Integrating Logic and Prob ML
a neuro-symbolic perspective

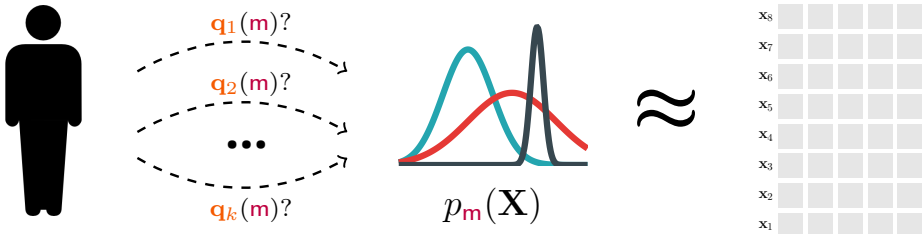
antonio vergari (he/him)

 *@tetraduzione*

11th Mar 2024 - **Advanced Probabilistic Modeling** - *University of Trento*

in the previous episodes...

why
(advanced) probabilistic ML?

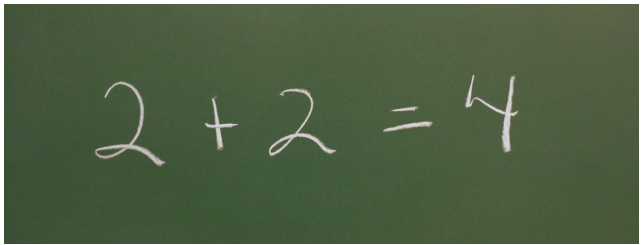


generative models that can reason probabilistically

...but some events are certain!

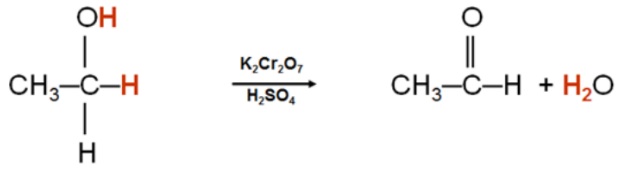
math reasoning

and logical deduction



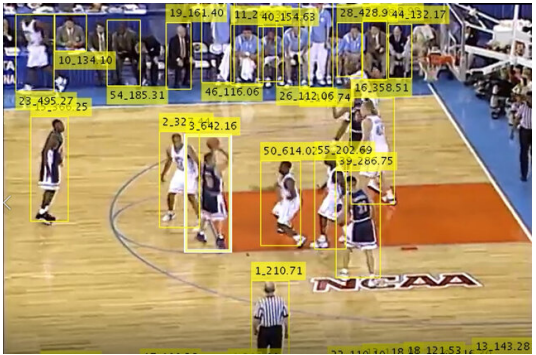
Constraints: carrying out arithmetic tasks, but also ***proving theorems***

physics laws



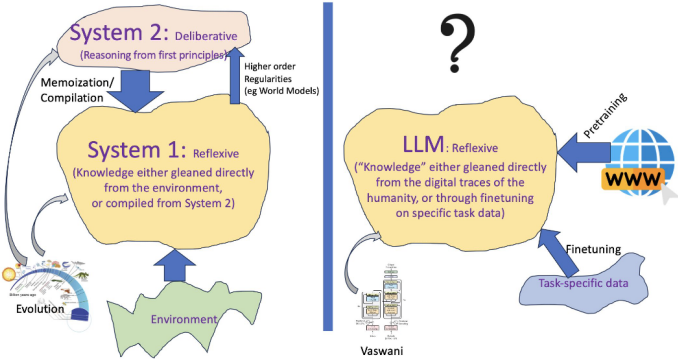
Constraints: preserving #atoms, #electrons (RedOx), ...in chemical reactions

scene understanding



Constraints: object permanence (players do not disappear), role preserving (each player has a fixed role), ...

planning

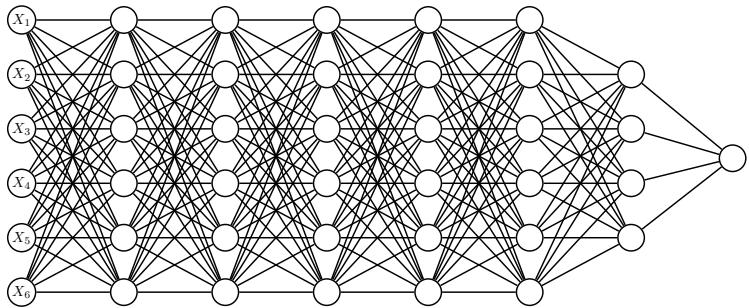


Constraints: design and execute plans to solve complex goals

Kambhampati, "Can large language models reason and plan?", Annals of the New York Academy of Sciences, 2024

the tabula rasa approach

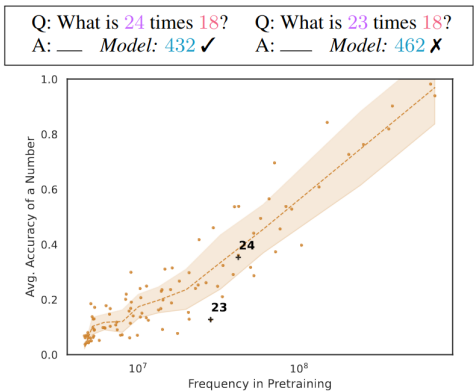
or "do we really need (external) hard constraints?"



a deep feedforward NN is a universal approximator...should be able to learn also hard constraints from data ?!

data efficiency

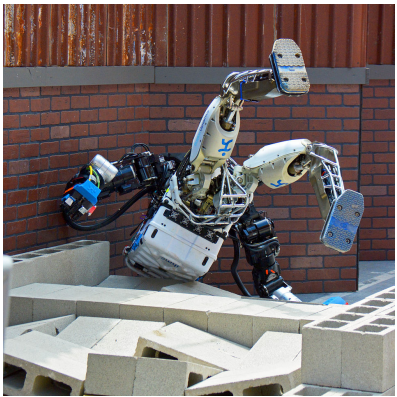
there is never enough data



Razeghi et al., "Impact of Pretraining Term Frequencies on Few-Shot Numerical Reasoning", ,

reliability

the classical motivation against "soft constraints are good enough"



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"

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

why ***deep generative models*** ...

[PROS] they encode ***expressive*** distributions with ***millions of parameters***

+

they ***scale learning*** to high-dimensional, large datasets via ***GPU***

...and why they struggle with ***hard constraints***

[PROS] they encode ***expressive*** distributions with ***millions of parameters***

+

they ***scale learning*** to high-dimensional, large datasets

[CONS] they are ***limited to soft constraints*** (probabilities)

+

their computations come with ***little or no guarantees*** of satisfying the constraints (unreliable)

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

logical inconsistency

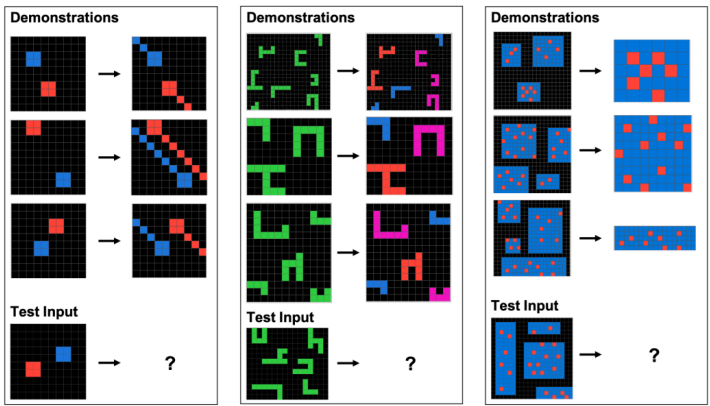
	Method	Train Subset	Antecedents F_1	Consequents F_1	Total F_1	Logical consistency
RQ1	ConCoRD				0.91	0.91
	Macaw-Large		0.52	0.90	0.81	0.83
	SFT	T1	0.13	0.01	0.03	0.72
	LoCo-LM	T1	0.79	0.98	0.96	0.99
RQ2	SFT	T1+T2 (5%)	0.23	0.78	0.72	0.82
	LoCo-LM	T1+T2 (5%)	0.67	0.83	0.81	0.92
	SFT	T1+T2 (10%)	0.55	0.97	0.91	0.90
	LoCo-LM	T1+T2 (10%)	0.45	0.97	0.89	0.93
	SFT	T1+T2 (75%)	0.85	0.99	0.97	0.98
	LoCo-LM	T1+T2 (75%)	0.79	0.99	0.95	0.98

LLMs confabulate and contraddict themselves [1]

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

abstraction?

generalizing through logic



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?", Annals of the New York Academy of Sciences, 2024

what about valid molecules?

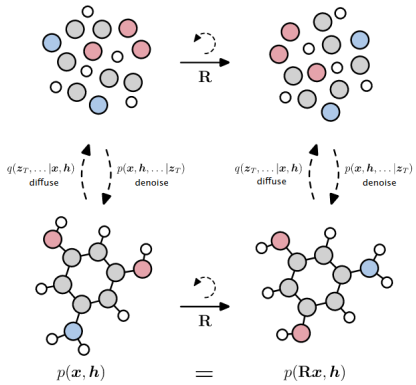
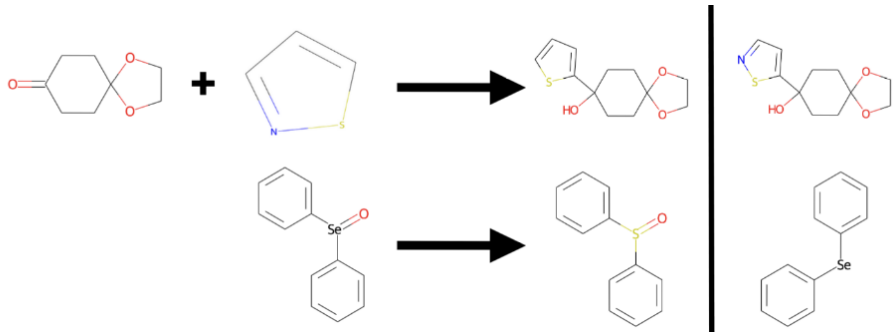


Table 2. Validity and uniqueness over 10000 molecules with standard deviation across 3 runs. Results marked (*) are not directly comparable, as they do not use 3D coordinates to derive bonds. H: model hydrogens explicitly

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

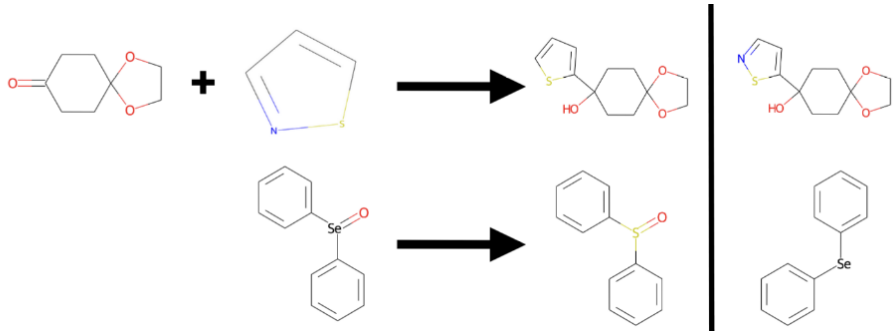
Hoogeboom et al., "Equivariant diffusion for molecule generation in 3d", ICML, 2022

and valid reactions?



“deep learning is doing alchemy”

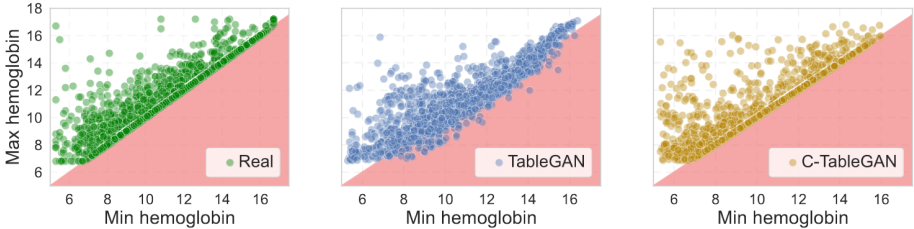
and valid reactions?



CHEMALGEBRA: ALGEBRAIC REASONING BY PREDICTING CHEMICAL REACTIONS

generating tabular data

simpler generative task?



violating even simple min/max constraints

Stoian et al., "How Realistic Is Your Synthetic Data? Constraining Deep Generative Models for Tabular Data", arXiv preprint arXiv:2402.04823, 2024

deep learning is differentiable lego

just stacking blocks can be unrealiable though...

*especially when dealing with **hard constraints***

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

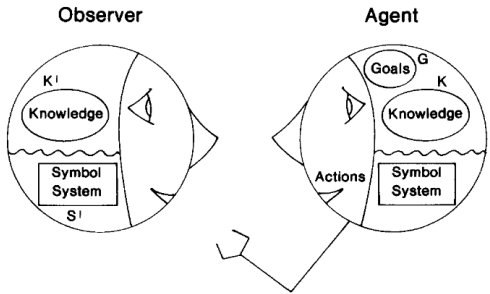
with deep neural nets

solution

probabilistic neuro-symbolic AI

...but what are symbols?

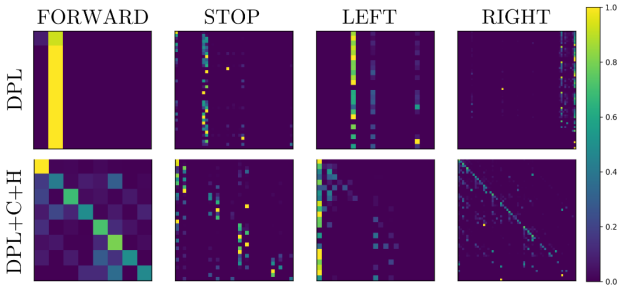
symbols



a symbol is a map between vocabularies

Newell, "The knowledge level", Artificial intelligence, 1982

symbols



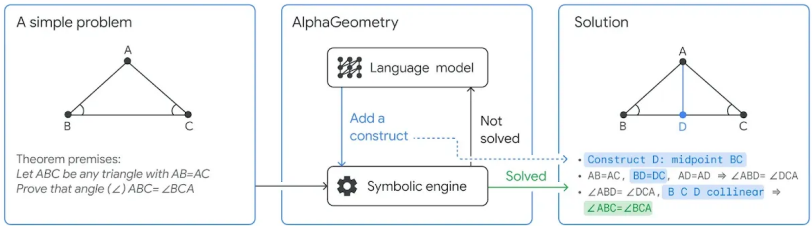
aligning vocabularies is extremely hard

Marconato et al., "Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts", Advances in Neural Information Processing Systems, 2024

so...
so how to integrate
probabilities + neural nets + logic?

prob NeSy AI

many flavors (I)

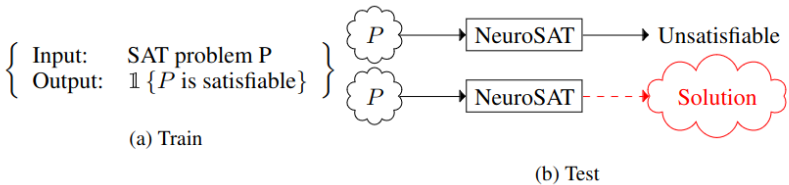


neural search: generate & test

Trinh et al., "Solving olympiad geometry without human demonstrations", Nature, 2024

prob NeSy AI

many flavors (II)

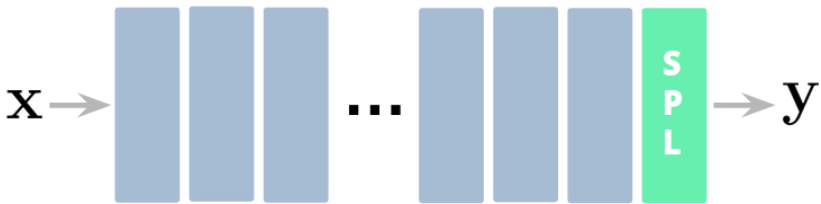


neural surrogates of symbolic systems

Selsam et al., "Learning a SAT solver from single-bit supervision", arXiv preprint arXiv:1802.03685, 2018

prob NeSy AI

many flavors (III)



integrating symbolic systems in neural nets

Ahmed et al., "Semantic Probabilistic Layers for Neuro-Symbolic Learning", , 2022

...and more!

	Dimension 1 (D)irected (U)ndirected	Dimension 2 (G)rounding (P)roofs	Dimension 3 (L)ogic (P)robability (N)eural	Dimension 4 (L)ogic (P)robability (F)uzzy	Dimension 5 (P)arameter (S)tructure	Dimension 6 (S)ymbols (Sub)symbols	Dimension 7 (P)ropositional (R)elational (FOL) (LP)
∂ILP [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

and more

De Raedt et al., "From statistical relational to neuro-symbolic artificial intelligence", arXiv preprint arXiv:2003.08316, 2020

Goal

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

Why?

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

integrate ***hard*** **(logical)** constraints

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)

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

higher-order logic (HOL)

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

which logic?

discrete vs continuous

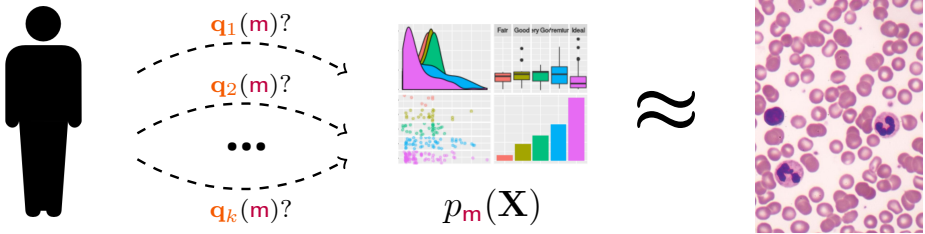
propositional logic (boolean variables)

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

satisfiability modulo theory (SMT)

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

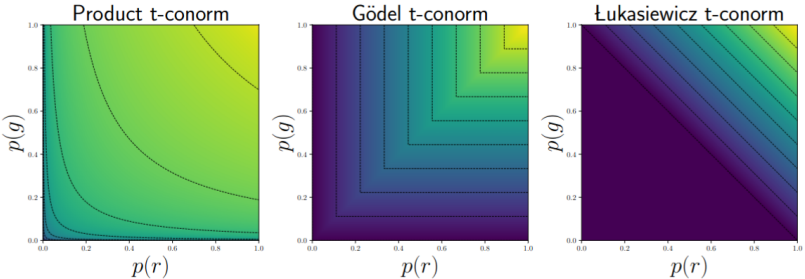
"What's the probability of the red blood cell count to exceed θ or the number of white cells? "



$$\mathbf{q}_k(\mathbf{m}) := p_{\mathbf{m}}((X_{\#\text{red}} \geq \theta) \vee (X_{\#\text{red}} \geq X_{\#\text{white}}))$$

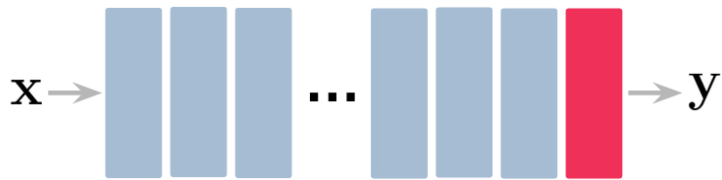
Zeng et al., "Scaling up Hybrid Probabilistic Inference with Logical and Arithmetic Constraints via Message Passing", ICML, 2020

fuzzy logic



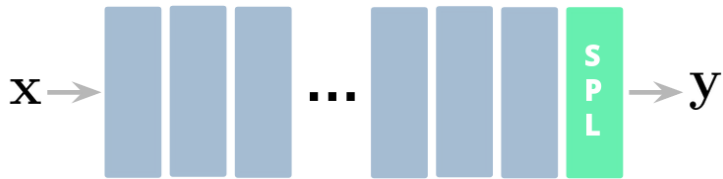
van Krieken, "Optimisation in Neurosymbolic Learning Systems", arXiv preprint arXiv:2401.10819, 2024

Why?



make any neural network architecture...

Why?



...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", ICLR, 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*

nesy structured output prediction (SOP) tasks

Vlastelica et al., "Differentiation of blackbox combinatorial solvers", ICLR, 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", ICLR, 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*

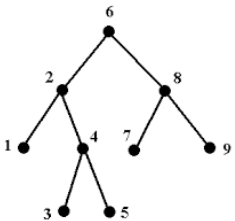
s.t. $\mathbf{y} \models \mathsf{K}$ *// e.g., that form a valid path*

// 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", ICLR, 2020

When?



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

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

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

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

hierarchical multi-label classification

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

When?

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

find $\mathbf{y}^* = \text{argmax}_{\mathbf{y}}\, p_\theta(\mathbf{y} \mid \mathbf{x})$ *// 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", IJCAI, 2015

When?

given $\mathbf{x}$ *// e.g., a pair of MNIST images*

find $\mathbf{y}^* = \text{argmax}_{\mathbf{y}}\, p_\theta(\mathbf{y} \mid \mathbf{x})$

$= \text{argmax}_{\mathbf{y}} \sum_{c_1,c_2} p_\theta(\mathbf{y}, c_1, c_2 \mid \mathbf{x})$ *// e.g. sum of two digit classes*

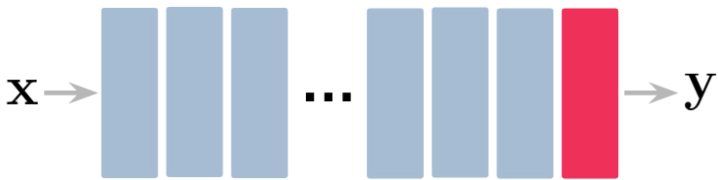
s.t. $\mathbf{y}, c_1, c_2 \models \mathsf{K}$ *// e.g., output valid sums*

// 5+3=8

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", , 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", , 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", , 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", , 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 $\mathbf{y}$

Xu et al., "A Semantic Loss Function for Deep Learning with Symbolic Knowledge", , 2018

Constraint losses

Ground Truth

ResNet-18

Semantic Loss

...but cannot guarantee consistency at test time!

what?

	Losses			Layers				
Desideratum	DL2 [29]	SL [80]	NeSyEnt [3]	FIL	EBM [43]	MultiplexNet [38]	CCN [33]	SPL *(ours)*
(D1) Probabilistic	✗	✓	✓	✓	✗	✓	✗	✓
(D2) Expressive	✗	✗	✗	✗	✓	✗	✗	✓
(D3) Consistent	✗	✗	✗	✗	✗	✓	✓	✓
(D4) General	✓	✓	✓	✗	✓	✓	✗	✓
(D5) Modular	✓	✓	✓	✓	✓	✓	✓	✓
(D6) Efficient	✓	✓	✓	✓	✗	✗	✓	✓

SPL

Ground Truth

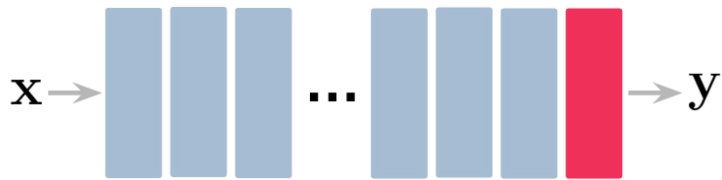
ResNet-18

Semantic Loss

SPL (ours)

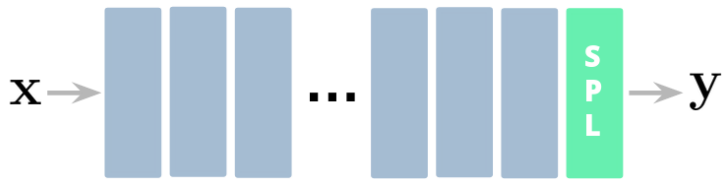
you can predict valid paths 100% of the time!

How?



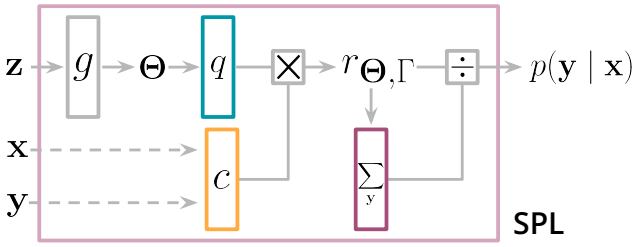
take an unreliable neural network architecture...

How?

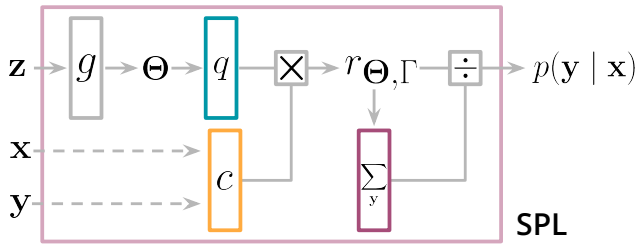


.....and replace the last layer with a semantic probabilistic layer

SPL



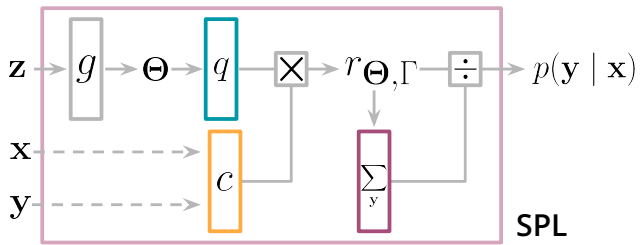
SPL



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

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

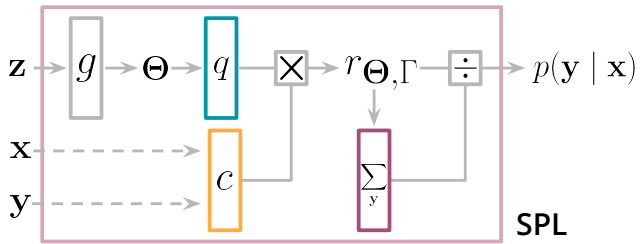
SPL



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

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

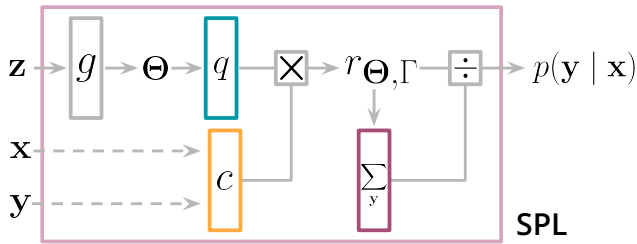
SPL



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

a product of experts : (

SPL



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

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

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?

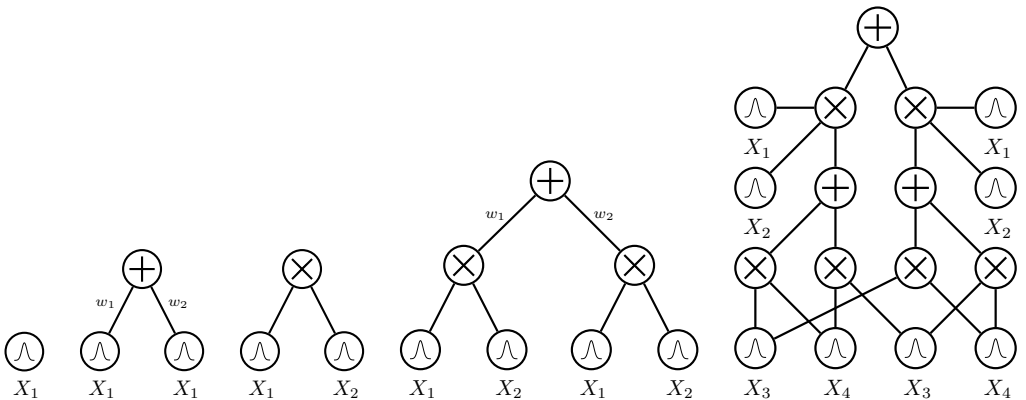
Goal

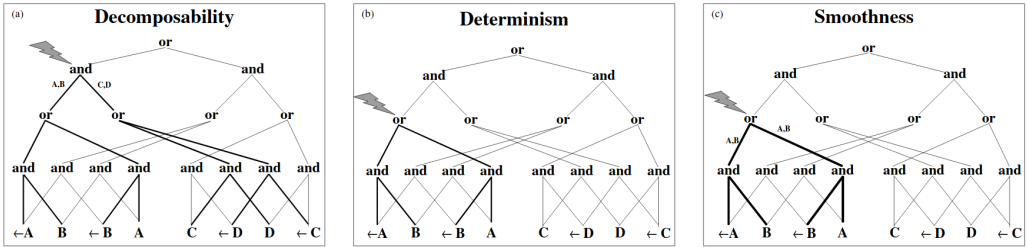
Can we design q and c
to be deep computational graphs
yet yielding a tractable product?

yes! as *circuits*!

Probabilistic Circuits (PCs)

A grammar for tractable computational graphs

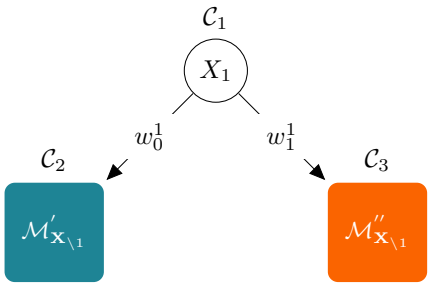


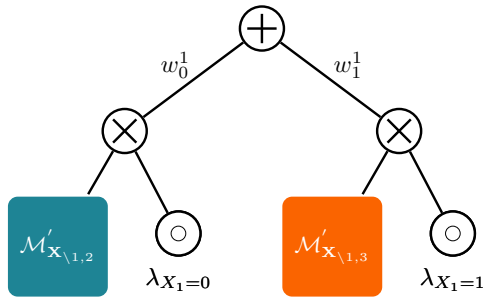


PCs generalize Logical circuits: sd-DNNFs, (O)BDDs,...

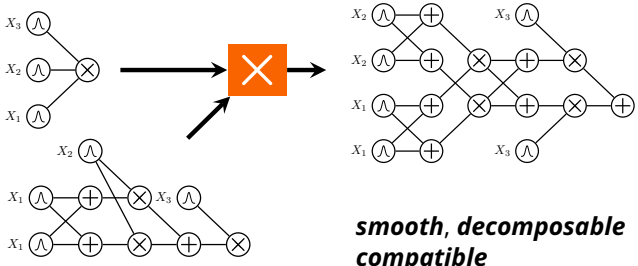
Darwiche and Marquis, "A knowledge compilation map", Journal of Artificial Intelligence Research, 2002

A/B/SDDs as PCs...





Tractable products

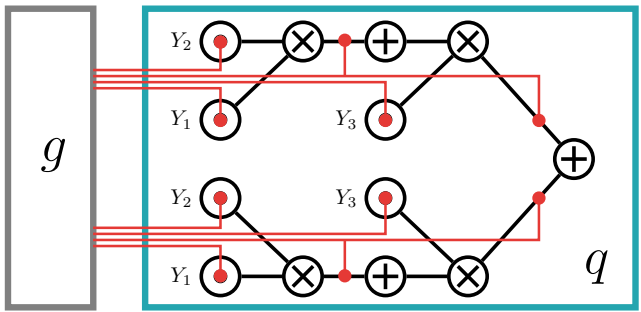


smooth, decomposable compatible

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

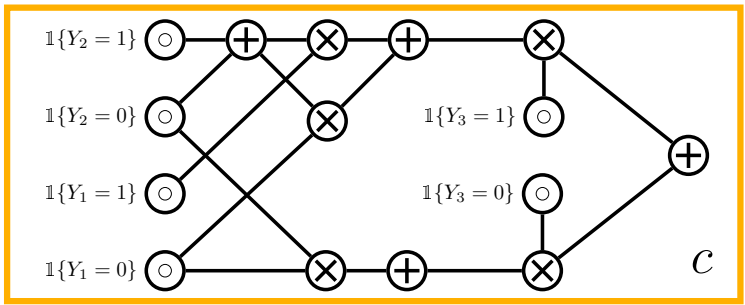
***Vergari** et al., "A Compositional Atlas of Tractable Circuit Operations: From Simple Transformations to Complex Information-Theoretic Queries", NeurIPS, 2021*

SPL



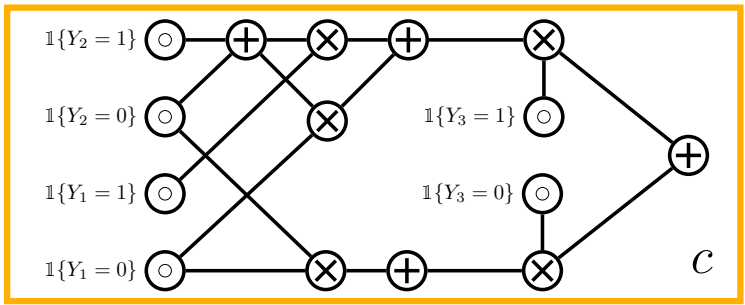
a conditional circuit $q(\mathbf{y}; \Theta = g(\mathbf{z}))$

SPL



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

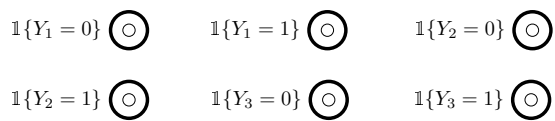
SPL



compiling logical formulas into circuits

Knowledge compilation

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

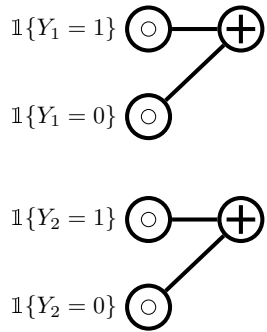


Pipatsrisawat and Darwiche, "New Compilation Languages Based on Structured Decomposability.", AAAI, 2008

Knowledge compilation

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

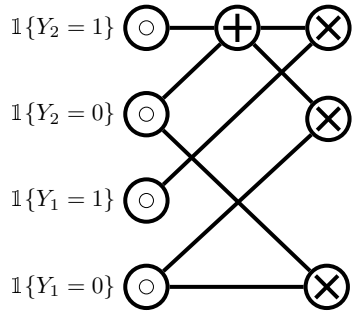
$$\wedge \; (Y_2 = 1 \implies Y_3 = 1)$$



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

Knowledge compilation

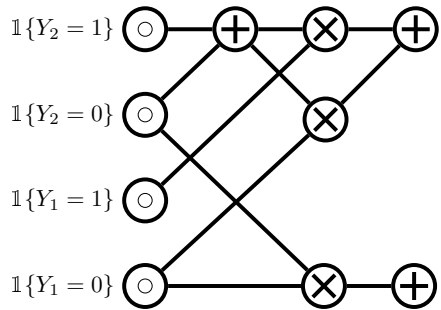
$$\mathsf{K}: \ (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \ \ (Y_2 = 1 \implies Y_3 = 1)$$



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

Knowledge compilation

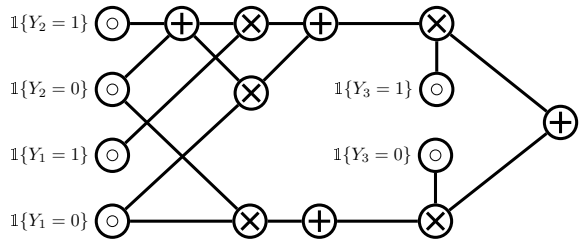
$$\mathsf{K}: \ (Y_1 = 1 \implies Y_3 = 1)$$
$$\wedge \ \ (Y_2 = 1 \implies Y_3 = 1)$$



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

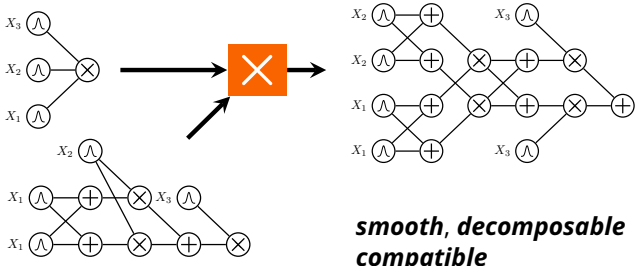
Knowledge compilation

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



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

Tractable products



smooth, decomposable compatible

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

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

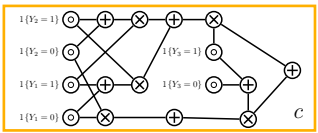
SPL recipe

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

1) Take a
logical constraint

SPL recipe

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

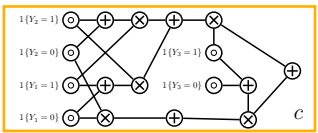


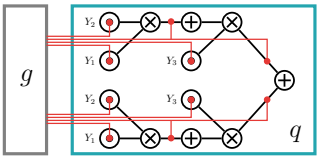
1) Take a logical constraint

2) Compile it into a constraint circuit

SPL recipe

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





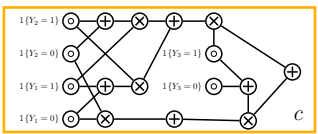
1) Take a logical constraint

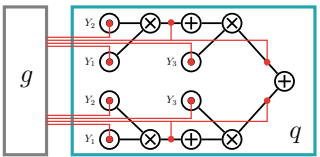
2) Compile it into a constraint circuit

3) Multiply it by a circuit distribution

SPL recipe

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





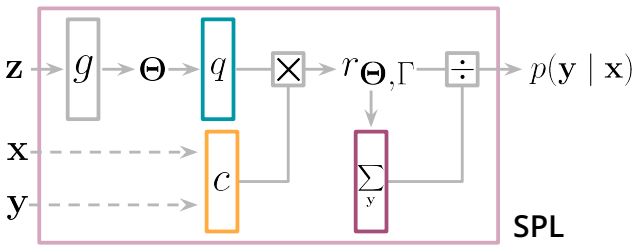
1) Take a logical constraint

2) Compile it into a constraint circuit

3) Multiply it by a circuit distribution

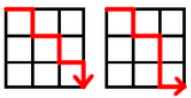
4) train end-to-end by sgd!

Experiments



how good are SPLs?

Experiments

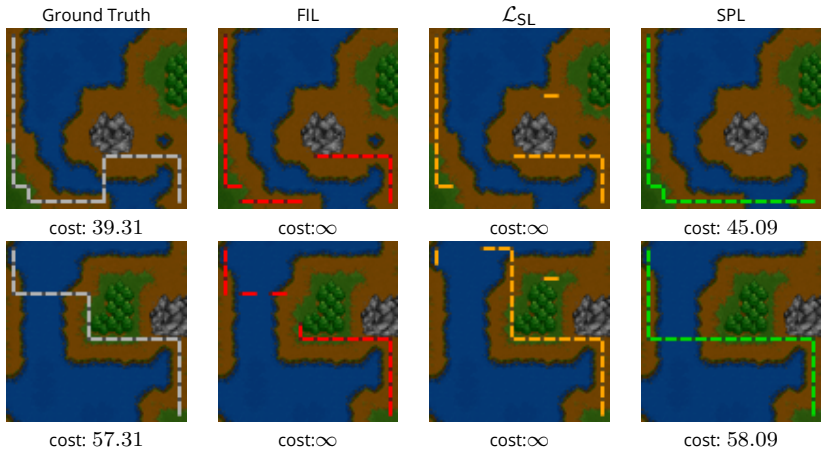


	Simple Path			Preference Learning		
Architecture	Exact	Hamming	Consistent	Exact	Hamming	Consistent
MLP+FIL	5.6	85.9	7.0	1.0	**75.8**	2.7
MLP+$\mathcal{L}_{\text{SL}}$	28.5	83.1	75.2	15.0	72.4	69.8
MLP+NeSyEnt	30.1	83.0	91.6	18.2	71.5	96.0
MLP+SPL	**37.6**	**88.5**	**100.0**	**20.8**	72.4	**100.0**

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



Experiments

Dataset	Exact Match	
	HMCNN	MLP+SPL
CellCycle	3.05 ± 0.11	$\mathbf{3.79 \pm 0.18}$
Derisi	1.39 ± 0.47	$\mathbf{2.28 \pm 0.23}$
Eisen	5.40 ± 0.15	$\mathbf{6.18 \pm 0.33}$
Expr	4.20 ± 0.21	$\mathbf{5.54 \pm 0.36}$
Gasch1	3.48 ± 0.96	$\mathbf{4.65 \pm 0.30}$
Gasch2	3.11 ± 0.08	$\mathbf{3.95 \pm 0.28}$
Seq	5.24 ± 0.27	$\mathbf{7.98 \pm 0.28}$
Spo	$\mathbf{1.97 \pm 0.06}$	1.92 ± 0.11
Diatoms	48.21 ± 0.57	$\mathbf{58.71 \pm 0.68}$
Enron	5.97 ± 0.56	$\mathbf{8.18 \pm 0.68}$
Imclef07a	79.75 ± 0.38	$\mathbf{86.08 \pm 0.45}$
Imclef07d	76.47 ± 0.35	$\mathbf{81.06 \pm 0.68}$

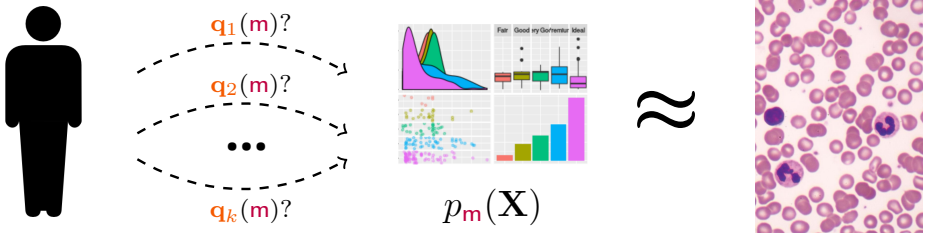
open problems

- **I** constraints over continuous variables
- **II** scaling to huge constraints
- **III** learn (partial) constraints
- **IV** revise constraints (continual learning)

hybrid SPL

Can we design SPL or SL
for constraints with
mixed continuous and discrete variables?

"What's the probability of the red blood cell count to exceed θ or the number of white cells? "



$$\mathbf{q}_k(\mathbf{m}) := p_{\mathbf{m}}((X_{\#\text{red}} \geq \theta) \vee (X_{\#\text{red}} \geq X_{\#\text{white}}))$$

Zeng et al., "Scaling up Hybrid Probabilistic Inference with Logical and Arithmetic Constraints via Message Passing", ICML, 2020

SMT + **weights**

$$\bigwedge_i 0 \leq X_{P_i} \leq 10$$
$$\bigwedge_j \bigwedge_{i \in T_j} \mid X_{T_j} - X_{P_i} \mid < 1$$
$$\bigwedge_j (B_{S_j} \Rightarrow X_{T_j} > 2)$$

SMT formula $\triangle$

+

$$\begin{cases} w(X_{P_i}), \\ \quad \text{if } 0 \leq X_{P_i} \leq 10 \\ \\ w(X_{T_j}, X_{P_i}), \\ \quad \text{if } \mid X_{T_j} - X_{P_i} \mid < 1 \\ \\ w(B_{S_j}, X_{T_j}), \\ \quad \text{if } B_{S_j} \Rightarrow X_{T_j} > 2 \end{cases}$$

weight functions $\mathcal{W}$

SMT + **weights** = **Weighted Model Integration**

$$\bigwedge_i 0 \leq X_{P_i} \leq 10$$
$$\bigwedge_j \bigwedge_{i \in T_j} \mid X_{T_j} - X_{P_i} \mid < 1$$
$$\bigwedge_j (B_{S_j} \Rightarrow X_{T_j} > 2)$$

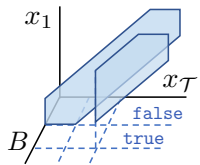
complex support

+

$$\begin{cases} w(X_{P_i}), & \text{if } 0 \leq X_{P_i} \leq 10 \\ w(X_{T_j}, X_{P_i}), & \text{if } \mid X_{T_j} - X_{P_i} \mid < 1 \\ w(B_{S_j}, X_{T_j}), & \text{if } B_{S_j} \Rightarrow X_{T_j} > 2 \end{cases}$$

densities

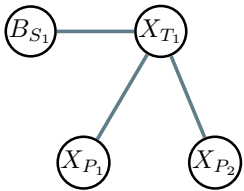
=



(***unnormalized***)

$p_\Delta(\mathbf{X}, \mathbf{B})$

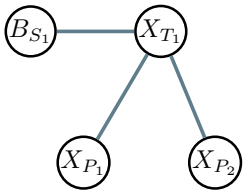
Exact probabilistic inference on SMT formulas



Tractability in terms of the ***primal graph*** $\mathcal{G}$ of an SMT formula

Zeng et al., "Scaling up Hybrid Probabilistic Inference with Logical and Arithmetic Constraints via Message Passing", ICML, 2020

Exact probabilistic inference on SMT formulas



Tractability in terms of the ***primal graph*** $\mathcal{G}$ of an SMT formula

If $\mathcal{G}$ has treewidth = 1 and logarithmic diameter

$\Rightarrow$ ***polytime****! via message passing*

If $\mathcal{G}$ has treewidth = 1 and linear diameter

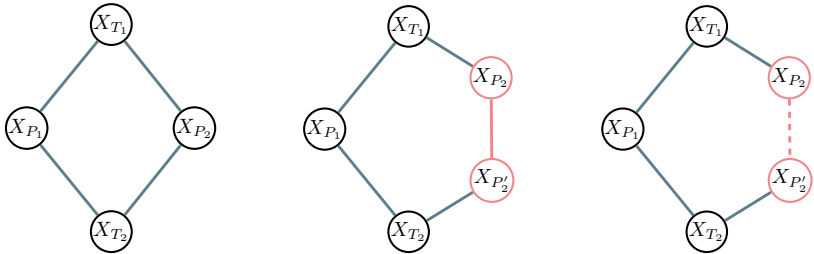
$\Rightarrow$ ***#P-Hard!***

If $\mathcal{G}$ has treewidth > 1 and logarithmic diameter

$\Rightarrow$ ***#P-Hard!***

Zeng et al., "Scaling up Hybrid Probabilistic Inference with Logical and Arithmetic Constraints via Message Passing", ICML, 2020

Approximate inference on SMT formulas



Performing exact inference on one approximate (relaxed) SMT problem

Zeng et al., "Probabilistic Inference with Algebraic Constraints: Theoretical Limits and Practical Approximations", , 2020

part I how to satisfy wanted constraints in NNs by design

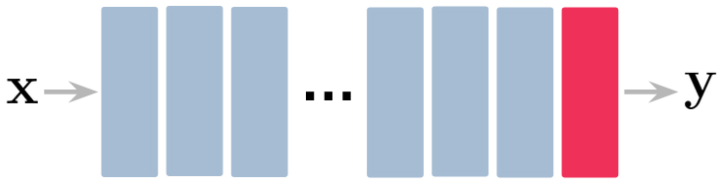
part II how the design of NNs implicitly poses unwanted constraints

Taming the Sigmoid Bottleneck: Provably Argmaxable Sparse Multi-Label Classification

Andreas Grivas[1], **Antonio Vergari**[†2], **Adam Lopez**[†1]

[1] Institute for Language, Cognition, and Computation
[2] Institute for Adaptive and Neural Computation
School of Informatics, University of Edinburgh
{agrivas, avergari, alopez}@ed.ac.uk

accepted at AAAI24



sigmoid linear layers

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

low-rank classifiers

labels (n) $>>$ embedding size (d)

e.g.,

clinically annotated text

$n \approx 9000$
$d \approx 200 - 500$

large-scale biomedical question answering

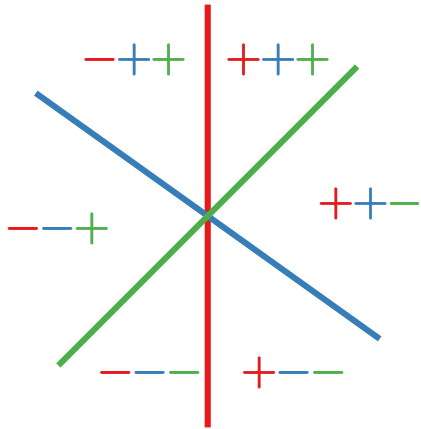
$n \approx 20000$
$d \approx 200 - 500$

OpenImages Dataset
object recognition

$n \approx 9000$
$d \approx 2000$

$n = 3$

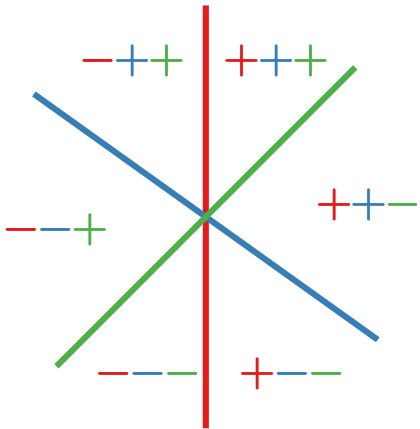
$d = 2$



some label configurations are *unargmaxable*!

$n = 3$

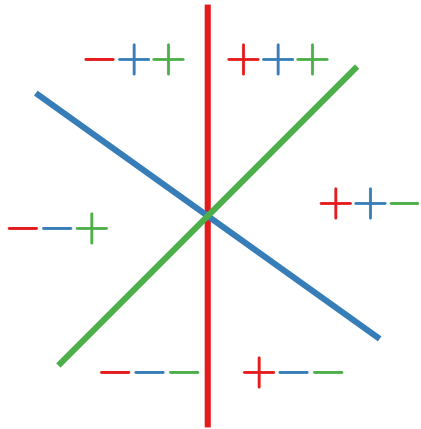
$d = 2$



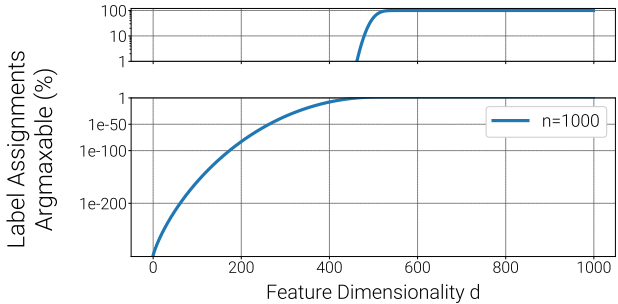
$$\nexists \mathbf{x} : \operatorname*{argmax}_{\mathbf{y}'} p(\mathbf{y}' \mid \mathbf{x}; \mathbf{W}) = \mathbf{y}$$

$n = 3$

$d = 2$



−+− and +−+ are ***unargmaxable***!



exponentially **many configurations are unargmaxable**

but real data is sparse...

K-active labels

clinically annotated text

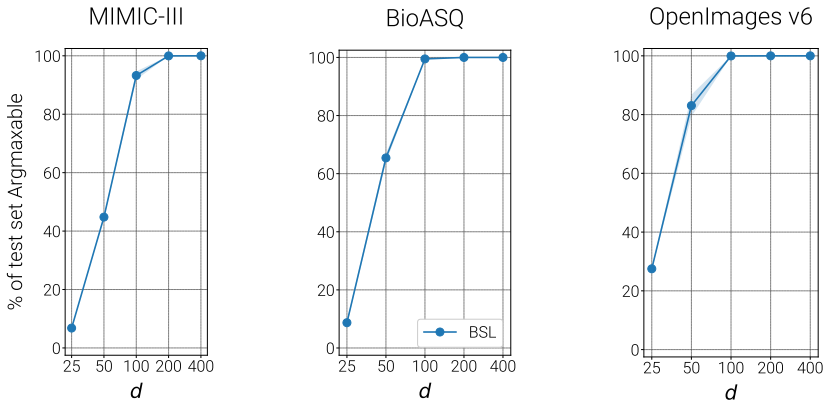
$n \approx 9000$
$K = 80$

large-scale biomedical question answering

$n \approx 20000$
$K = 50$

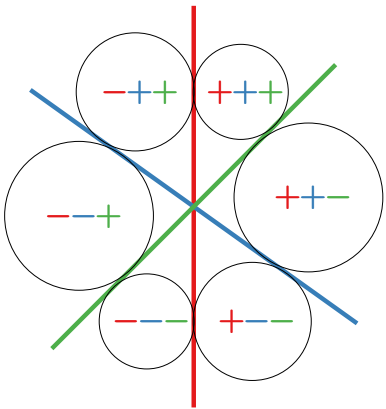
OpenImages Dataset
object recognition

$n \approx 9000$
$K = 50$

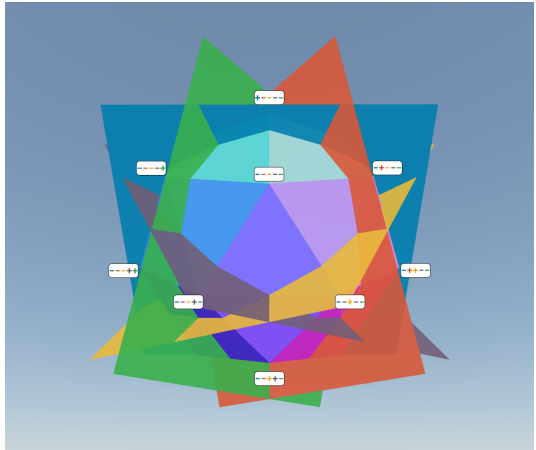


even sparse label configurations are unargmaxable

how do we know?

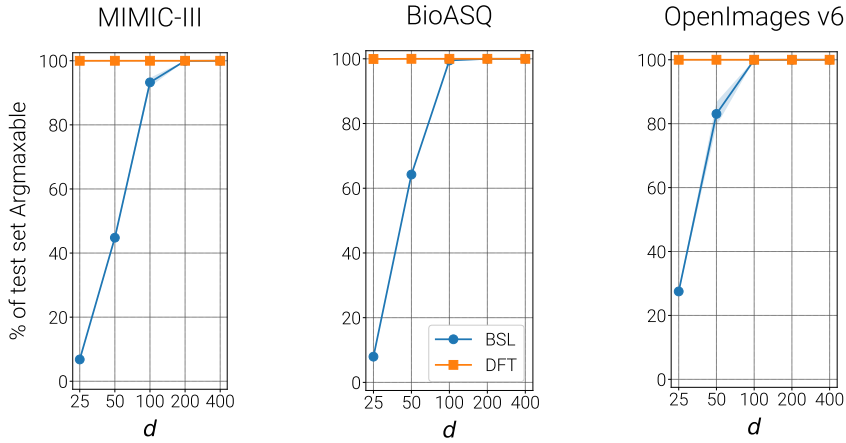


we provide a Chebyshev LP to *verify argmaxability*



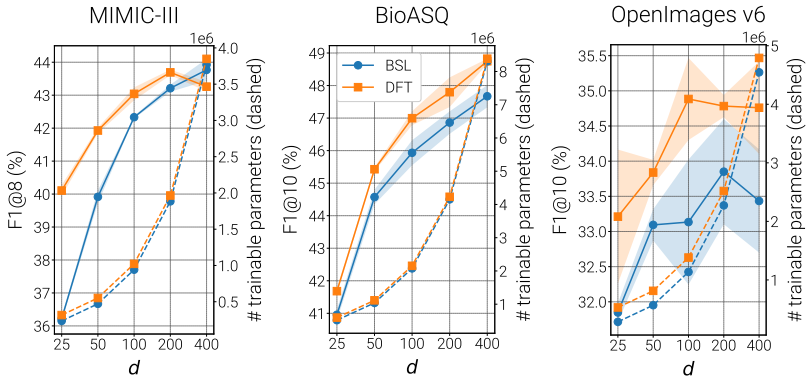
and a new differentiable layer to *guarantee argmaxability*

for K-active label configurations



and a new differentiable layer to guarantee argmaxability

based on the DFT



with comparable or better performance

open problems

I does SPL ensure argmaxability?

II scaling verifier to millions labels

III adversarial attacks