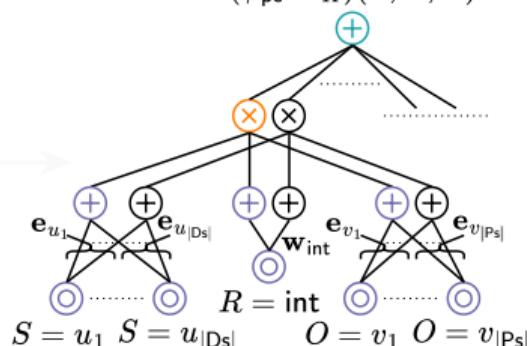
$$\phi_{pc}(S, R, O)$$



p_K

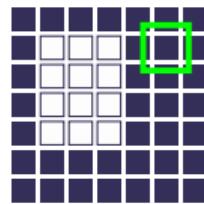
$$\text{loxoprofen, interacts, phosph-acid}) = 0$$

$$(\phi_{pc} \cdot c_K)(S, R, O)$$

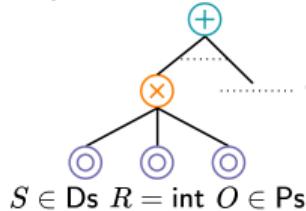


Guaranteed satisfaction of constraints

$$\mathbb{1}\{(S, \text{interacts}, O) \models K\}$$



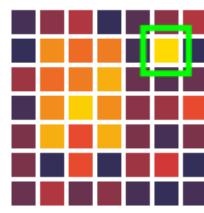
$$c_K(S, R, O)$$



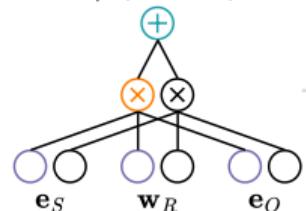
K : only drugs and proteins interact

$$(\phi_{pc} \cdot c_K)(S, R, O)$$

$$\phi_{pc}(S, \text{interacts}, O)$$



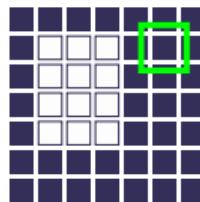
$$\phi_{pc}(S, R, O)$$



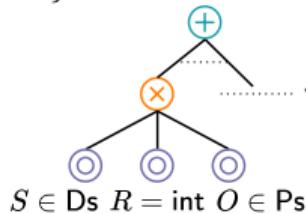
$$p_K (\square \text{ loxoprofen, interacts, phosph-acid}) = 0$$

Guaranteed satisfaction of constraints

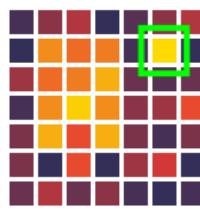
$$\mathbb{1}\{(S, \text{interacts}, O) \models K\}$$



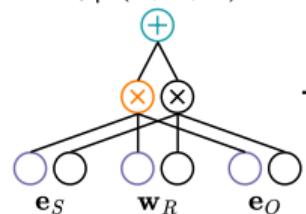
$$c_K(S, R, O)$$



$$\phi_{pc}(S, \text{interacts}, O)$$



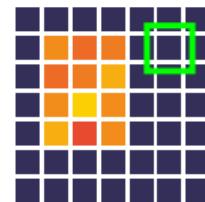
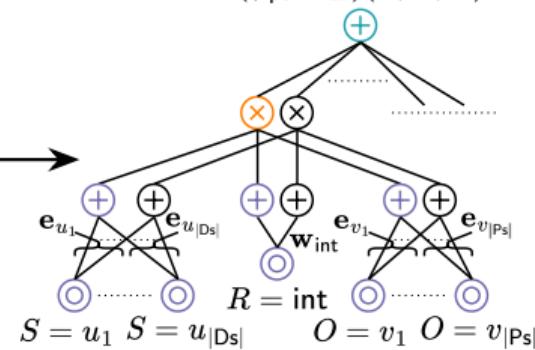
$$\phi_{pc}(S, R, O)$$



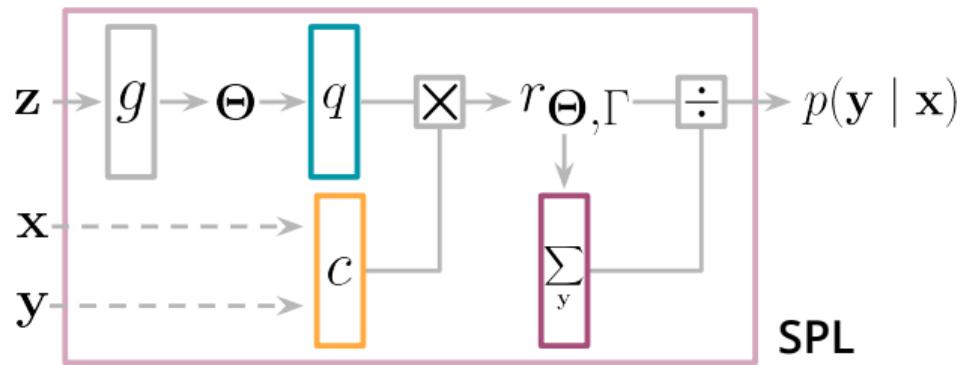
$$p_K$$

$$(\square \text{ loxoprofen, interacts, phosph-acid}) = 0$$

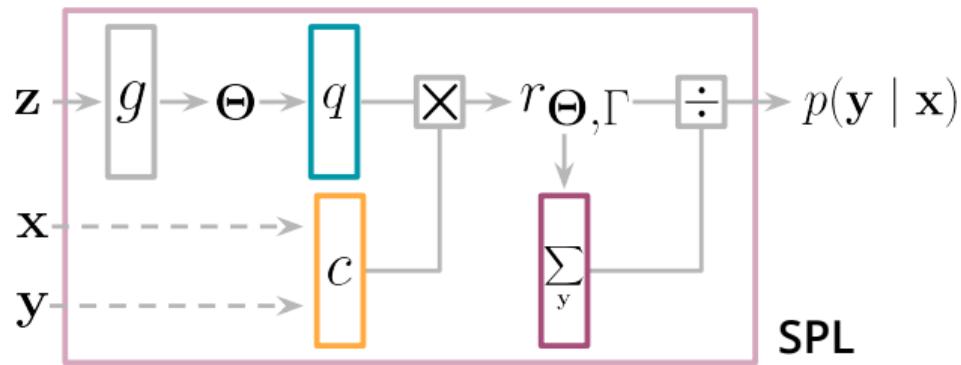
$$(\phi_{pc} \cdot c_K)(S, R, O)$$



logic + probability + tractability

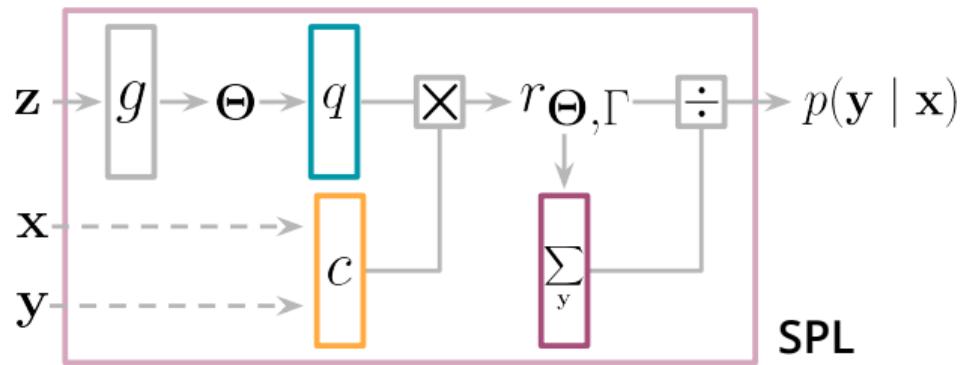


logic + probability + tractability

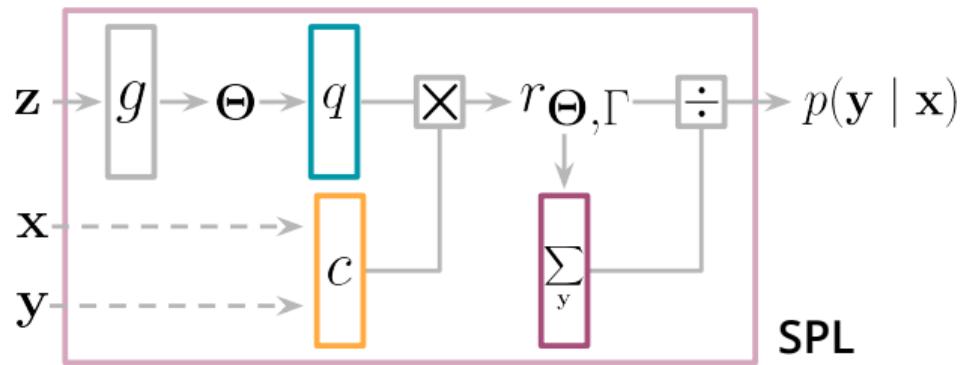


SPL

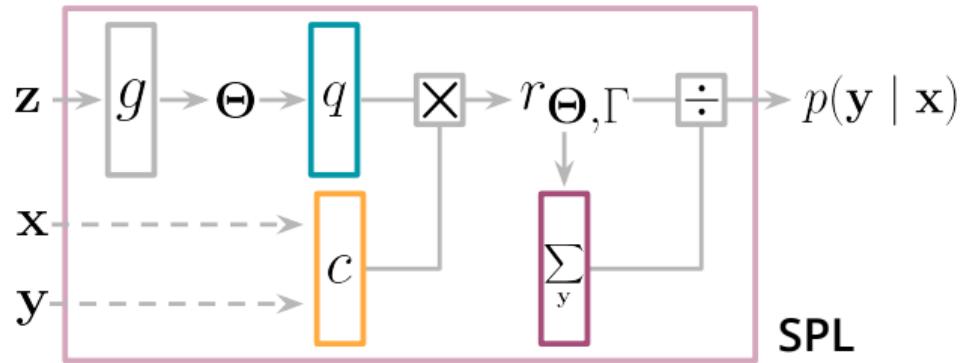
logic + probability + tractability



logic + probability + tractability



logic + ***probability*** + ***tractability***



$$p(\mathbf{y} \mid \mathbf{x}) = \mathbf{q}_{\Theta}(\mathbf{y} \mid g(\mathbf{z})) \cdot \mathbf{c}_{\kappa}(\mathbf{x}, \mathbf{y}) / \mathbf{Z}(\mathbf{x})$$