logically-consistent deep learning via probabilistic circuits

antonio vergari (he/him)

@tetraduzione

17th Oct 2024 - **PICS PhD School** - *Copenhagen*

probabilistic circuits (PCs)

A grammar for tractable computational graphs

I. *A simple tractable function is a circuit*

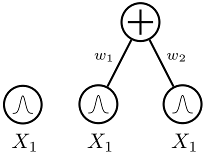
$\Rightarrow$ *e.g., a multivariate Gaussian, or a logical literal*

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*



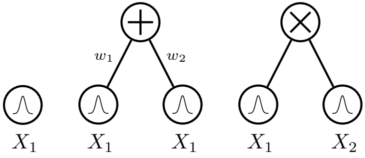
probabilistic circuits (PCs)

A grammar for tractable computational graphs

I. *A simple tractable function is a circuit*

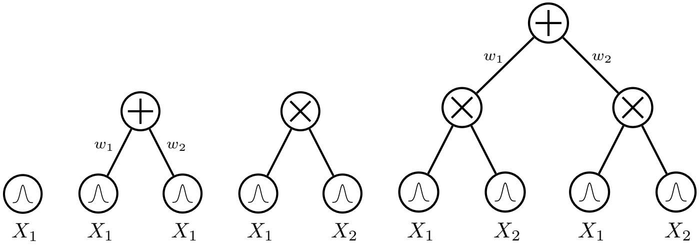
II. *A weighted combination of circuits is a circuit*

III. *A product of circuits is a circuit*



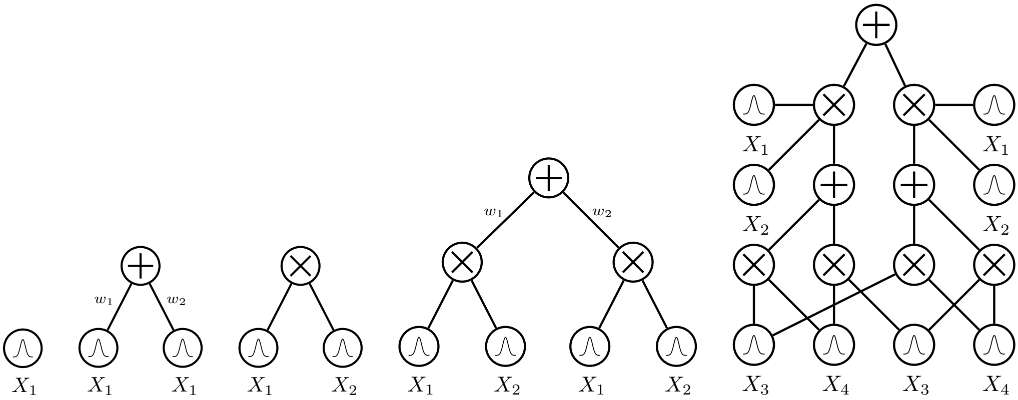
probabilistic circuits (PCs)

A grammar for tractable computational graphs



probabilistic circuits (PCs)

A grammar for tractable computational graphs



structural properties

smoothness

decomposability

compatibility

determinism

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

structural properties

smoothness

decomposability

compatibility

determinism

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

structural properties

smoothness

decomposability

compatibility

determinism

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

structural properties

smoothness

decomposability

compatibility

determinism

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

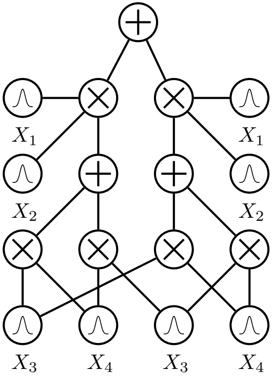
determinism + **decomposability** = **tractable MAP**

Computing maximization with arbitrary evidence $\mathbf{e}$

$\Longrightarrow$ *linear in circuit size!*

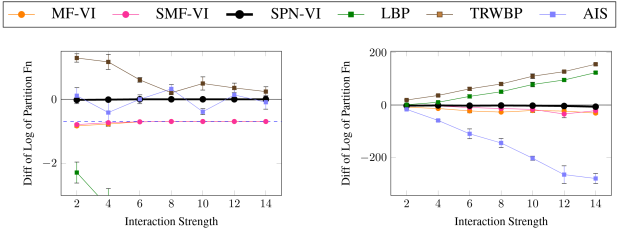
E.g., suppose we want to compute:

$$\max_{\mathbf{q}} p(\mathbf{q} \mid \mathbf{e})$$



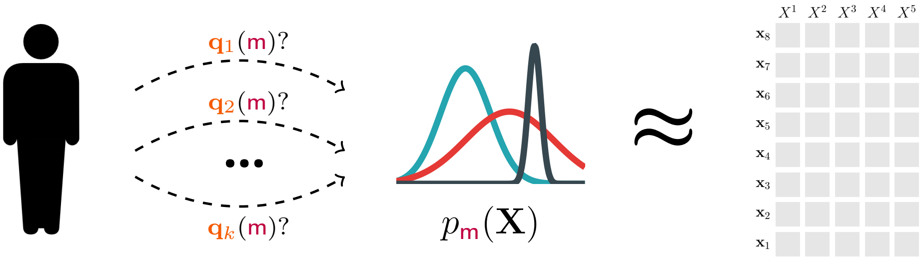
determinism + decomposability = tractable ELBO

Using deterministic and decomposable PCs as expressive variational family $\mathcal{Q}$ for discrete polynomial log-densities, i.e. $\operatorname{argmax}_{q \in \mathcal{Q}} \mathbb{E}_{\mathbf{x} \sim q} \left[\log w(\mathbf{x})\right] + \mathbb{H}(q)$



Closed-form computation for the entropy $\mathbb{H}$ *[**Vergari** et al. 2021]*

Shih and Ermon, "Probabilistic Circuits for Variational Inference in Discrete Graphical Models", NeurIPS, 2020

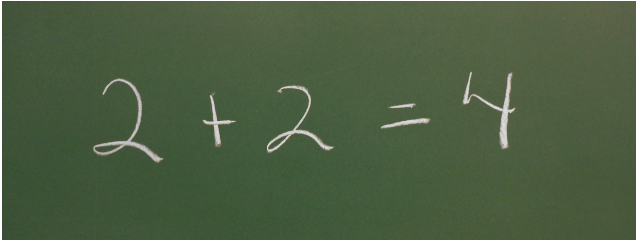


generative models that can reason probabilistically

…but some events are certain!

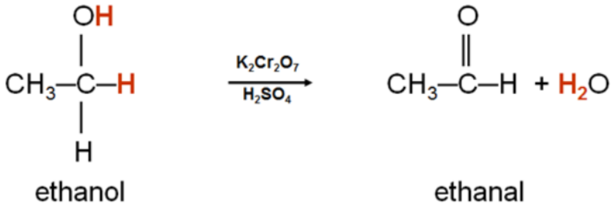
math reasoning

and logical deduction



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

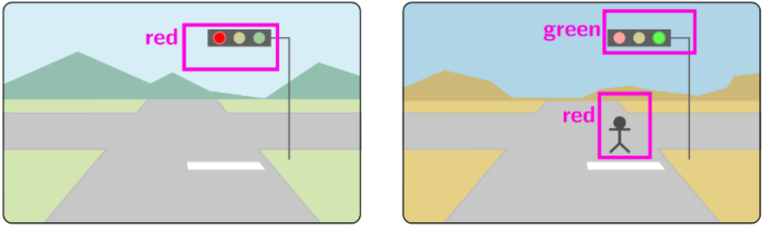
physics laws



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

AI safety

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



Constraints: traffic rules, scene understanding (objects do not disappear) ...

Marconato et al., "Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts", NeurIPS, 2023

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

code understanding

Reference

```
def find_length(string, n):
    current_sum = 0
    max_sum = 0
    for i in range(n):
        current_sum += (1 if string[i] == '0'
            ↪ else -1)
        if current_sum < 0:
            current_sum = 0
        max_sum = max(current_sum, max_sum)
    return max_sum if max_sum else 0
```

Transformed

```
def find_length(string, n):
    current_sum = 0
    max_sum = 0
    for i in range(n):
        current_sum += (1 if string[i] != '0'
            ↪ else -1)
        if current_sum < 0:
            current_sum = 0
        max_sum = max(current_sum, max_sum)
    return max_sum if max_sum else 0
```

Model	Prediction
Llama2-7B	YES ✖
Llama2-13B	YES ✖
CodeLlama-7B	YES ✖
CodeLlama-13B	YES ✖
CodeLlama-34B	YES ✖
StarCoder2-3B	YES ✖
StarCoder2-7B	YES ✖
StarCoder2-15B	YES ✖

Maveli, Vergari, and Cohen, "What can Large Language Models Capture about Code Functional Equivalence?", arXiv, 2024

what about valid molecules?

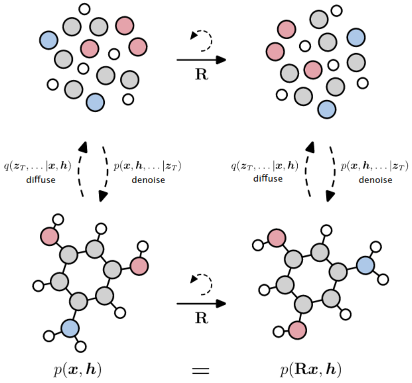
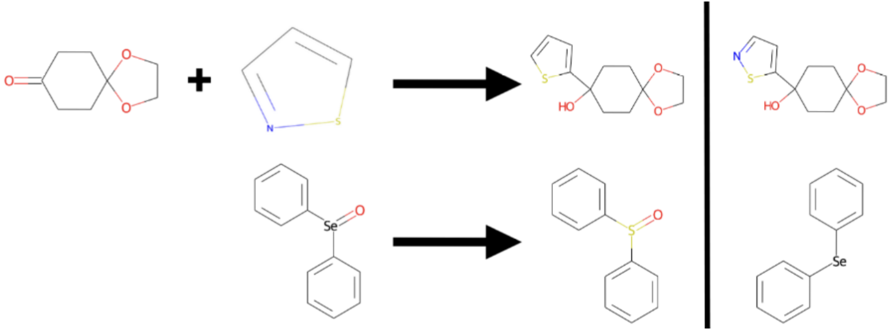


Table 2. Validity and uniqueness over 10000 molecules with standard deviation across 3 runs. Results marked (*) are not directly comparable, as they do not use 3D coordinates to derive bonds. H: model hydrogens explicitly

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

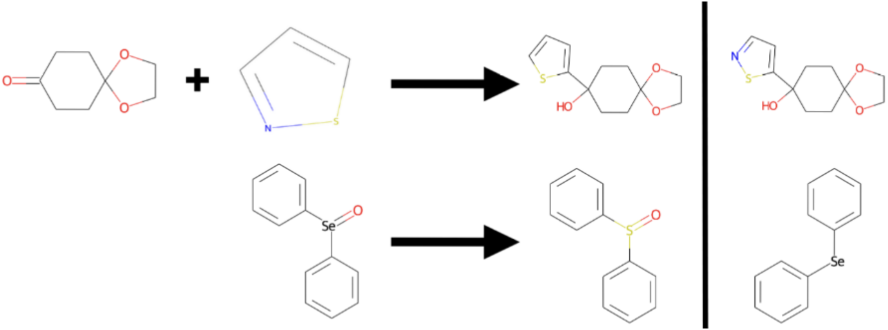
Hoogeboom et al., "Equivariant diffusion for molecule generation in 3d", International Conference on Machine Learning, 2022

and valid reactions?



"deep learning is doing alchemy"

and valid reactions?



CHEMALGEBRA: ALGEBRAIC REASONING BY PREDICTING CHEMICAL REACTIONS

planning

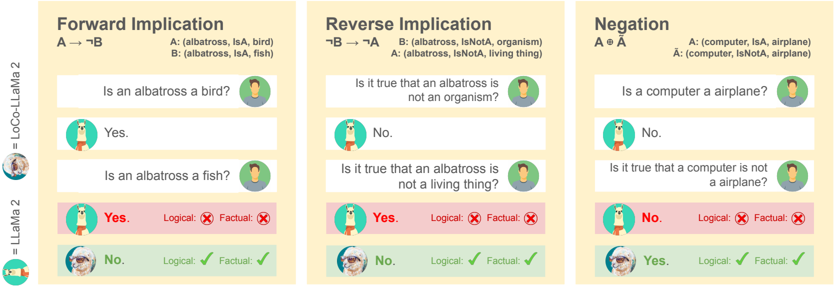
Can Large Language Models Reason and Plan?

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

logical inconsistency



LLMs confabulate and contradict themselves [1]

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

Logically Consistent Language Models via Neuro-Symbolic Integration

Diego Calanzone*
DISI, University of Trento
diego.calanzone@studenti.unitn.it

Stefano Teso
CIMeC & DISI, University of Trento
stefano.teso@unitn.it

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

Goal

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

the issues!

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

$\Longrightarrow$ ***a single framework*** *for implications, negation, paths, hierarchies, ...*

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 implications, negation, paths, hierarchies, ...*

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 implications, negation, paths, hierarchies, ...*

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****!*

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

which logic?

or which kind of constraints to represent?

propositional logic (zeroth-order)

$$(a \land b) \lor d \implies c$$

first-order logic (FOL)

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

satisfiability modulo theory (SMT)

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)

which logical consistency?

factuality

we know that some facts f in a KB are true.

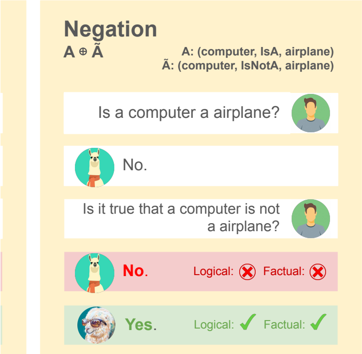
f_1 : *"an albatross is a bird"*

how to query an LLM?

$$p_\theta(z_f = \top) = p_\theta(x_t = \ell_{\text{true}} \mid x_1, \ldots, x_{t-1} = \text{"Is an albatross a bird?"})$$

which logical consistency?

negation



if we know the following fact

f : *"an albatross is a bird"*

and what to query the truth value $z_{\widetilde{f}}$ of

$\widetilde{f}$: *"an albatross is **not** a bird"*

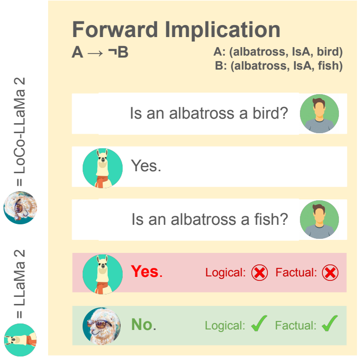
because f is the **negation** of $\widetilde{f}$:

$$z_f \oplus z_{\widetilde{f}} \iff (z_f \land \neg z_{\widetilde{f}}) \lor (\neg z_f \land z_{\widetilde{f}})$$

we expect the answer to be that $\widetilde{f}$ is **false**.

which logical consistency?

implication



if we know the following fact

f_1 : *"an albatross is a bird"*

and what to query the truth value z_{f_2} of

f_2 : *"an albatross is an animal"*

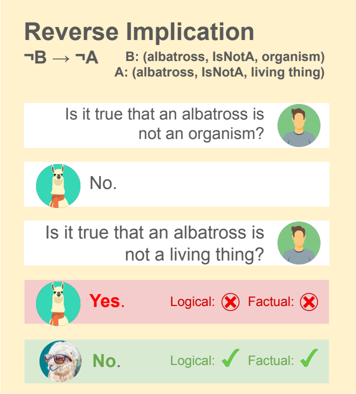
because f_1 **implies** f_2:

$$(z_{f_1} \rightarrow z_{f_2}) \iff (\neg z_{f_1} \vee z_{f_2})$$

we expect the answer to be that f_2 is **true**.

which logical consistency?

reverse implication



we can **reverse an implication**

$$z_{\widetilde{f}_2} \rightarrow z_{\widetilde{f}_1}$$

where $\widetilde{f}_2$: *"an albatross is **not** an animal"*

and we ask if the following is true

$\widetilde{f}_1$: *"an albatross is **not** a bird"*

we expect the answer to be that $\widetilde{f}_1$ is **true**.

how

to enforce constraints?

$$\max p_\theta(\mathsf{K}_i)$$

maximise the probability of the constraint to hold!

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

how

to enforce constraints?

$$\min \mathcal{L}(\mathsf{K}_i, p_\theta) = \min -\log \sum_{\mathbf{z} \models \mathsf{K}_i} \prod_{j:\mathbf{z} \models z_{f_j}} p_\theta(z_{f_j}) \prod_{j:\mathbf{z} \models \neg z_{f_j}} (1 - p_\theta(z_{f_j}))$$

minimize the semantic loss

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

$$p_\theta(\mathsf{K}(\mathbf{z})) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\mathbb{1}\{\mathbf{z} \models \mathsf{K}\}]$$

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

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

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

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

assuming independence of $\mathbf{z}$ (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

WMC

computing the probability of logical formulas

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

computing WMC is #P-hard in general : (

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

more complex constraints

EntailmentBank

$$(z_{f_1} \wedge z_{f_2} \rightarrow z_{f_3}) \wedge z_{f_4} \rightarrow z_{f_5}$$

f_1 : *"melting is a kind of phase change"*

f_2 : *"the ice melts"*

f_3 : *"the ice undergoes a phase change"*

f_4 : *"phase changes do not change mass"*

f_5 : *"the mass of the ice will not change"*

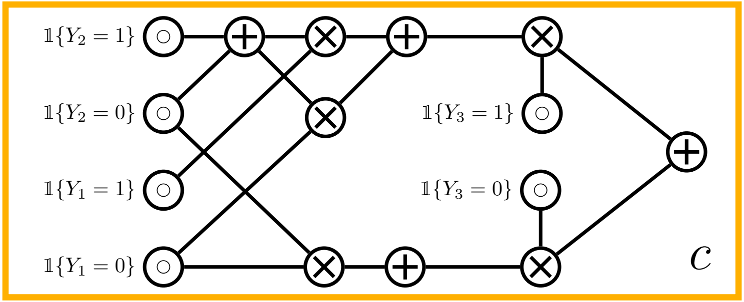
Goal

Can we encode K
to yield a tractable WMC?

Goal

Can we encode K
to yield a tractable WMC?
yes, as a circuit!

semantic loss



compiling logical formulas into circuits

knowledge compilation

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

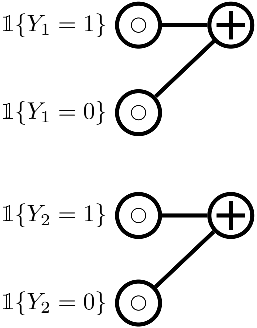
$\mathbb{1}\{Y_1 = 0\}$ $\mathbb{1}\{Y_1 = 1\}$ $\mathbb{1}\{Y_2 = 0\}$

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

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

knowledge compilation

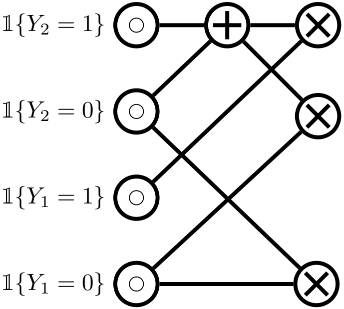
$$\mathsf{K}: \ (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \ \ (Y_2 = 1 \implies Y_3 = 1)$$



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

knowledge compilation

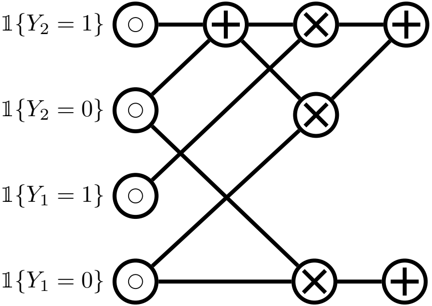
$$\mathsf{K}:\ (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge\ (Y_2 = 1 \implies Y_3 = 1)$$



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

knowledge compilation

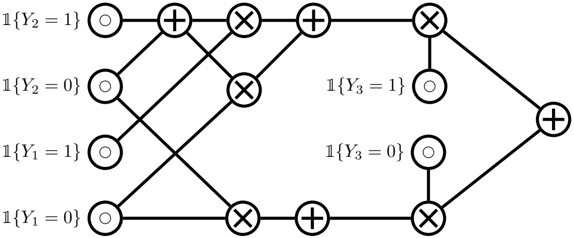
$$\mathsf{K}:\ (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge\ (Y_2 = 1 \implies Y_3 = 1)$$



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

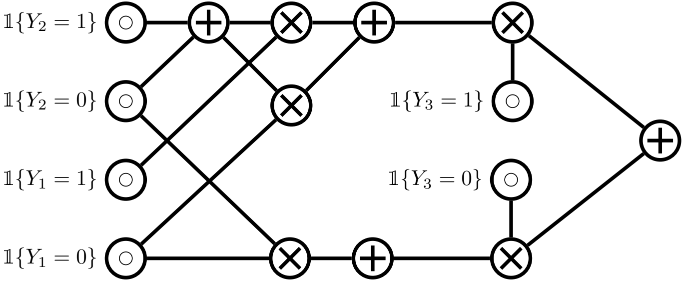
knowledge compilation

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



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

tractable WMC



exactly compute WMC in time $O(|c|)$

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

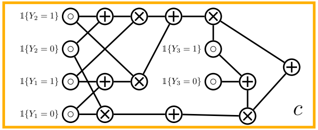
SL recipe

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

1) Take a
logical constraint

SL recipe

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

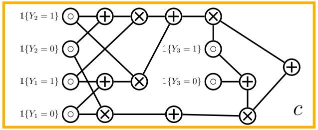


1) Take a logical constraint

2) Compile it into a constraint circuit

SL recipe

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



$$-\log \mathrm{WMC}(\mathsf{K}_i, p_\theta)$$

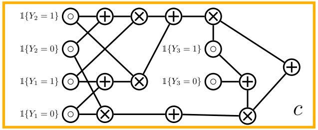
1) Take a logical constraint

2) Compile it into a constraint circuit

3) minimize the semantic loss

SL recipe

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



$$-\log \mathrm{WMC}(\mathsf{K}_i, p_\theta)$$

1) Take a logical constraint

2) Compile it into a constraint circuit

3) minimize the semantic loss

4) train end-to-end by sgd!

MODEL	TRAIN	PPL	CONSISTENCY FAC	CONSISTENCY IMP	REV	SELF-CONSISTENCY NEG	SELF-CONSISTENCY IMP	REV	AVG
LLaMA-2-7B Zero Shot		62.41	0.39	0.52	0.13	0.42	0.30	0.15	0.32
LLaMA-2-7B Few Shot		52.30	0.53	0.71	0.34	0.38	0.48	0.47	0.48
LLaMA-2-7B CoT		52.30	0.52	0.64	0.67	0.40	0.64	0.67	0.59
LLaMA-2-70B Zero Shot		44.90	0.47	0.69	0.81	0.13	0.31	0.91	0.55
LLaMA-2-7B + XENT	T1+T2	116.85	0.25	0.46	0.01	0.07	0.81	0.01	0.27
LoCo-LLaMA-2-7B (NEG)	T1	62.21	0.44	0.65	0.43	**0.96**	0.28	0.36	0.52
LoCo-LLaMA-2-7B (F-IMP)	T1	67.15	**0.99**	**0.99**	0.07	0.00	**0.99**	0.07	0.51
LoCo-LLaMA-2-7B (Super)	T1	62.23	0.74	0.77	**0.77**	0.87	0.71	**0.77**	**0.77**

greatly improving (self-)consistency

evaluate on unseen constraints

EntailmentBank

$$(z_{f_1} \wedge z_{f_2} \rightarrow z_{f_3}) \wedge z_{f_4} \rightarrow z_{f_5}$$

f_1 : *"melting is a kind of phase change"*
f_2 : *"the ice melts"*
f_3 : *"the ice undergoes a phase change"*
f_4 : *"phase changes do not change mass"*
f_5 : *"the mass of the ice will not change"*

MODEL	DEPTH				
	1	2	3	4	5
LLaMa-2-7B	0.87	0.76	0.59	0.61	0.63
LoCo-LLaMa-2-7B (NEG)	0.51	0.51	0.51	0.52	0.52
LoCo-LLaMa-2-7B (F-IMP)	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**
LoCo-LLaMa-2-7B (SUPER)	0.69	0.68	0.68	0.68	0.69

finetune on BeliefBank, test on EntailmentBank

nice...

...but!

assuming facts to be independent...

nice...

...but!

assuming facts to be independent...

no guarantees to satisfy constraints at test time...

nice...

...but!

assuming facts to be independent...

no guarantees to satisfy constraints at test time...

WMC

on the independence assumption

$$K : \neg \mathbf{r} \vee \neg \mathbf{g}$$

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

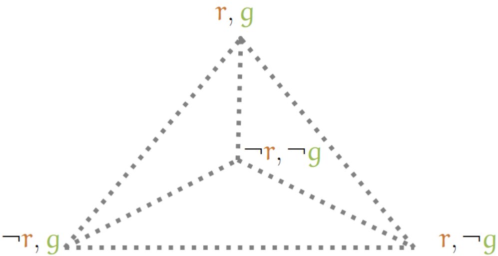
van Krieken et al., "On the Independence Assumption in Neurosymbolic Learning", ICML, 2024

WMC

on the independence assumption

$$K : \neg \mathbf{r} \vee \neg \mathbf{g}$$

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



van Krieken et al., "On the Independence Assumption in Neurosymbolic Learning", ICML, 2024

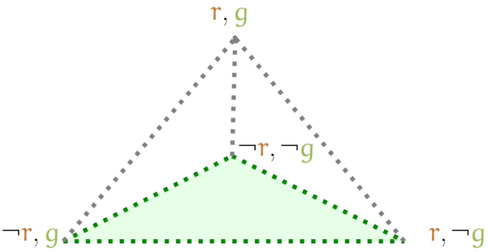
WMC

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)



van Krieken et al., "On the Independence Assumption in Neurosymbolic Learning", ICML, 2024

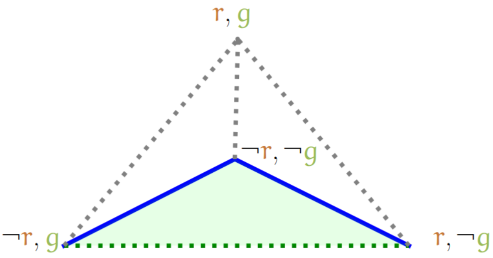
WMC

on the independence assumption

$$\mathsf{K} : \neg \mathbf{r} \vee \neg \mathbf{g}$$

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

but assuming $p(\mathbf{r}, \mathbf{g}) = p(\mathbf{r})p(\mathbf{g})$ *restricts this even further (only blue lines)*



van Krieken et al., "On the Independence Assumption in Neurosymbolic Learning", ICML, 2024

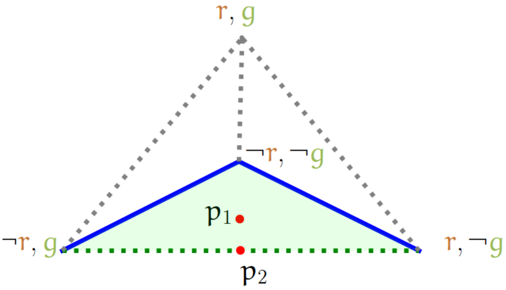
WMC

on the independence assumption

$$K : \neg \mathbf{r} \vee \neg \mathbf{g}$$

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

but assuming $p(\mathbf{r}, \mathbf{g}) = p(\mathbf{r})p(\mathbf{g})$ *restricts this even further (only blue lines)*



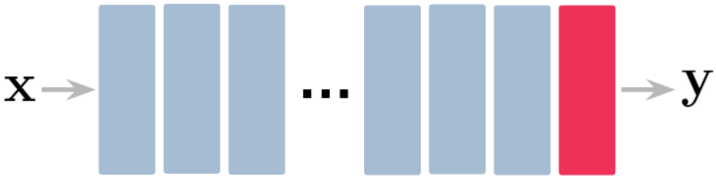
nice...

...but!

assuming facts to be independent...

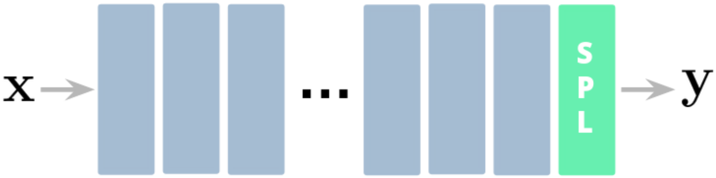
no guarantees to satisfy constraints at test time...

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

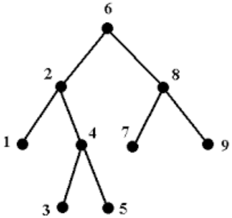
s.t. $\mathbf{y} \models \mathsf{K}$ *// e.g., that form a valid path*

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

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

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

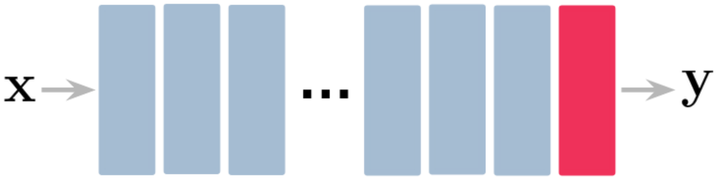
hierarchical multi-label classification

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

How?

"which neural network architecture to use?"

e.g.,

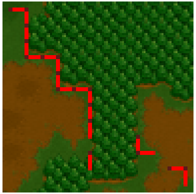


sigmoid linear layers

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

When?

Ground Truth

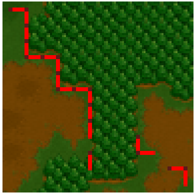


ResNet-18

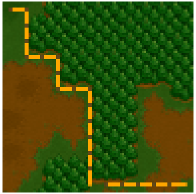
neural nets struggle to satisfy validity constraints!

Constraint losses

Ground Truth



ResNet-18

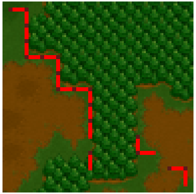


Semantic Loss

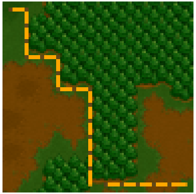
...but cannot guarantee consistency at test time!

SPL

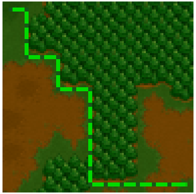
Ground Truth



ResNet-18



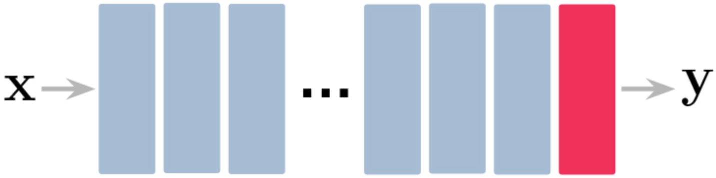
Semantic Loss



SPL (ours)

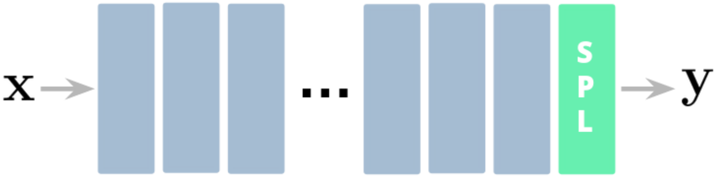
you can predict valid paths 100% of the time!

How?



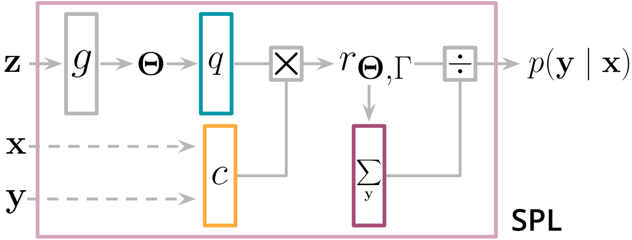
take an unreliable neural network architecture...

How?



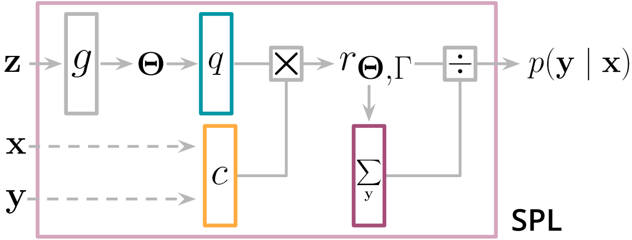
......and replace the last layer with
a semantic probabilistic layer

SPL



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

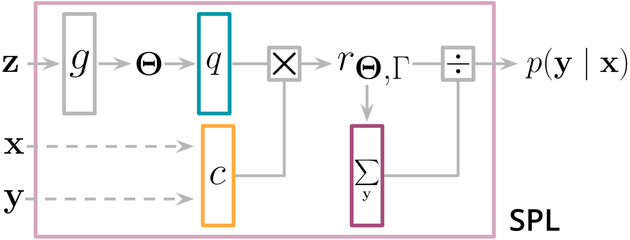
SPL



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

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

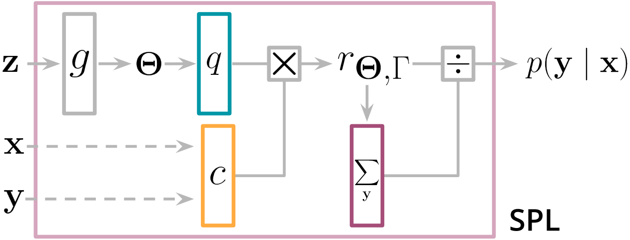
SPL



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

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

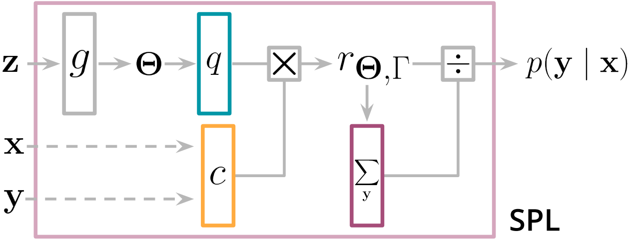
SPL



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

a product of experts : (

SPL



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

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

Goal

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

Goal

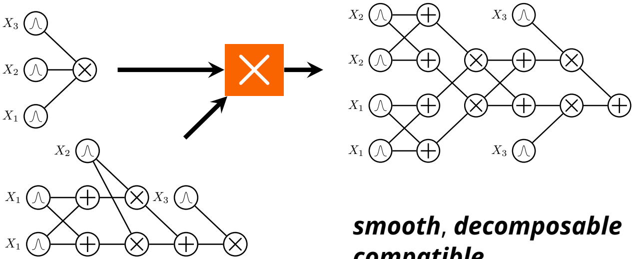
Can we design q and c
to be deep computational graphs
yet yielding a tractable product?

Goal

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

yes! as *circuits*!

Tractable products

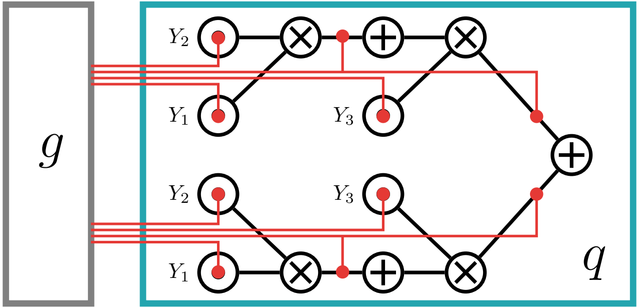


smooth**, **decomposable
compatible

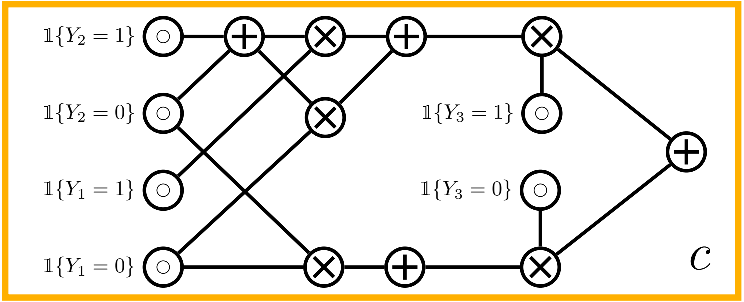
exactly compute $\mathcal{Z}$ in time $O(|\boldsymbol{q}||\boldsymbol{c}|)$

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

SPL

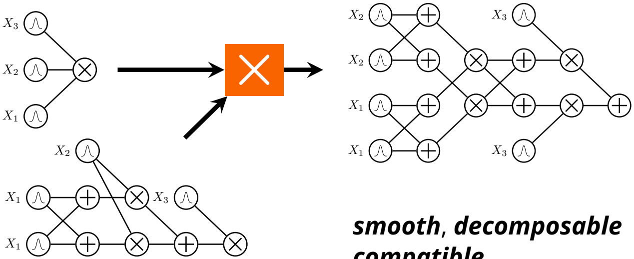


a conditional circuit $q(\mathbf{y}; \Theta = g(\mathbf{z}))$



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

Tractable products



smooth, ***decomposable***
compatible

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

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

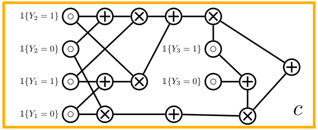
SPL recipe

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

1) Take a
logical constraint

SPL recipe

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

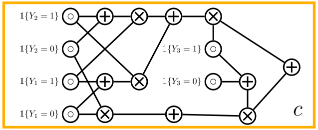


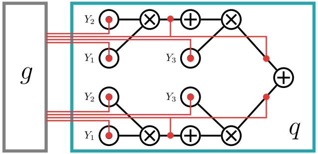
1) Take a logical constraint

2) Compile it into a constraint circuit

SPL recipe

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





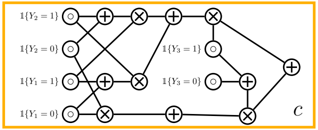
1) Take a
logical constraint

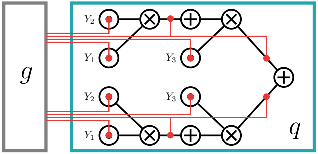
2) Compile it into
a constraint circuit

3) Multiply it
by a circuit distribution

SPL recipe

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





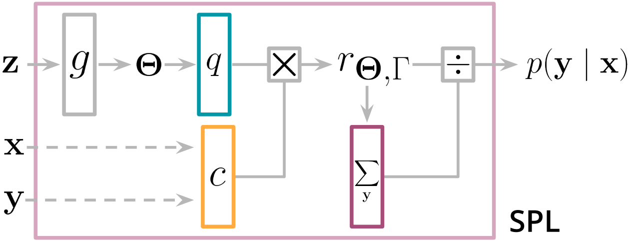
1) Take a logical constraint

2) Compile it into a constraint circuit

3) Multiply it by a circuit distribution

4) train end-to-end by sgd!

Experiments

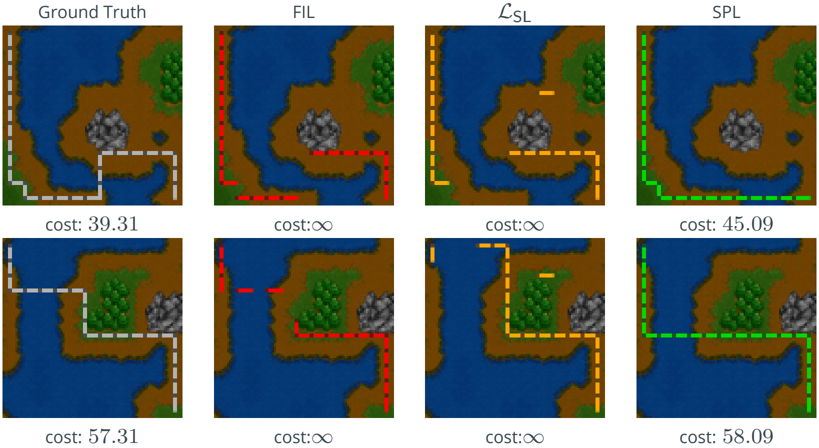


how good are SPLs?

Experiments

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



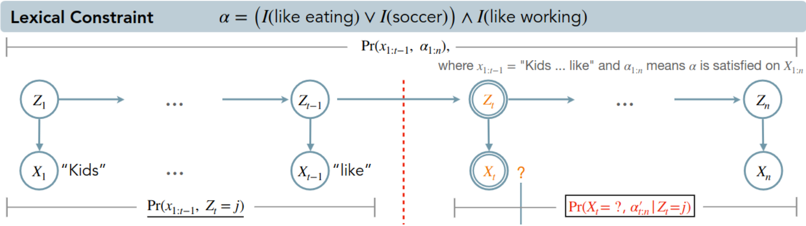
SPLs

(and more circuits)

everywhere

Tractable Control for Autoregressive Language Generation

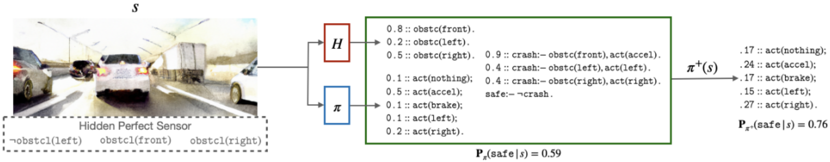
Honghua Zhang [*1] Meihua Dang [*1] Nanyun Peng [1] Guy Van den Broeck [1]



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)

How to Turn Your Knowledge Graph Embeddings into Generative Models

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

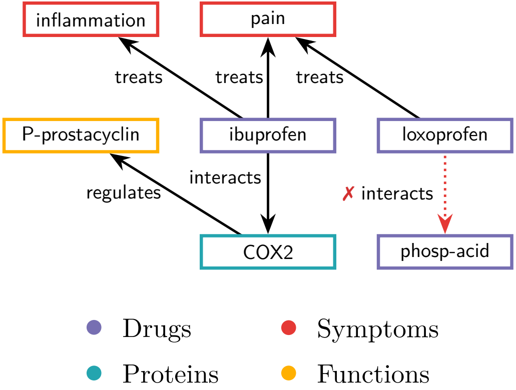
Nicola Di Mauro
University of Bari, Italy
nicola.dimauro@uniba.it

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

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

enforce constraints in knowledge graph embeddings
oral at NeurIPS 2023

constraints in KGs



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

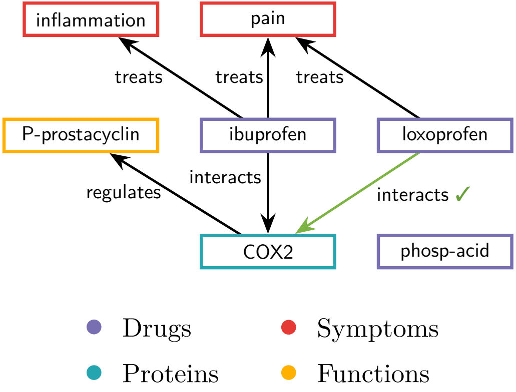
$\mathcal{A}$: $\langle$loxoprofen, interacts, **phosp-acid**$\rangle$

✘

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

✔

constraints in KGs



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

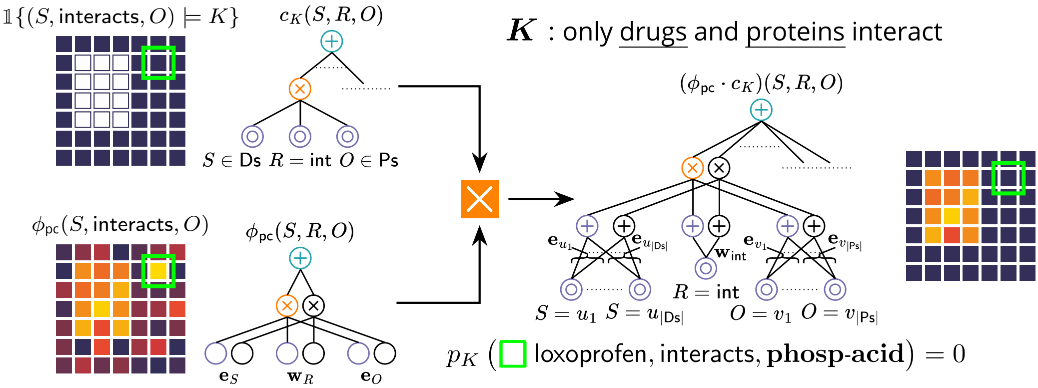
$\mathcal{A}$: ⟨loxoprofen, interacts, **phosp-acid**⟩

✘

$\mathcal{A}$: ⟨loxoprofen, interacts, **COX2**⟩

✔

guaranteed satisfaction of constraints



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

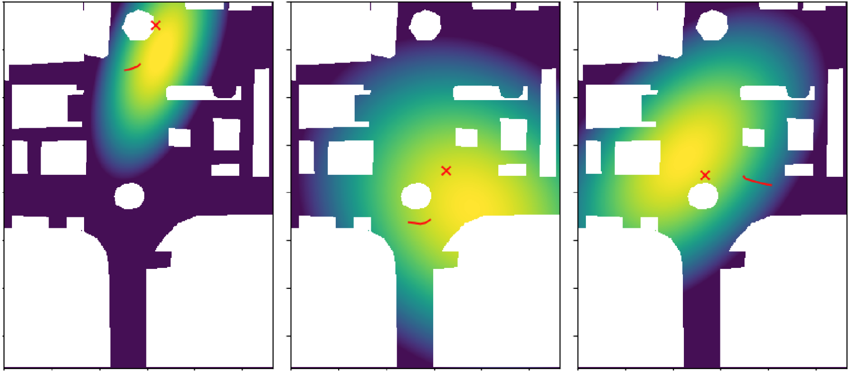
open problems

I constraints over continuous variables

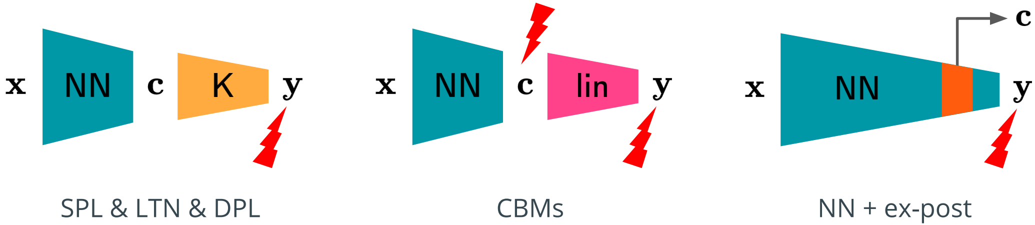
II scaling to H U G E constraints

III learn (partial) constraints

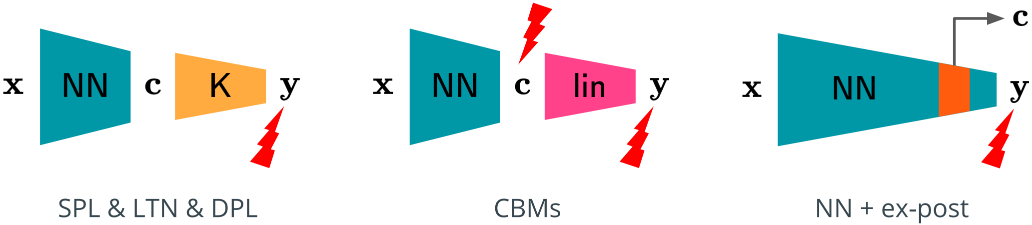
IV revise constraints (continual learning)



extending it to SMT constraints



NeSy models are concept bottlenecks

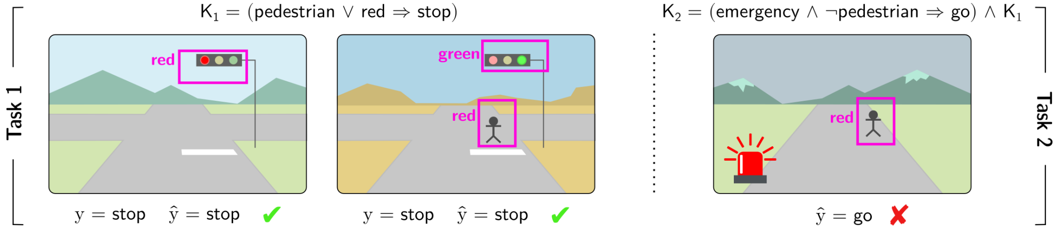


NeSy models can suffer from reasoning shortcuts!

Task	Example Data	Knowledge K	Example RS	Impact
MNIST math	$\begin{cases} 2 \cdot 2 + 2 & = 6 \\ 3 + 4 & = 7 \end{cases}$	Equations must hold.	$\begin{cases} 2 \to 2 \\ 3 \to 4 \\ 4 \to 3 \end{cases}$	$2 + 4 = 5$

NeSy models can suffer from reasoning shortcuts!

Bortolotti et al., "A Benchmark Suite for Systematically Evaluating Reasoning Shortcuts", NeurIPS Benchmark track, 2024



how to detect and mitigate them

Marconato et al., "Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts", NeurIPS, 2023

Bortolotti et al., "A Benchmark Suite for Systematically Evaluating Reasoning Shortcuts", NeurIPS Benchmark track, 2024

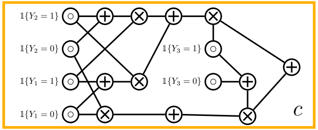
colorai

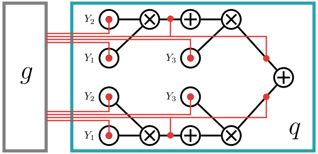
connecting low-rank representations in ai

workshop at AAAI-25, Philadelfia

april-tools.github.io/colorai/

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





questions?