Probabilistic and Logistic Circuits: A New Synthesis of Logic and Machine Learning

Guy Van den Broeck

KULeuven Symposium
Dec 12, 2018

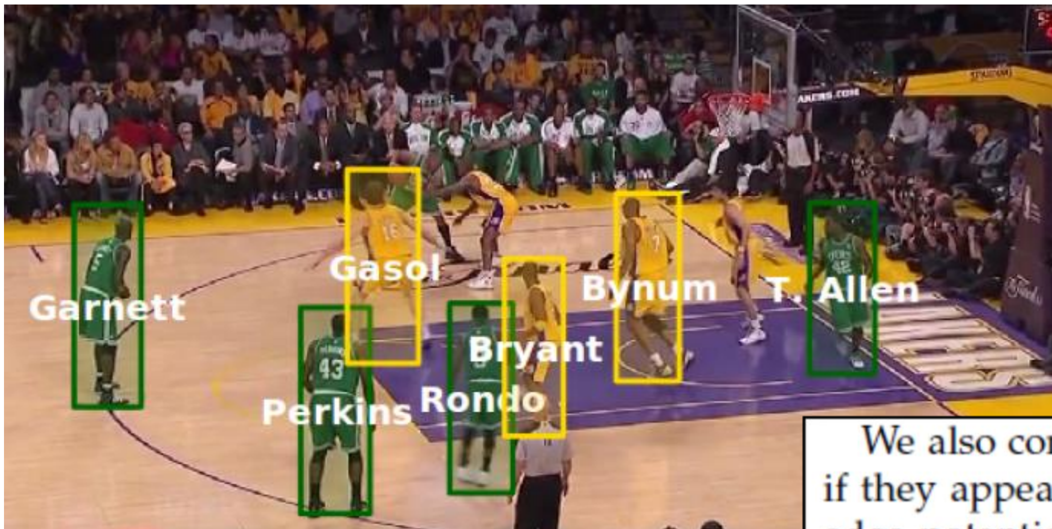
Outline

- Learning
  - Adding knowledge to deep learning
  - Logistic circuits for image classification
- Reasoning
  - Collapsed compilation
  - DIPPL: Imperative probabilistic programs

Outline

- Learning
  - **Adding knowledge to deep learning**
  - Logistic circuits for image classification
- Reasoning
  - Collapsed compilation
  - DIPPL: Imperative probabilistic programs

Motivation: Video



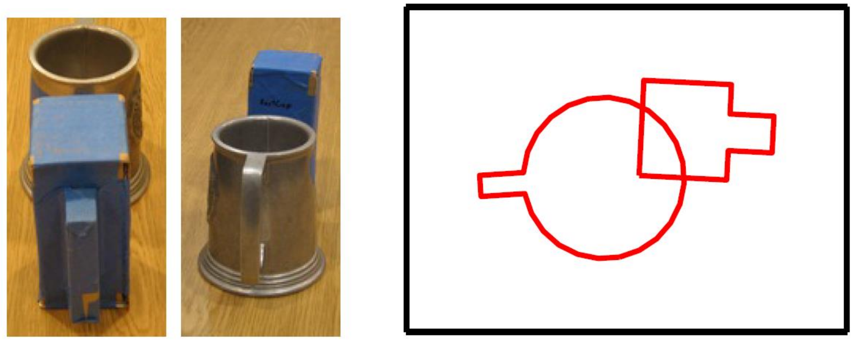
We also connect all pairs of identity nodes $y_{t,i}$ and $y_{t,j}$ if they appear in the same time t. We then introduce an edge potential that enforces mutual exclusion:

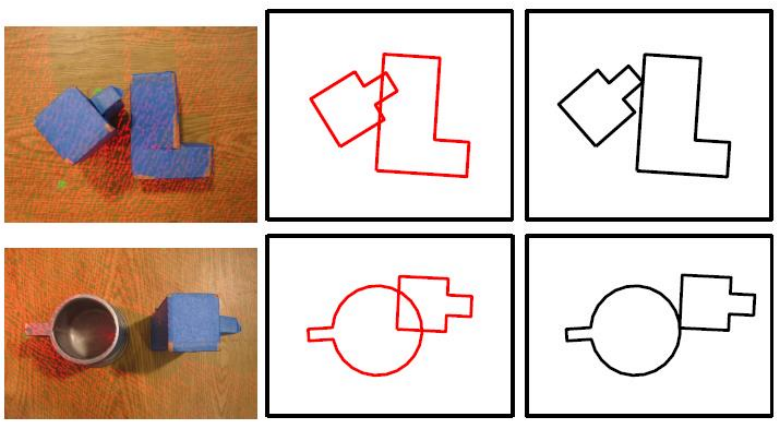
$$\psi_{\text{mutex}}(y_{t,i}, y_{t,j}) = \begin{cases} 1 & \text{if } y_{t,i} \neq y_{t,j} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

This potential specifies the constraint that a player can be appear only *once* in a frame. For example, if the i-th detection $y_{t,i}$ has been assign to Bryant, $y_{t,j}$ cannot have the same identity because Bryant is impossible to appear twice in a frame.

[Lu, W. L., Ting, J. A., Little, J. J., & Murphy, K. P. (2013). Learning to track and identify players from broadcast sports videos.]

Motivation: Robotics





The method developed in this paper can be used in a broad variety of semantic mapping and object manipulation tasks, providing an efficient and effective way to incorporate collision constraints into a recursive state estimator, obtaining optimal or near-optimal solutions.

[Wong, L. L., Kaelbling, L. P., & Lozano-Perez, T., Collision-free state estimation. ICRA 2012]

Motivation: Language

- Non-local dependencies:

  At least one verb in each sentence

- Sentence compression

  If a modifier is kept, its subject is also kept

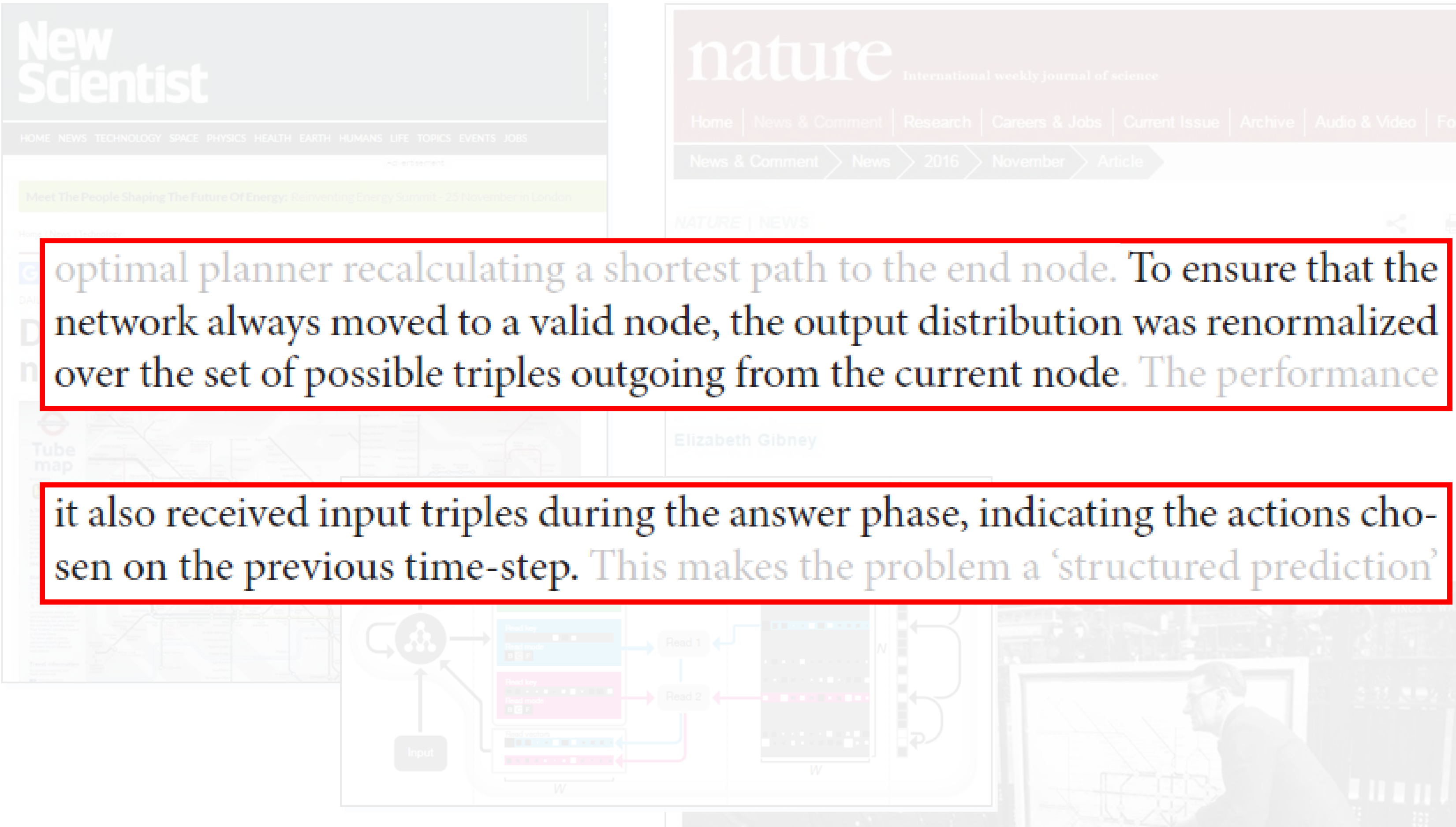
- Information extraction
- Semantic role labeling

… and many more!

Citations	
Start	The citation must start with author or editor.
AppearsOnce	Each field must be a consecutive list of words, and can appear at most once in a citation.
Punctuation	State transitions must occur on punctuation marks.
BookJournal	The words *proc, journal, proceedings, ACM* are *JOURNAL* or *BOOKTITLE*.
...	...
TechReport	The words *tech, technical* are *TECH_REPORT*.
Title	Quotations can appear only in titles.
Location	The words *CA, Australia, NY* are *LOCATION*.

[Chang, M., Ratinov, L., & Roth, D. (2008). Constraints as prior knowledge],…, [Chang, M. W., Ratinov, L., & Roth, D. (2012). Structured learning with constrained conditional models.], [https://en.wikipedia.org/wiki/Constrained_conditional_model]

Motivation: Deep Learning



[Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al.. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.]

Learning in Structured Spaces

L	K	P	A	Students
0	0	1	0	6
0	0	1	1	54
0	1	1	1	10
1	0	0	0	5
1	0	1	0	1
1	0	1	1	0
1	1	0	0	17
1	1	1	0	4
1	1	1	1	3

Data + **Constraints**

(Background Knowledge)
(Physics)

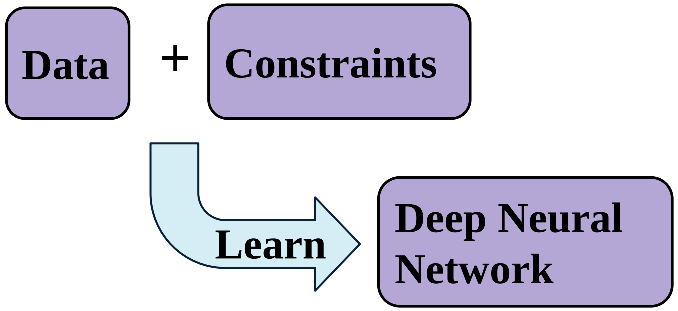
$$P \vee L$$
$$A \Rightarrow P$$
$$K \Rightarrow (P \vee L)$$

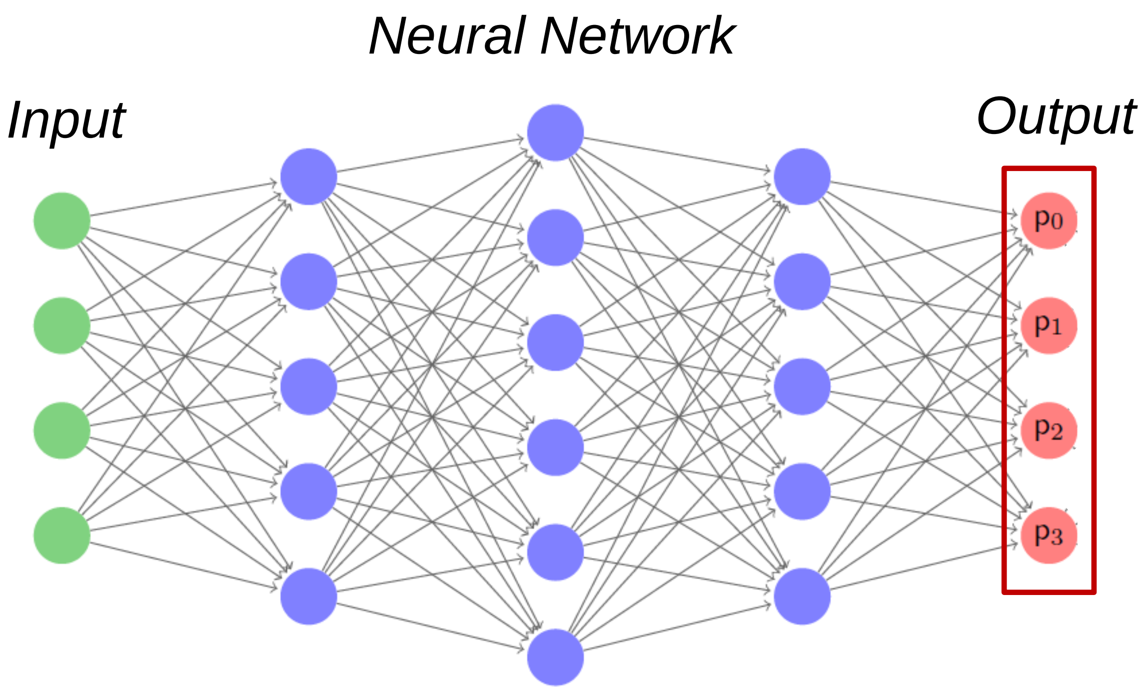
Learn

ML Model

Today's machine learning tools
don't take knowledge as input! ☹

Deep Learning with Logical Knowledge





Output is probability vector **p**, not Boolean logic!

Semantic Loss

Q: How close is output **p** *to satisfying constraint?*

Answer: Semantic loss function L(α,**p**)

- Axioms, for example:
  - If **p** is Boolean then L(**p**,**p**) = 0
  - If α implies β then L(α,**p**) ≥ L(β,**p**) (*α more strict*)
- Properties:
  - If α is equivalent to β then L(α,**p**) = L(β,**p**)
  - If **p** is Boolean and satisfies α then L(α,**p**) = 0

SEMANTIC Loss!

Semantic Loss: Definition

<u>Theorem</u>: Axioms imply unique semantic loss:

$$\mathrm{L}^{\mathrm{s}}(\alpha, \mathsf{p}) \propto -\log \underbrace{\sum_{\mathbf{x} \models \alpha} \underbrace{\prod_{i:\mathbf{x} \models X_i} \mathsf{p}_i \prod_{i:\mathbf{x} \models \neg X_i} (1 - \mathsf{p}_i)}_{\substack{\text{Probability of getting } \mathbf{x} \text{ after} \\ \text{flipping coins with prob. } \mathbf{p}}}}_{\substack{\text{Probability of satisfying } \alpha \text{ after} \\ \text{flipping coins with prob. } \mathbf{p}}}$$

Example: Exactly-One

- Data must have some label

  We agree this must be one of the 10 digits:

- Exactly-one constraint

  $\rightarrow$ For 3 classes: $\begin{cases} \boldsymbol{x_1} \lor \boldsymbol{x_2} \lor \boldsymbol{x_3} \\ \neg \boldsymbol{x_1} \lor \neg \boldsymbol{x_2} \\ \neg \boldsymbol{x_2} \lor \neg \boldsymbol{x_3} \\ \neg \boldsymbol{x_1} \lor \neg \boldsymbol{x_3} \end{cases}$

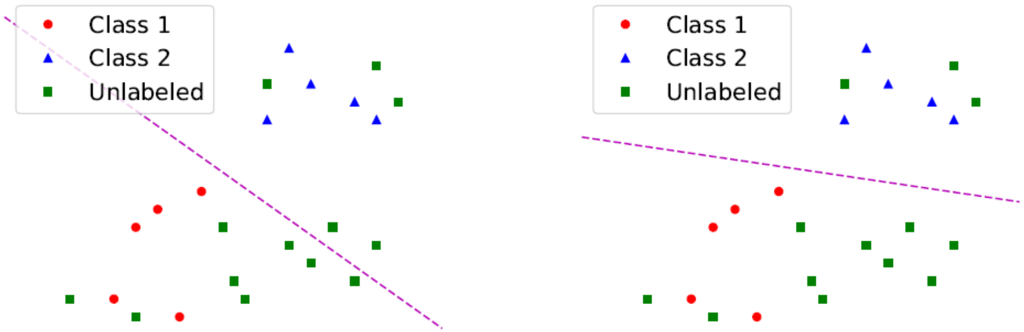
- Semantic loss:

$$\mathrm{L}^{\mathrm{s}}(\text{exactly-one}, \mathrm{p}) \propto -\log \underbrace{\sum_{i=1}^{n} \underbrace{\mathrm{p}_i \prod_{j=1, j \neq i}^{n} (1 - \mathrm{p}_j)}_{\text{Only } \boldsymbol{x_i} = \mathbf{1} \text{ after flipping coins}}}_{\text{Exactly one true } \boldsymbol{x} \text{ after flipping coins}}$$

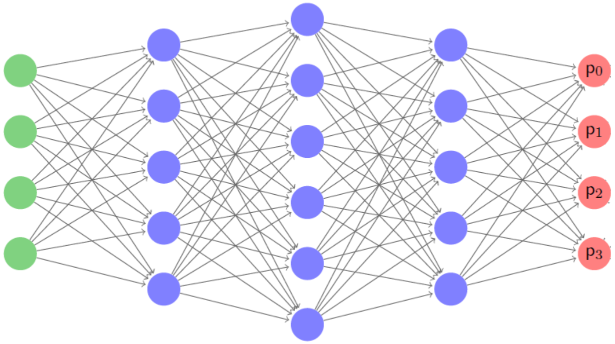
Semi-Supervised Learning

- Intuition: Unlabeled data must have some label

  Cf. entropy constraints, manifold learning



- Minimize exactly-one semantic loss on unlabeled data



Train with

$$existing\ loss + w \cdot semantic\ loss$$

MNIST Experiment

Accuracy % with # of used labels	100	1000	ALL
AtlasRBF (Pitelis et al., 2014)	91.9 (± 0.95)	96.32 (± 0.12)	98.69
Deep Generative (Kingma et al., 2014)	96.67(± 0.14)	97.60(± 0.02)	99.04
Virtual Adversarial (Miyato et al., 2016)	97.67	98.64	99.36
Ladder Net (Rasmus et al., 2015)	**98.94** (±0.37)	**99.16** (±0.08)	99.43 (± 0.02)
Baseline: MLP, Gaussian Noise	78.46 (±1.94)	94.26 (±0.31)	99.34 (±0.08)
Baseline: Self-Training	72.55 (±4.21)	87.43 (±3.07)	
MLP with Semantic Loss	98.38 (±0.51)	98.78 (±0.17)	99.36 (±0.02)

Competitive with state of the art in semi-supervised deep learning

FASHION Experiment

(a) Confidently Correct

(b) Unconfidently Correct

(c) Unconfidently Incorrect

(d) Confidently Incorrect

Accuracy % with # of used labels	100	500	1000	ALL
Ladder Net (Rasmus et al., 2015)	81.46 (±0.64)	85.18 (±0.27)	86.48 (± 0.15)	90.46
Baseline: MLP, Gaussian Noise	69.45 (±2.03)	78.12 (±1.41)	80.94 (±0.84)	89.87
MLP with Semantic Loss	**86.74** (±0.71)	**89.49** (±0.24)	89.67 (±0.09)	89.81

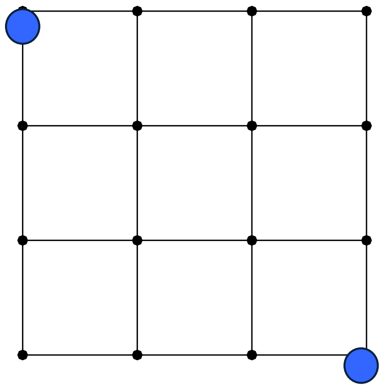
Outperforms Ladder Nets!

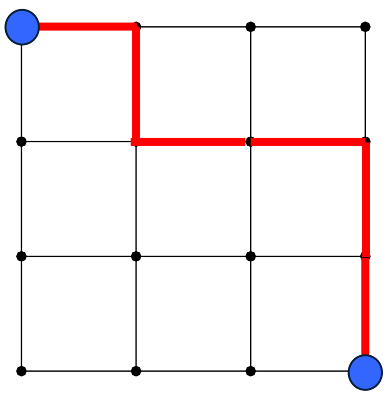
Same conclusion on CIFAR10

Accuracy % with # of used labels	4000	ALL
CNN Baseline in Ladder Net	76.67 (± 0.61)	90.73
Ladder Net (Rasmus et al., 2015)	79.60 (±0.47)	
Baseline: CNN, Whitening, Cropping	77.13	90.96
CNN with Semantic Loss	**81.79**	90.92

What about real constraints? Paths

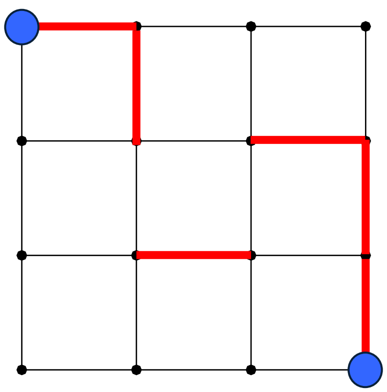
cf. Nature paper





Good variable assignment (represents route)

184



Bad variable assignment (does not represent route)

16,777,032

Unstructured probability space: 184+16,777,032 = 2^{24}

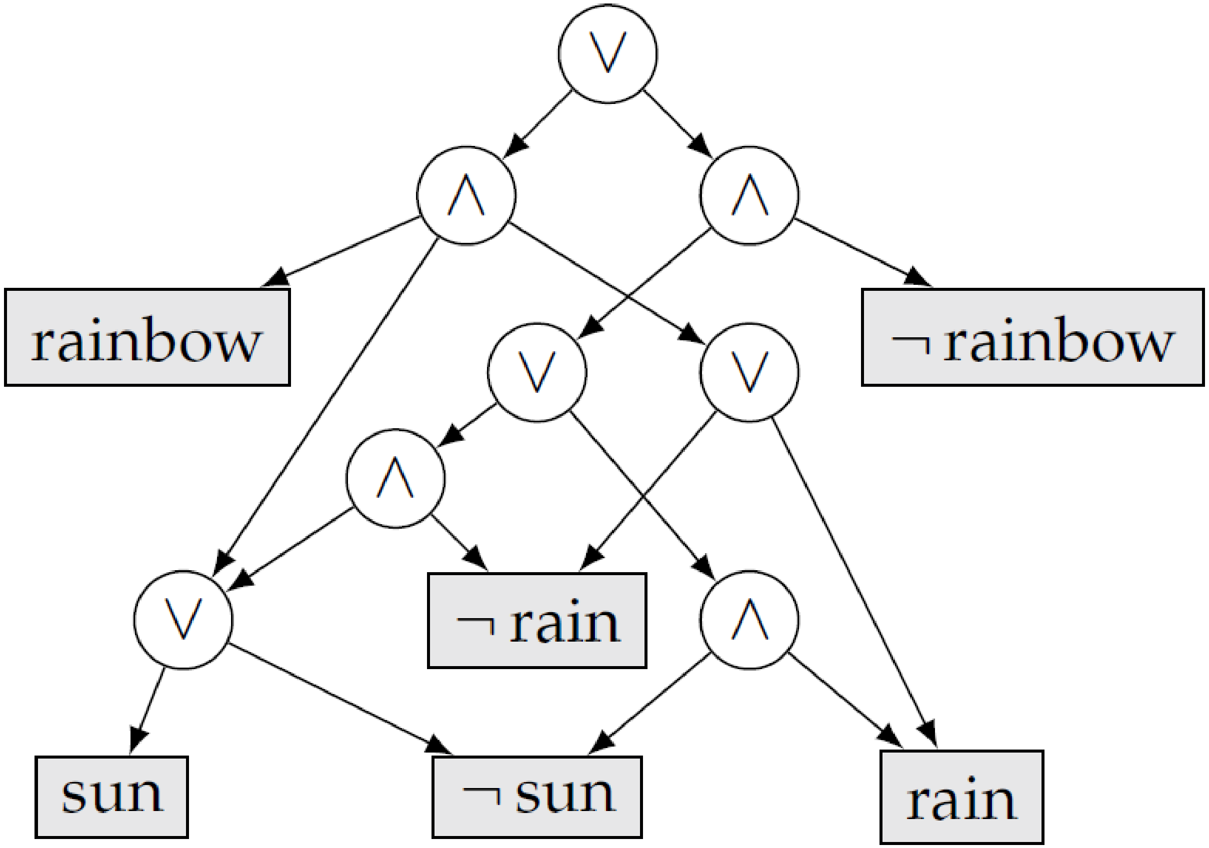
Space easily encoded in logical constraints ☺ [Nishino et al.]

How to Compute Semantic Loss?

- In general: #P-hard ☹

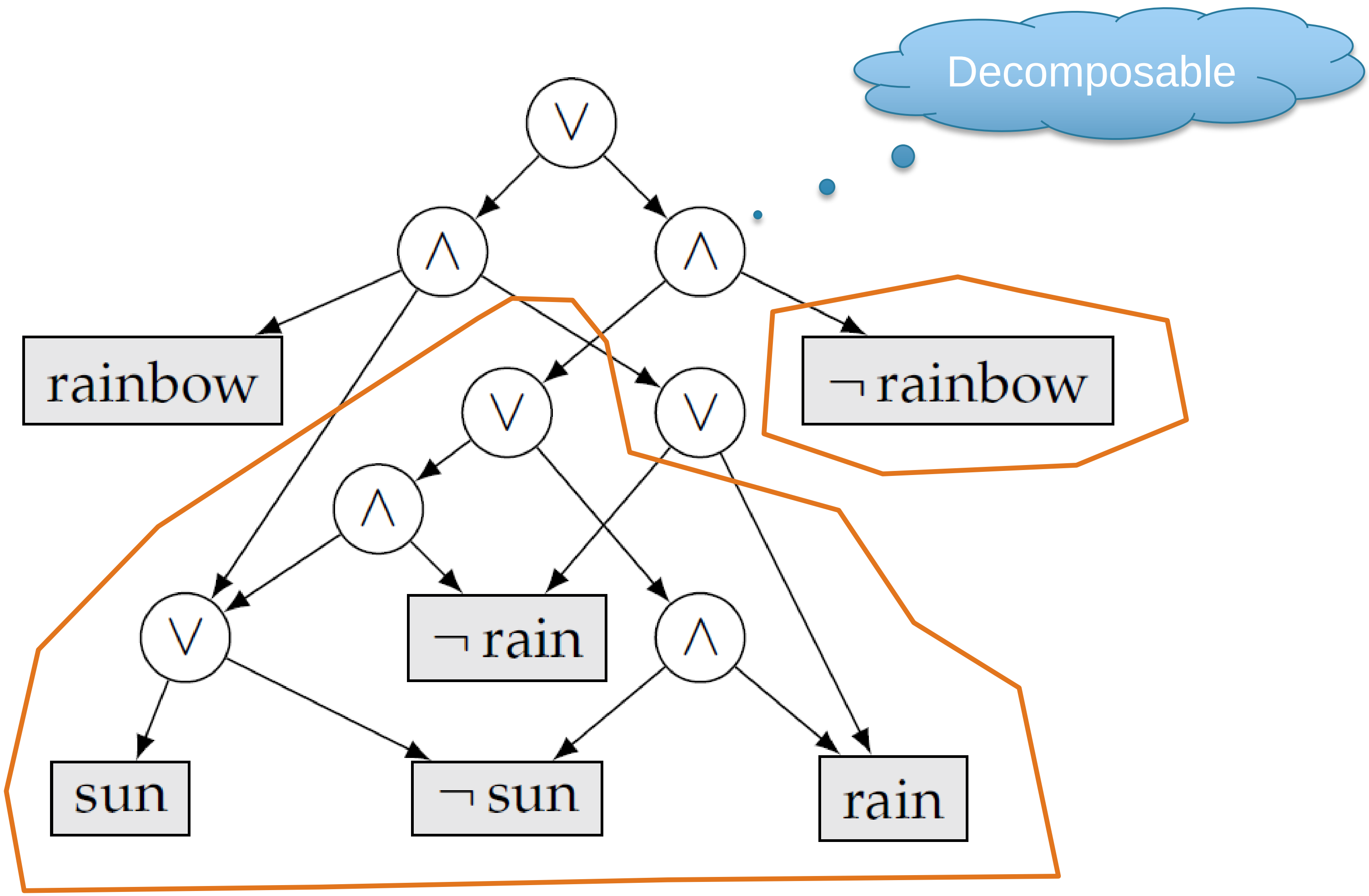
Negation Normal Form Circuits

Δ = (sun ∧ rain ⇒ rainbow)



[Darwiche 2002]

Decomposable Circuits

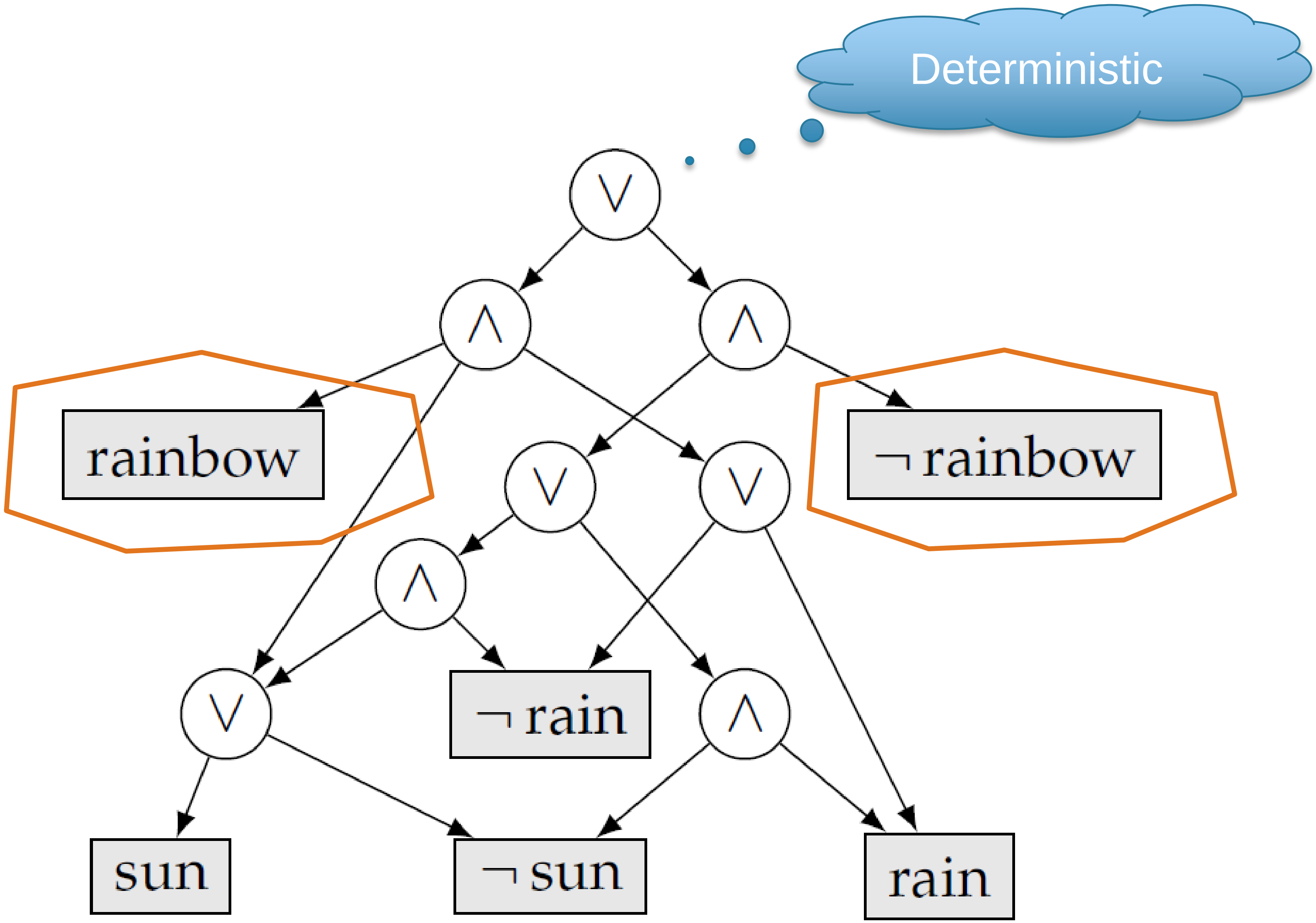


[Darwiche 2002]

Tractable for Logical Inference

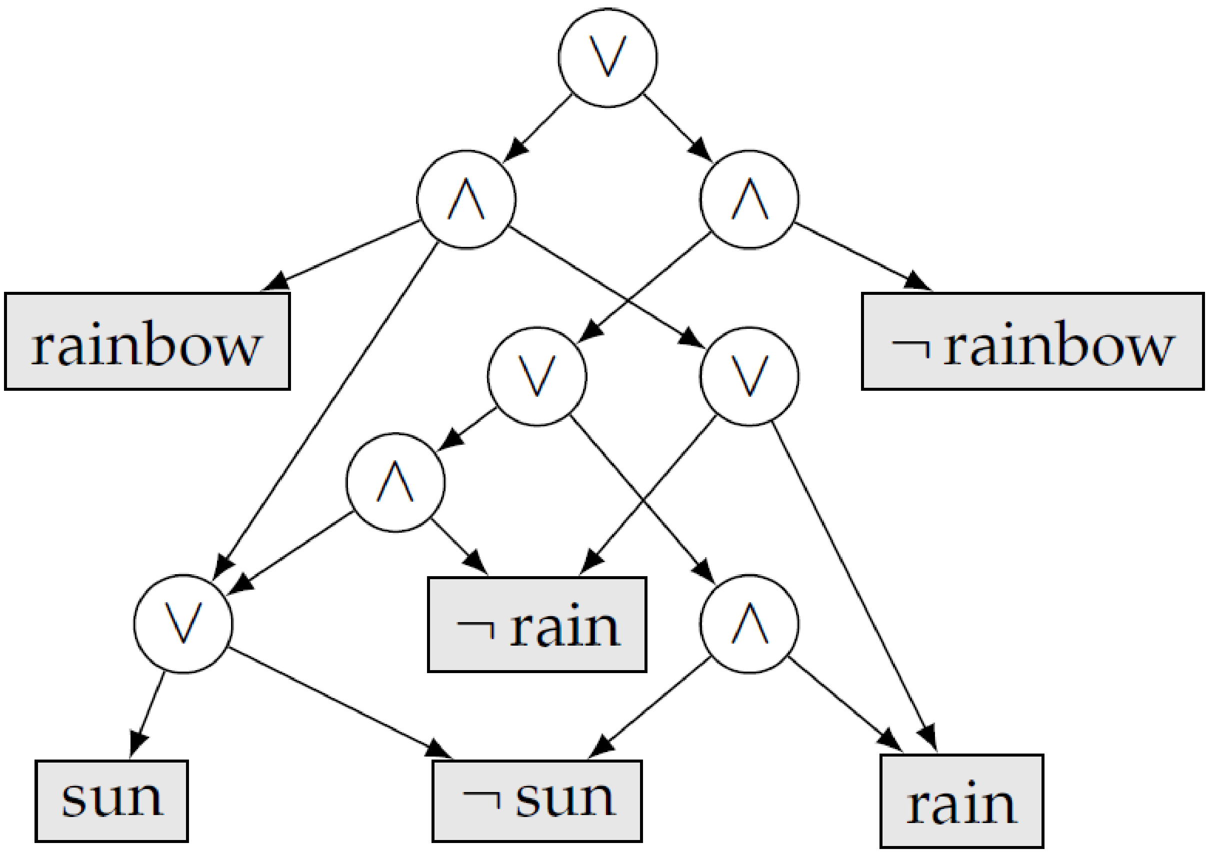
- Is there a solution? (SAT) ✓
  - SAT($\alpha \lor \beta$) iff SAT(α) or SAT(β) (*always*)
  - SAT($\alpha \land \beta$) iff SAT(α) and SAT(β) (*decomposable*)
- How many solutions are there? (#SAT)
- Complexity linear in circuit size ☺

Deterministic Circuits

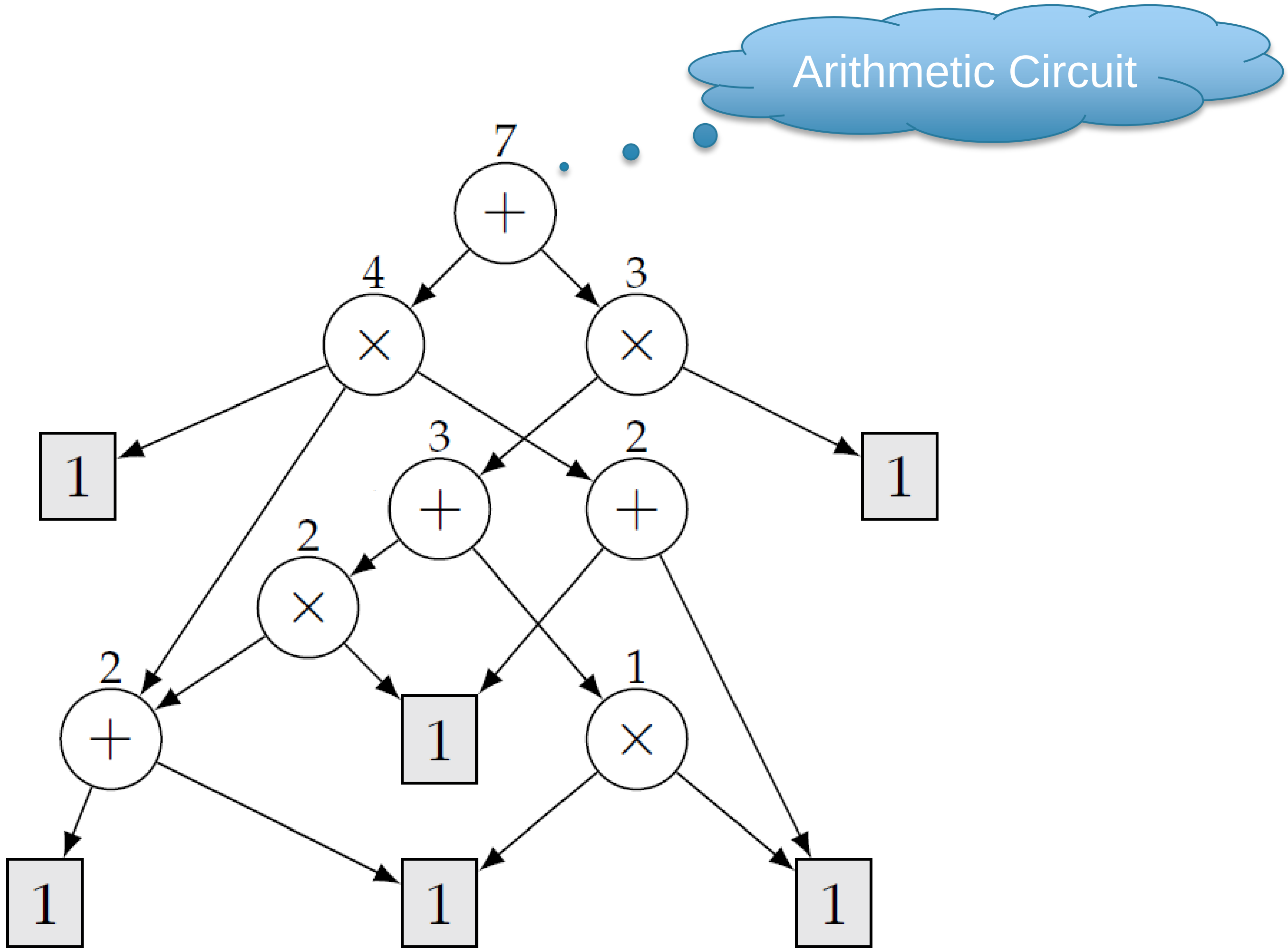


[Darwiche 2002]

How many solutions are there? (#SAT)



How many solutions are there? (#SAT)

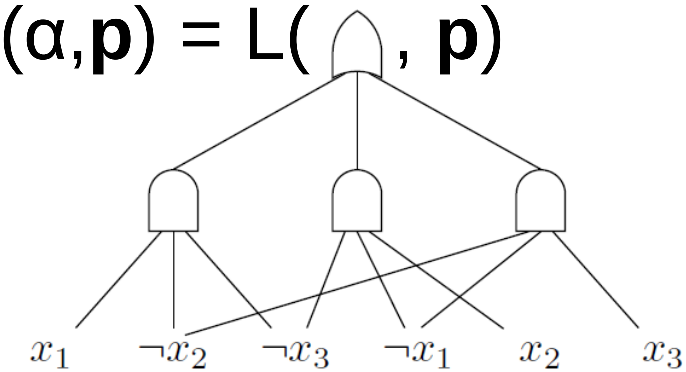
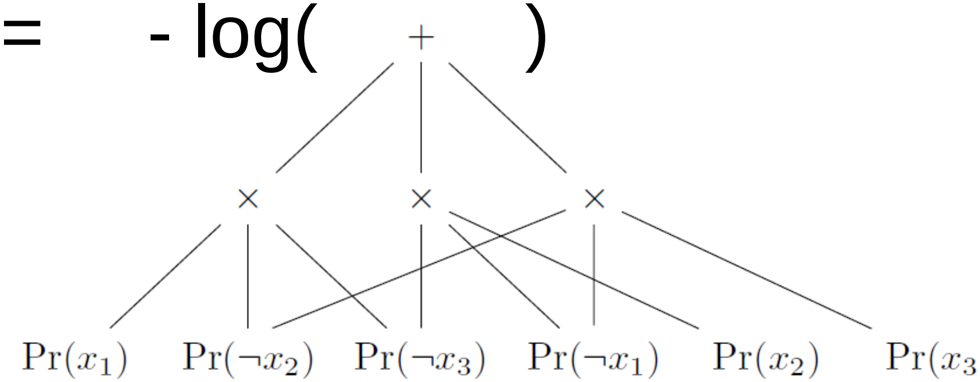


Tractable for Logical Inference

- Is there a solution? (SAT) ✓
- How many solutions are there? (#SAT) ✓
- Stricter languages (e.g., BDD, SDD):
  - Equivalence checking ✓
  - Conjoin/disjoint/negate circuits ✓
- Complexity linear in circuit size ☺
- Compilation into circuit language by either
  - ↓ exhaustive SAT solver
  - ↑ conjoin/disjoin/negate

How to Compute Semantic Loss?

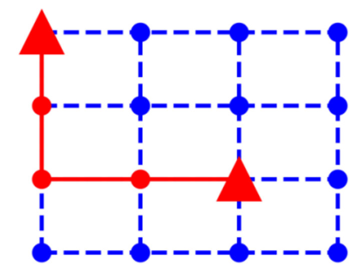
- In general: #P-hard ☹
- With a logical circuit for α: Linear!
- Example: exactly-one constraint:

$L(\alpha,\mathbf{p}) = L($  $, \mathbf{p}) \quad = \quad -\log($  $)$

- *Why?* Decomposability and determinism!

Predict Shortest Paths

Add semantic loss
for path constraint



Test accuracy %	Coherent	Incoherent	Constraint
5-layer MLP	5.62	**85.91**	6.99
Semantic loss	**28.51**	83.14	**69.89**

Is prediction the shortest path?
This is the real task!

Are individual edge predictions correct?

Is output a path?

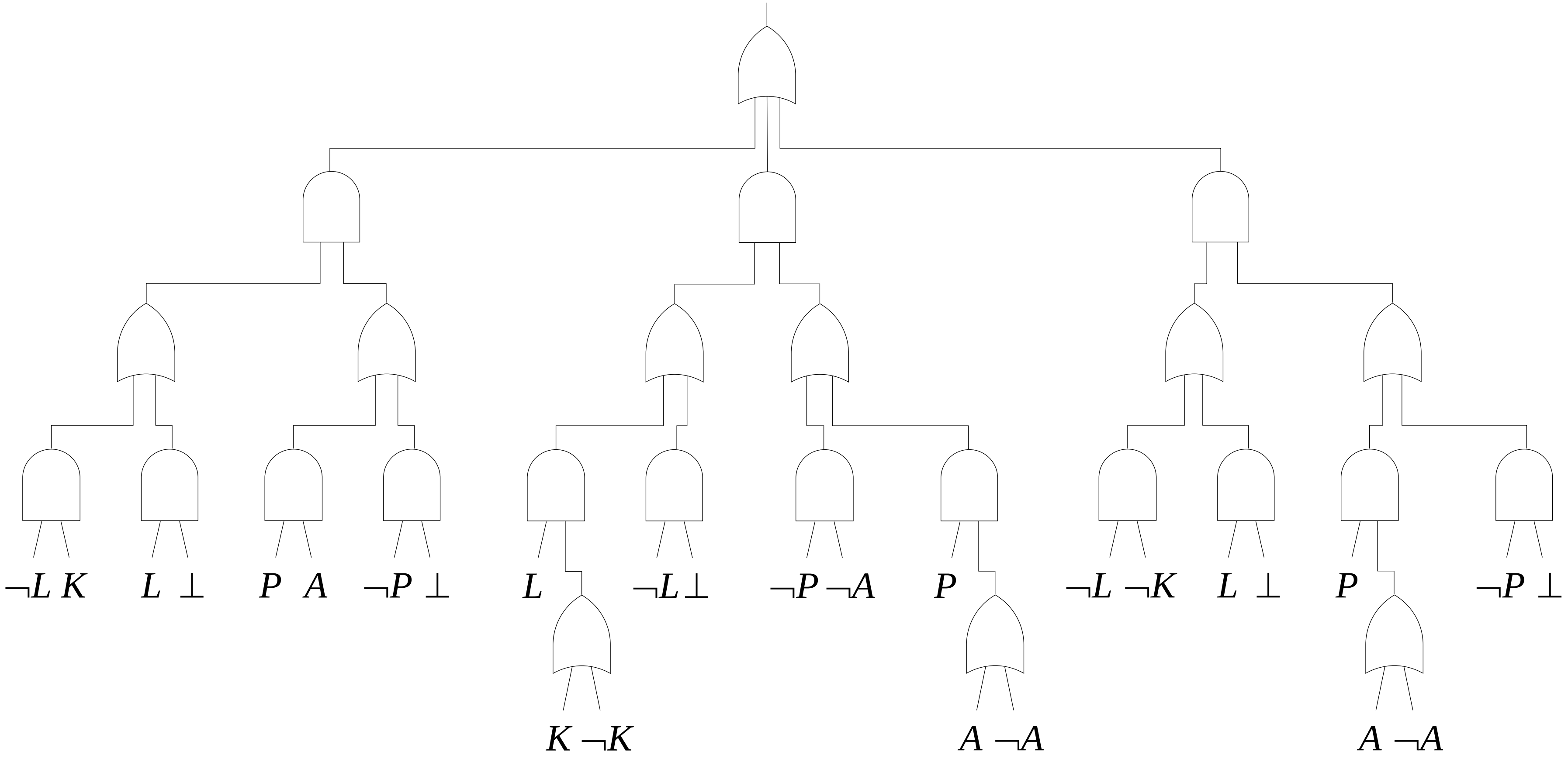
(same conclusion for predicting sushi preferences, see paper)

Outline

- Learning
  - Adding knowledge to deep learning
  - **Logistic circuits for image classification**
- Reasoning
  - Collapsed compilation
  - DIPPL: Imperative probabilistic programs

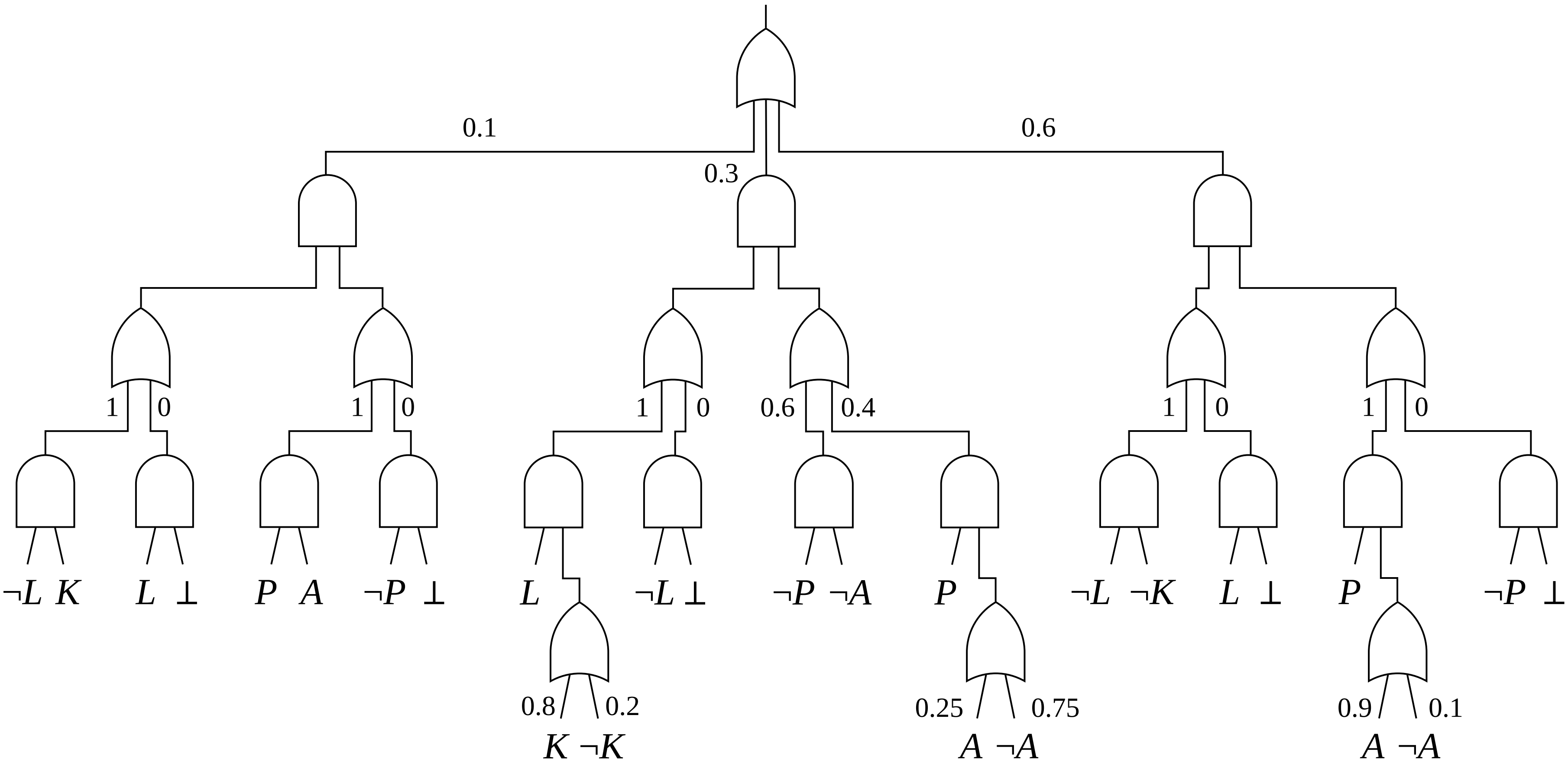
Logical Circuits

$P \vee L$
$A \Rightarrow P$
$K \Rightarrow (P \vee L)$



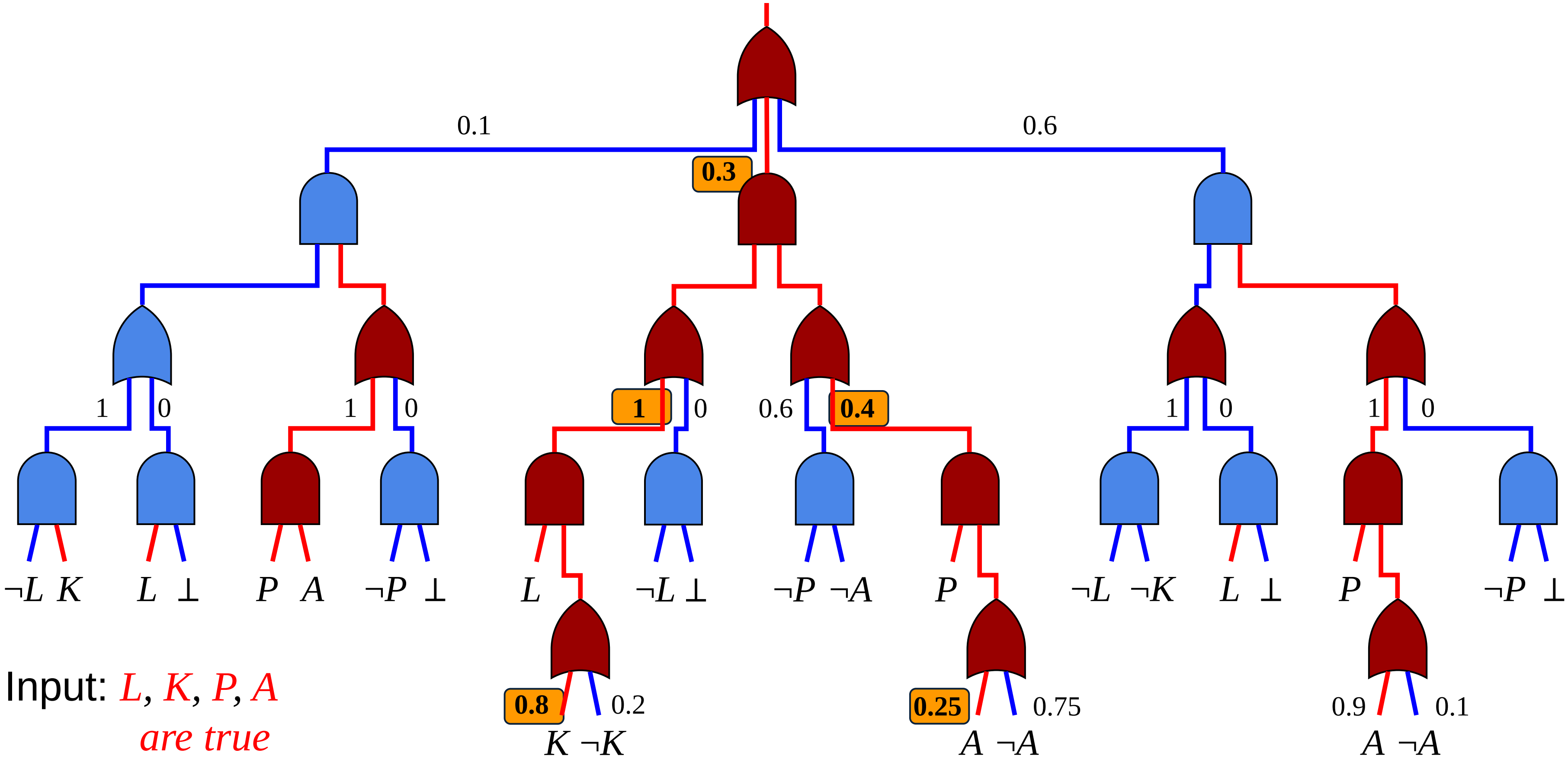
Can we represent a **distribution** over the solutions to the constraint?

Probabilistic Circuits



Syntax: assign a normalized probability to each OR gate input

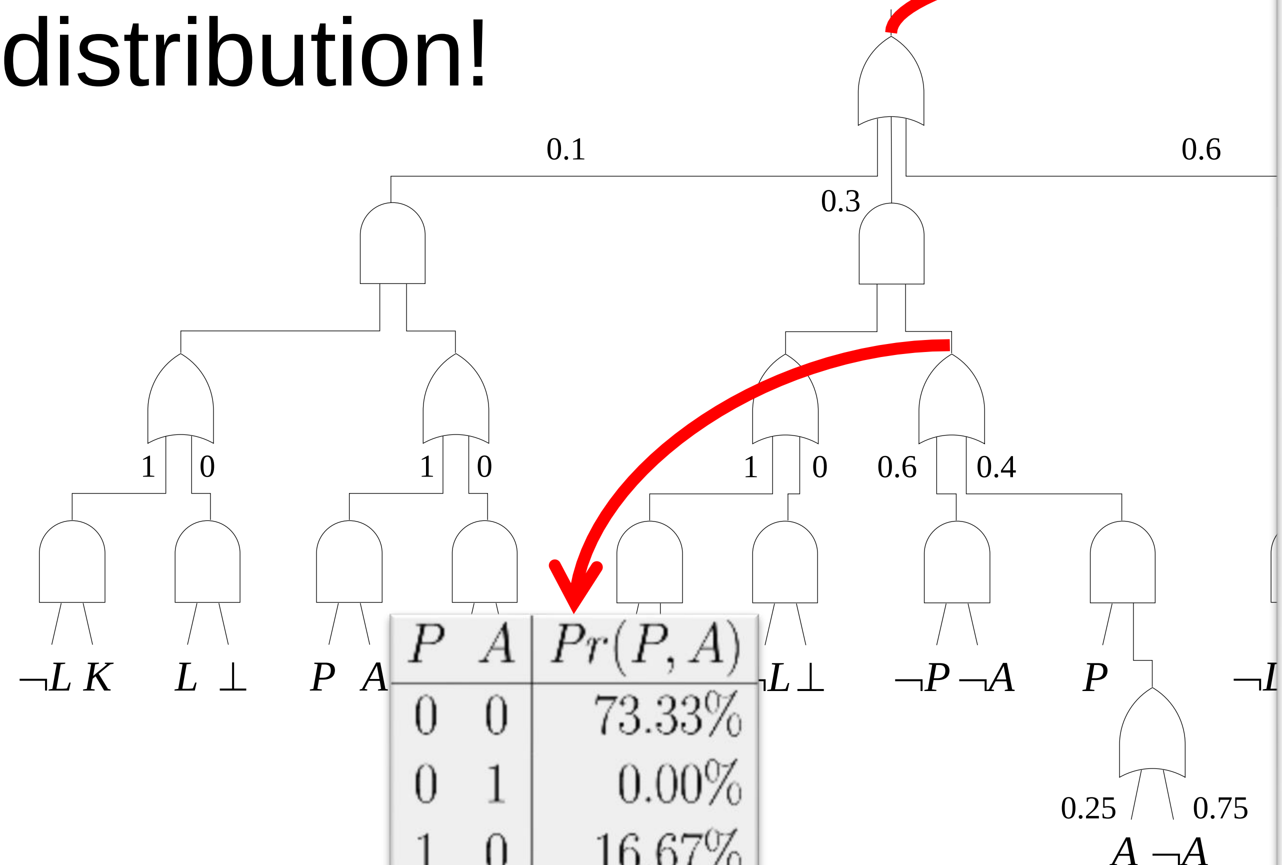
PSDD: Probabilistic SDD



Input: *L*, *K*, *P*, *A* *are true*

Pr(*L*,*K*,*P*,*A*) = 0.3 x 1 x 0.8 x 0.4 x 0.25 = **0.024**

Each node represents a normalized distribution!



P	A	$Pr(P, A)$
0	0	73.33%
0	1	0.00%
1	0	16.67%
1	1	10.00%

L	K	P	A	$Pr(L, K, P, A)$
0	0	0	0	0.00%
0	0	0	1	0.00%
0	0	1	0	6.00%
0	0	1	1	54.00%
0	1	0	0	0.00%
0	1	0	1	0.00%
0	1	1	0	0.00%
0	1	1	1	10.00%
1	0	0	0	4.40%
1	0	0	1	0.00%
1	0	1	0	1.00%
1	0	1	1	0.60%
1	1	0	0	17.6%
1	1	0	1	0.00%
1	1	1	0	4.00%
1	1	1	1	2.40%

Can read probabilistic independences off the circuit structure!

Can interpret every parameter as a conditional probability! (XAI)

Tractable for Probabilistic Inference

- **MAP inference**:
  Find most-likely assignment to x given y
  (otherwise NP-hard)
- Computing **conditional probabilities** Pr(x|y)
  (otherwise #P-hard)
- **Sample** from Pr(x|y)
- Algorithms linear in circuit size ☺
  (pass up, pass down, similar to backprop)

Parameter Learning Algorithms

- Closed form
  max likelihood
  from complete data

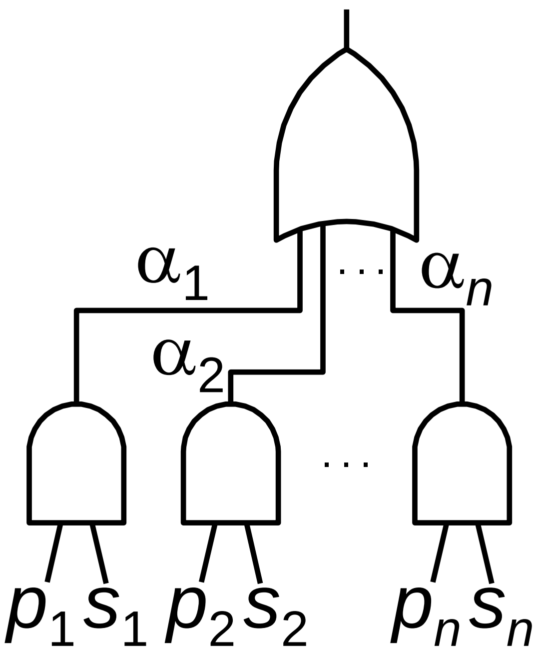
L	K	P	A	Students
0	0	1	0	6
0	0	1	1	54
0	1	1	1	10
1	0	0	0	5
1	0	1	0	1
1	0	1	1	0
1	1	0	0	17
1	1	1	0	4
1	1	1	1	3

- One pass over data to estimate Pr(x|y)

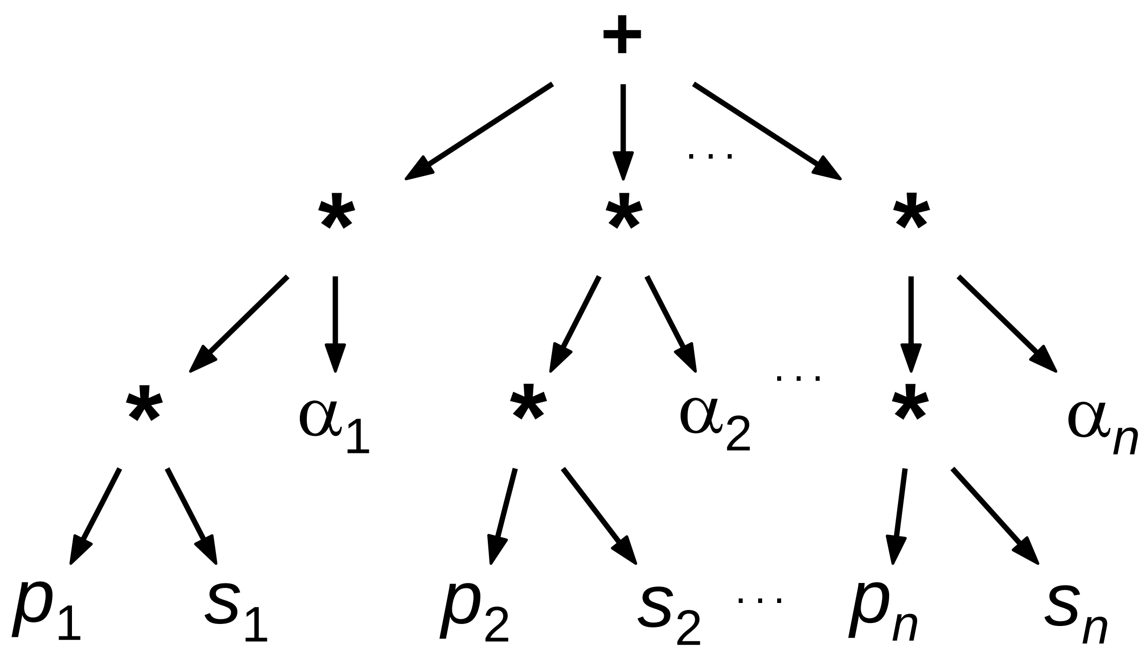
Not a lot to say: very easy! ☺

PSDDs

…are Sum-Product Networks
…are Arithmetic Circuits



PSDD



AC

Learn Mixtures of PSDD Structures

Datasets	|Var|	LearnPSDD Ensemble	Best-to-Date
NLTCS	16	$-5.99^{\dagger}$	-6.00
MSNBC	17	$-6.04^{\dagger}$	$-6.04^{\dagger}$
KDD	64	$-2.11^{\dagger}$	-2.12
Plants	69	-13.02	$-11.99^{\dagger}$
Audio	100	-39.94	$-39.49^{\dagger}$
Jester	100	-51.29	$-41.11^{\dagger}$
Netflix	100	$-55.71^{\dagger}$	-55.84
Accidents	111	-30.16	$-24.87^{\dagger}$
Retail	135	$-10.72^{\dagger}$	-10.78
Pumsb-Star	163	-26.12	$-22.40^{\dagger}$
DNA	180	-88.01	$-80.03^{\dagger}$
Kosarek	190	$-10.52^{\dagger}$	-10.54
MSWeb	294	-9.89	$-9.22^{\dagger}$
Book	500	-34.97	$-30.18^{\dagger}$
EachMovie	500	-58.01	$-51.14^{\dagger}$
WebKB	839	-161.09	$-150.10^{\dagger}$
Reuters-52	889	-89.61	$-80.66^{\dagger}$
20NewsGrp.	910	-155.97	$-150.88^{\dagger}$
BBC	1058	-253.19	$-233.26^{\dagger}$
AD	1556	-31.78	$-14.36^{\dagger}$

State of the art on 6 datasets!

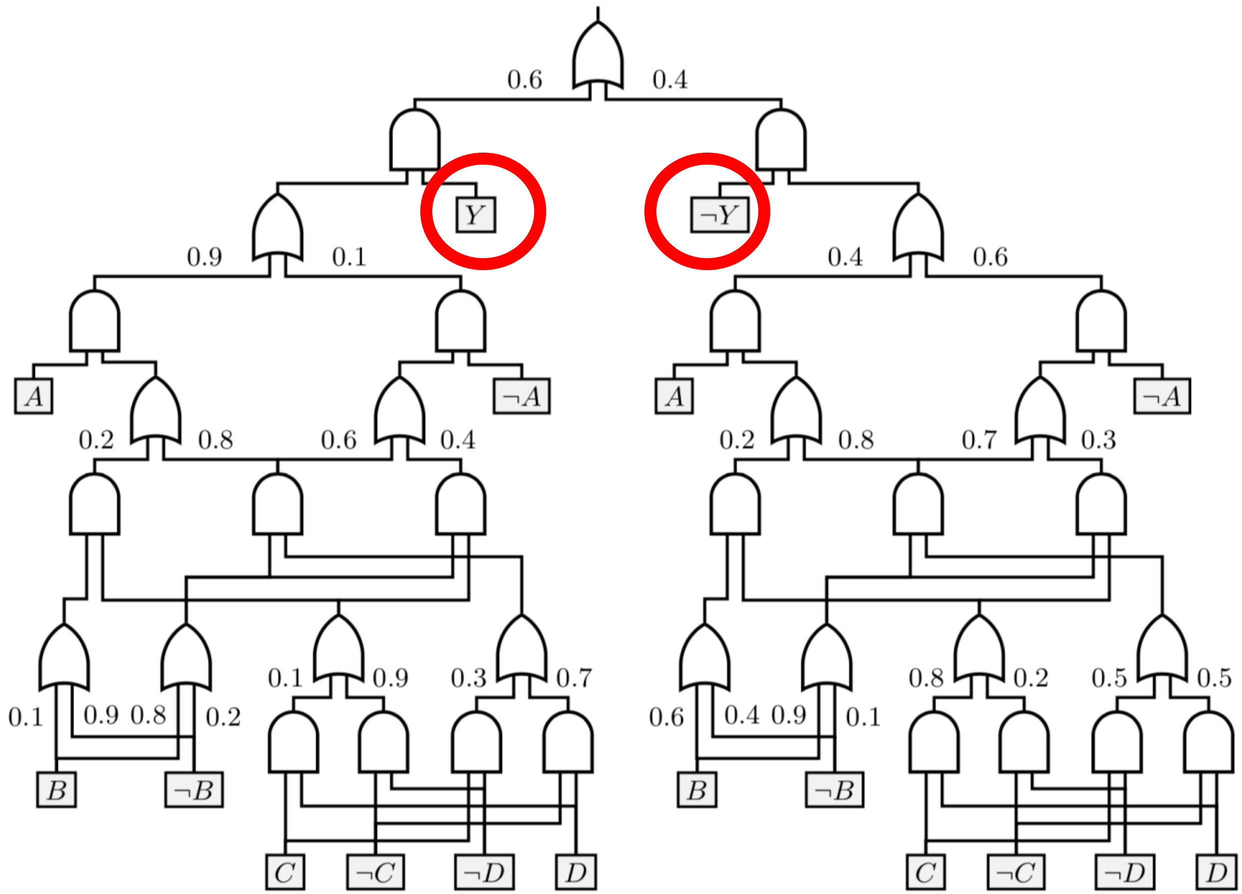
Q: “Help! I need to learn a discrete probability distribution…”
A: Learn mixture of PSDDs!

Strongly outperforms
- Bayesian network learners
- Markov network learners

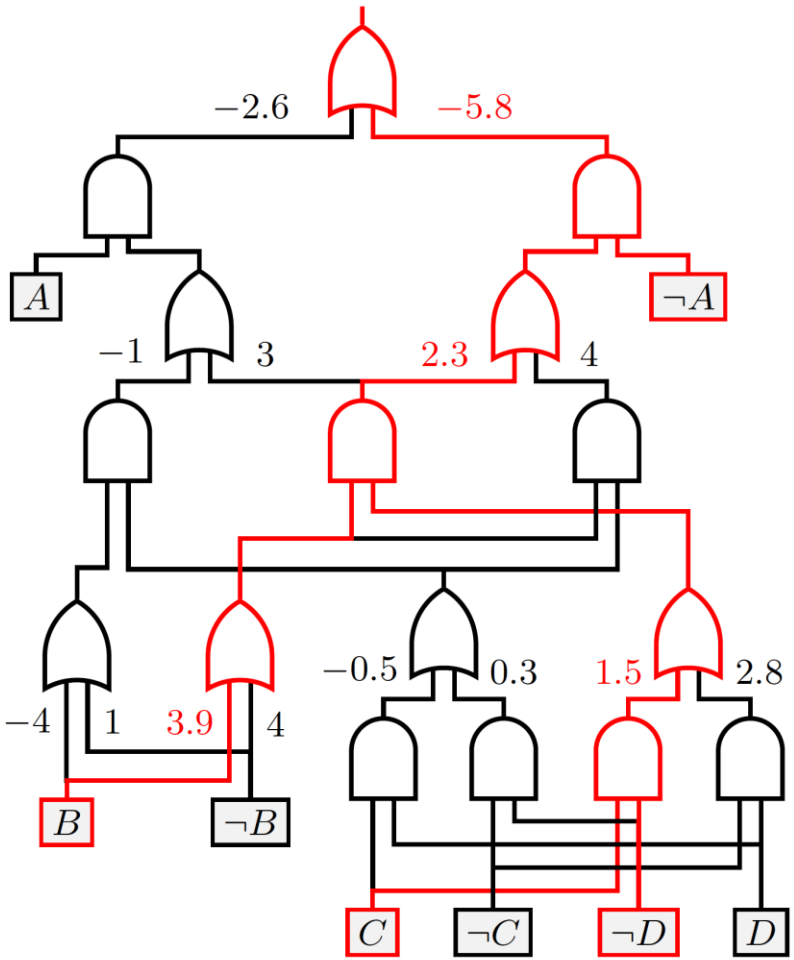
Competitive with
- SPN learners
- Cutset network learners

What if I only want to classify Y?

$\Pr(Y, A, B, C, D)$



Logistic Circuits



Represents $\Pr(Y \mid A, B, C, D)$

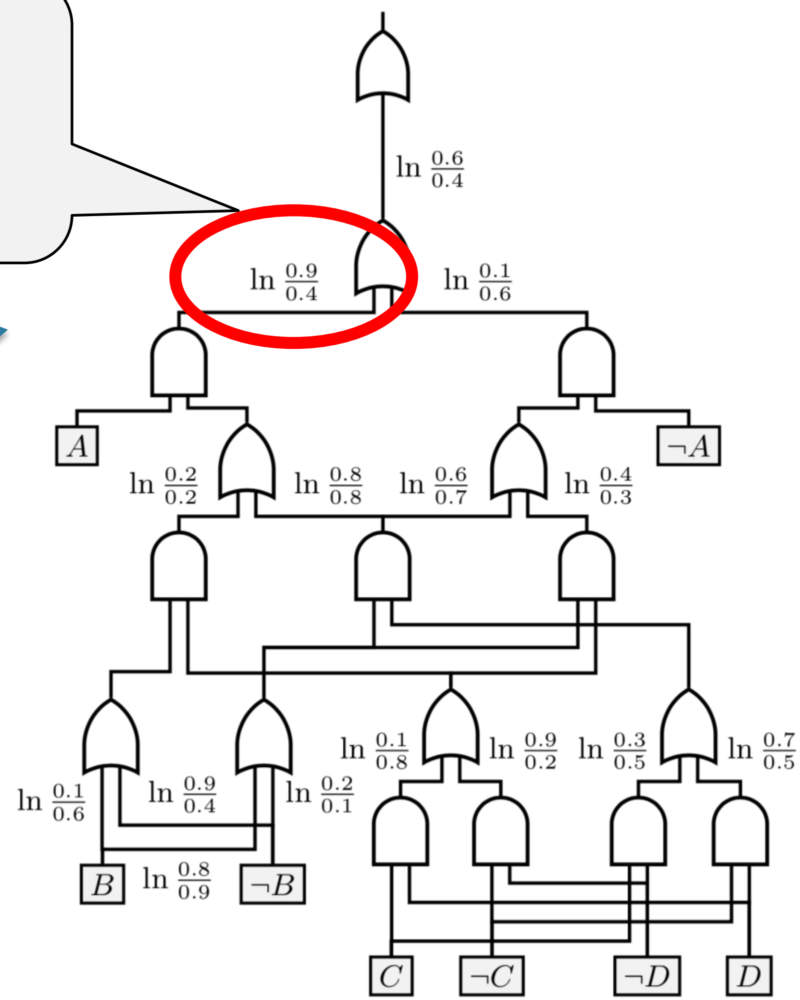
- Take all 'hot' wires
- Sum their weights
- Push through logistic function

A	B	C	D	$g_r(ABCD)$	$\Pr(Y=1 \mid ABCD)$
1	0	1	1	-3.1	4.31%
0	1	1	0	1.9	86.99%
1	1	1	0	5.8	99.70%

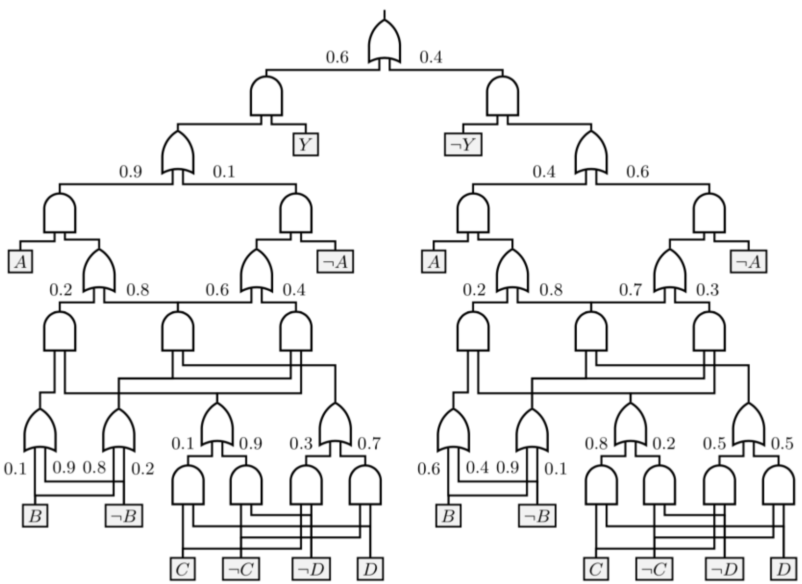
Logistic vs. Probabilistic Circuits

Probabilities
become
log-odds

$\Pr(Y \mid A, B, C, D)$



$\Pr(Y, A, B, C, D)$



Parameter Learning

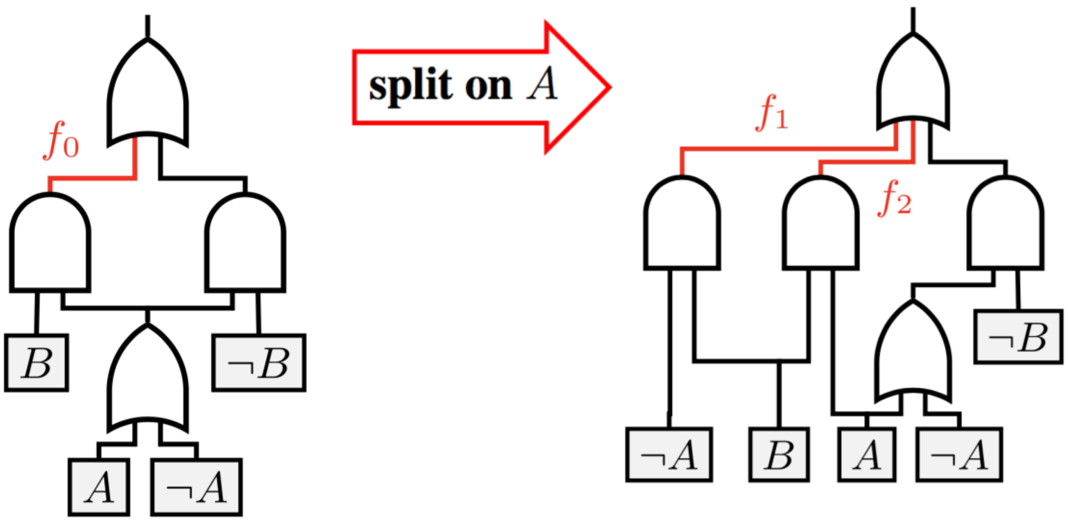
Reduce to logistic regression:

$$\Pr(Y = 1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x} \cdot \boldsymbol{\theta})}$$

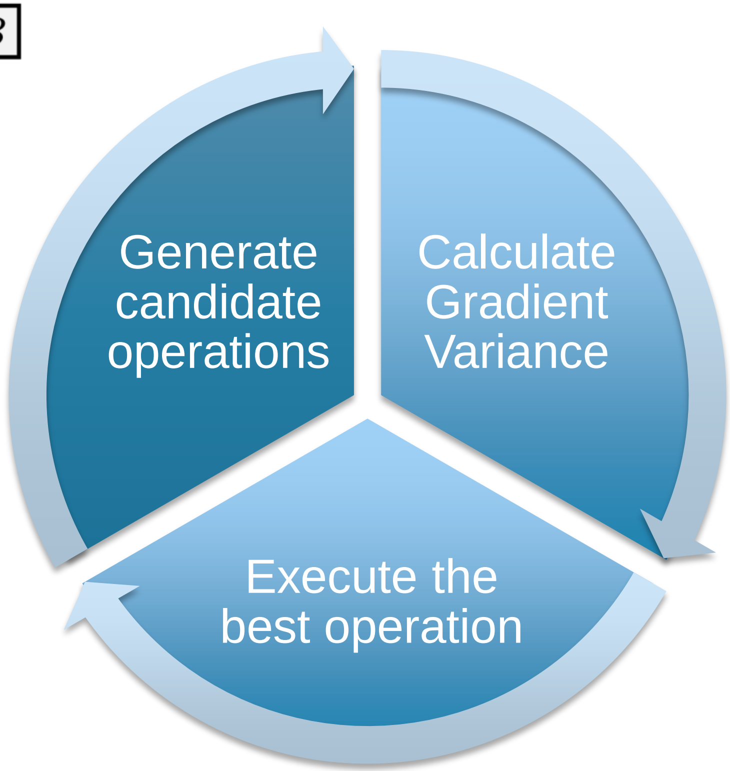
Features associated with each wire
"Global Circuit Flow" features

Learning parameters θ is convex optimization!

Logistic Circuit Structure Learning



Similar to LearnPSDD structure learning



Comparable Accuracy with Neural Nets

Accuracy % on Dataset	Mnist	Fashion
Baseline: Logistic Regression	85.3	79.3
Baseline: Kernel Logistic Regression	97.7	88.3
Random Forest	97.3	81.6
3-layer MLP	97.5	84.8
RAT-SPN (Peharz et al. 2018)	98.1	89.5
SVM with RBF Kernel	98.5	87.8
5-Layer MLP	99.3	89.8
Logistic Circuit (binary)	97.4	87.6
Logistic Circuit (real-valued)	99.4	91.3
CNN with 3 conv layers	99.1	90.7
Resnet (He et al. 2016)	99.5	93.6

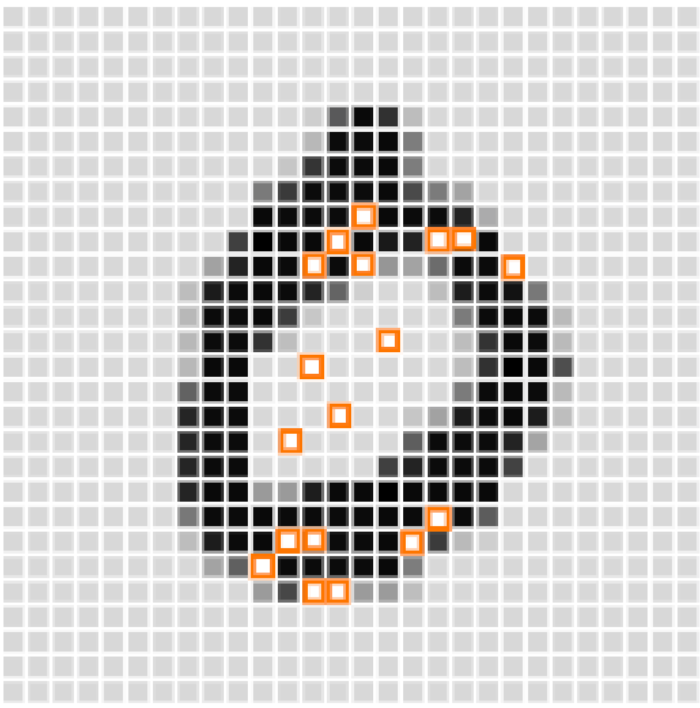
Significantly Smaller in Size

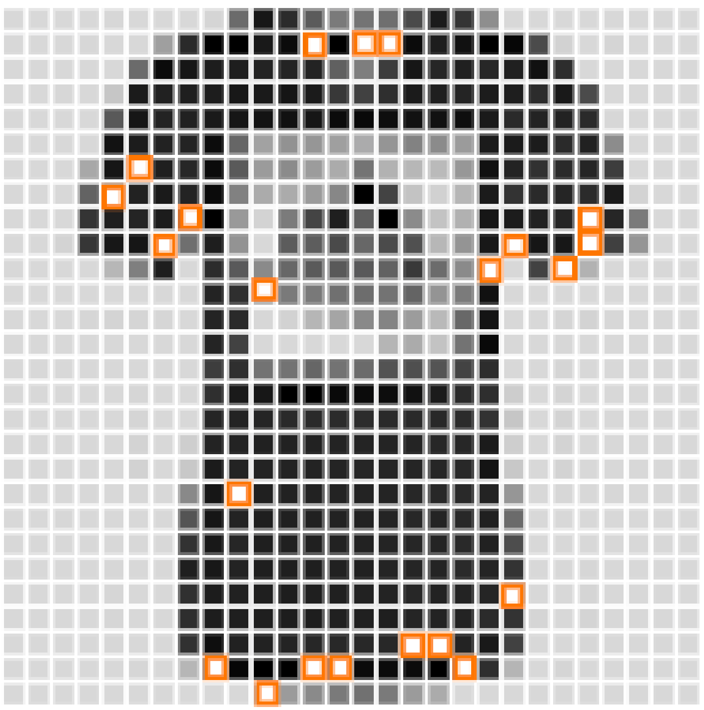
Number of Parameters	Mnist	Fashion
Baseline: Logistic Regression	<1K	<1K
Baseline: Kernel Logistic Regression	1,521 K	3,930K
Logistic Circuit (real-valued)	182K	467K
Logistic Circuit (binary)	268K	614K
3-layer MLP	1,411K	1,411K
RAT-SPN (Peharz et al. 2018)	8,500K	650K
CNN with 3 conv layers	2,196K	2,196K
5-layer MLP	2,411K	2,411K
Resnet (He et al. 2016)	4,838K	4,838K

Better Data Efficiency

Accuracy % with % of Training Data	MNIST			Fashion		
	100%	10%	2%	100%	10%	2%
5-layer MLP	99.3	**98.2**	94.3	89.8	86.5	80.9
CNN with 3 Conv Layers	99.1	98.1	95.3	90.7	87.6	83.8
Logistic Circuit (Binary)	97.4	96.9	94.1	87.6	86.7	83.2
Logistic Circuit (Real-Valued)	**99.4**	97.6	**96.1**	**91.3**	**87.8**	**86.0**

Interpretable?

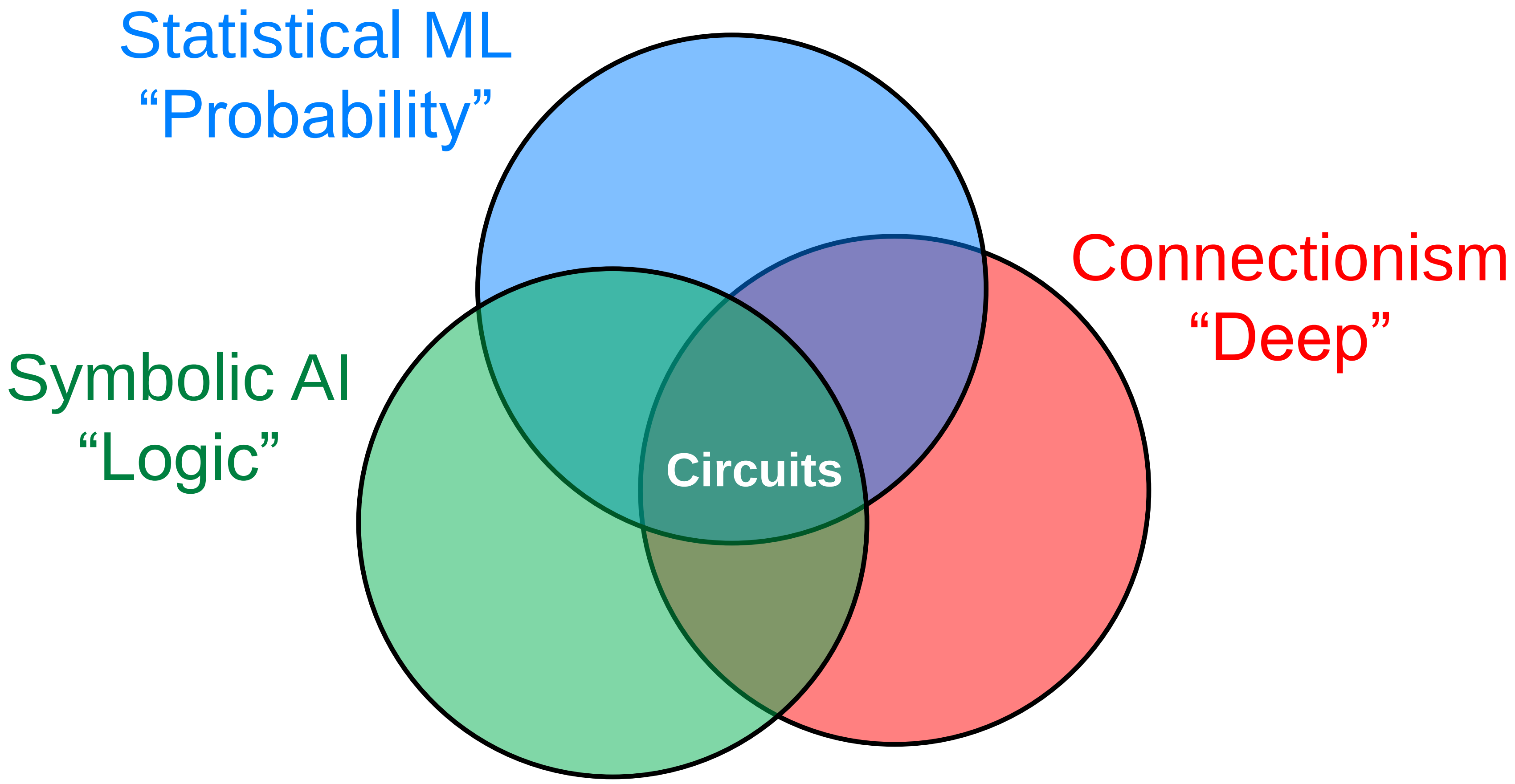




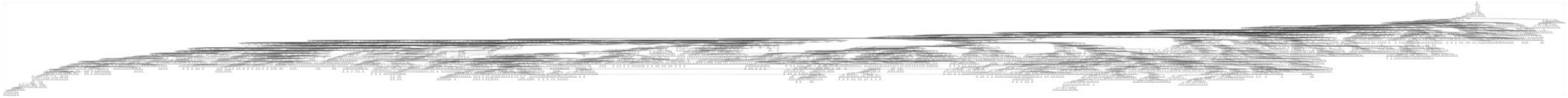
Outline

- Learning
  - Adding knowledge to deep learning
  - Logistic circuits for image classification
- Reasoning
  - **Collapsed compilation**
  - **DIPPL: Imperative probabilistic programs**

Conclusions



Questions?



PSDD with 15,000 nodes

References

- Doga Kisa, Guy Van den Broeck, Arthur Choi and Adnan Darwiche. **Probabilistic sentential decision diagrams**, *In Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2014.
- Arthur Choi, Guy Van den Broeck and Adnan Darwiche. **Tractable Learning for Structured Probability Spaces: A Case Study in Learning Preference Distributions**, *In Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.
- Arthur Choi, Guy Van den Broeck and Adnan Darwiche. **Probability Distributions over Structured Spaces**, *In Proceedings of the AAAI Spring Symposium on KRR*, 2015.
- Jessa Bekker, Jesse Davis, Arthur Choi, Adnan Darwiche and Guy Van den Broeck. **Tractable Learning for Complex Probability Queries**, *In Advances in Neural Information Processing Systems 28 (NIPS)*, 2015
- Yitao Liang, Jessa Bekker and Guy Van den Broeck. **Learning the Structure of Probabilistic Sentential Decision Diagrams**, *In Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

.

References

- Yitao Liang and Guy Van den Broeck. **Towards Compact Interpretable Models: Shrinking of Learned Probabilistic Sentential Decision Diagrams**, *In IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI)*, 2017.
- Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang and Guy Van den Broeck. **A Semantic Loss Function for Deep Learning with Symbolic Knowledge**, *In Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.
- Tal Friedman and Guy Van den Broeck. **Approximate Knowledge Compilation by Online Collapsed Importance Sampling**, *In Advances in Neural Information Processing Systems 31 (NIPS)*, 2018.
- Yitao Liang and Guy Van den Broeck. **Learning Logistic Circuits**, *In Proceedings of the 33rd Conference on Artificial Intelligence (AAAI)*, 2019.