UCLA Computer Science

STAR AI RESEARCH LAB UCLA

Towards a New Synthesis of Reasoning and Learning

Guy Van den Broeck

WUSTL CSE, Jan 23, 2020

The AI Dilemma

The AI Dilemma

Pure Logic **Pure Learning**

- Slow thinking: deliberative, cognitive, model-based, extrapolation
- Amazing achievements until this day
- “*Pure logic is brittle*”
  noise, uncertainty, incomplete knowledge, …

The AI Dilemma

Pure Logic **Pure Learning**

- Fast thinking: instinctive, perceptive, model-free, interpolation
- Amazing achievements recently
- “*Pure learning is brittle*”

  bias, algorithmic fairness, interpretability, explainability, adversarial attacks, unknown unknowns, calibration, verification, missing features, missing labels, data efficiency, shift in distribution, general robustness and safety

  fails to incorporate a sensible model of the world

The **FALSE** AI Dilemma

So all hope is lost?

Probabilistic World Models

- Joint distribution P(X)
- Wealth of representations: can be causal, relational, etc.
- Knowledge + data
- Reasoning + learning

Pure Logic Probabilistic World Models Pure Learning

High-Level Probabilistic Representations Reasoning, and Learning

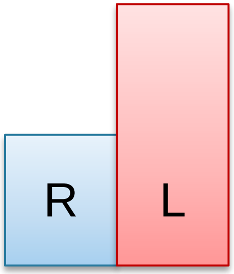
Pure Logic Probabilistic World Models Pure Learning

A New Synthesis of Learning and Reasoning

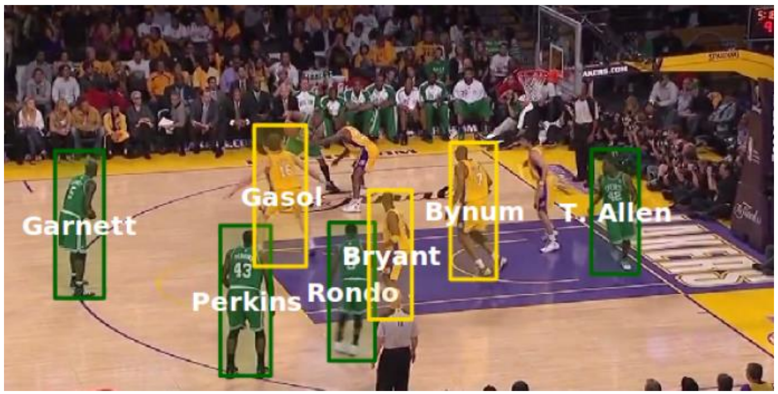
Outline: Reasoning ∩ Learning

1. Deep Learning with Symbolic Knowledge
2. Efficient Reasoning During Learning
3. Probabilistic and Logistic Circuits

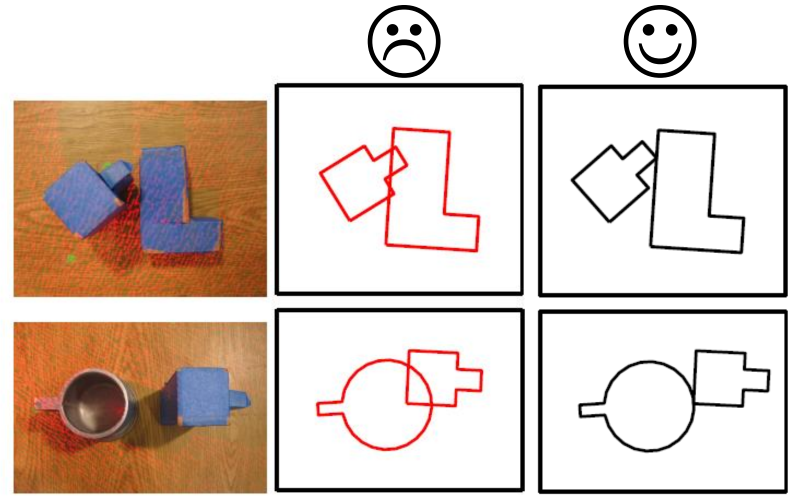
Deep Learning with Symbolic Knowledge



Motivation: Vision, Robotics, NLP



People appear at most once in a frame

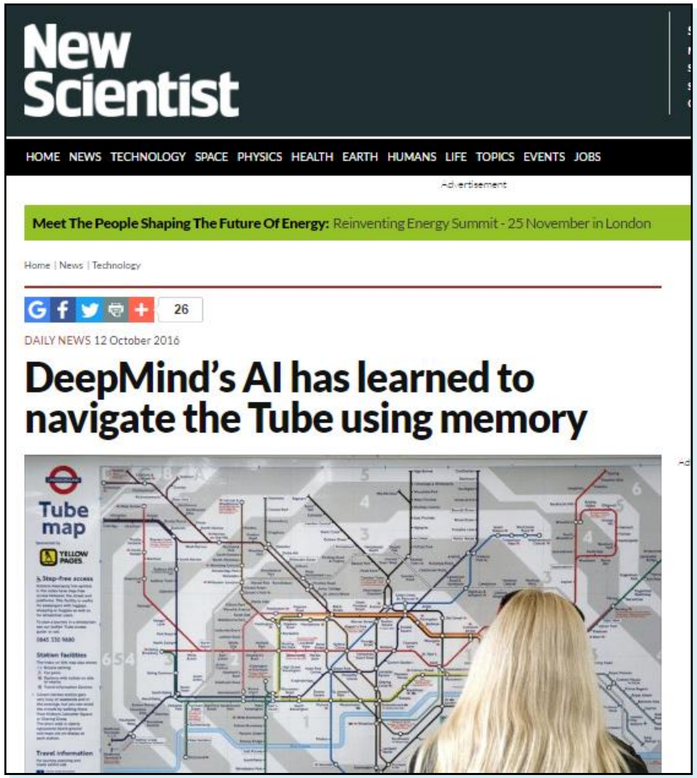


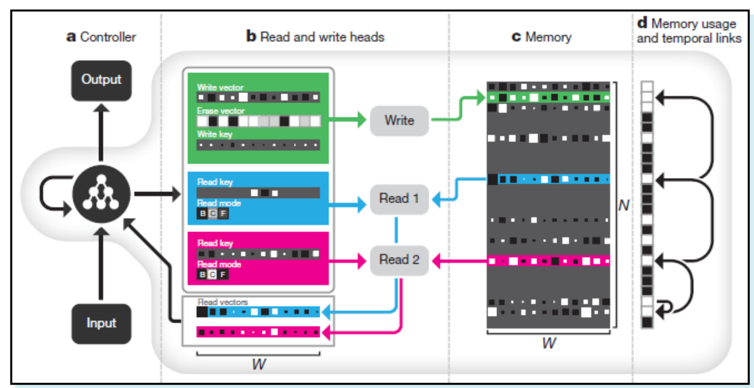
Rigid objects don't overlap

At least one verb in each sentence.
If X and Y are married, then they are people.

[Lu, W. L., Ting, J. A., Little, J. J., & Murphy, K. P. (2013). Learning to track and identify players from broadcast sports videos.], [Wong, L. L., Kaelbling, L. P., & Lozano-Perez, T., Collision-free state estimation. ICRA 2012], [Chang, M., Ratinov, L., & Roth, D. (2008). Constraints as prior knowledge], [Ganchev, K., Gillenwater, J., & Taskar, B. (2010). Posterior regularization for structured latent variable models]… and many many more!

Motivation: Deep Learning



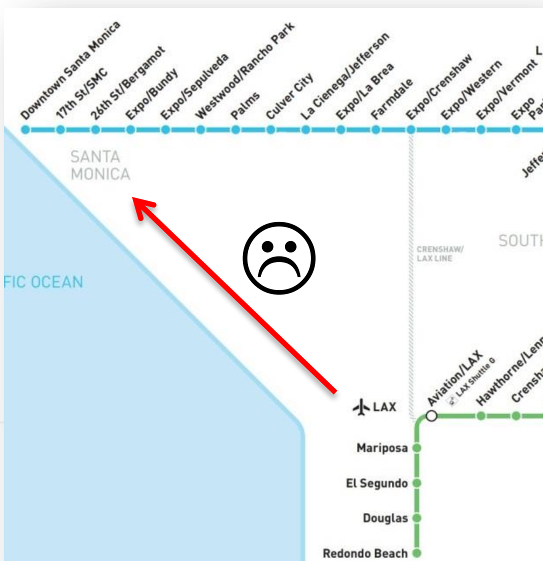


[Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al.. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.]

Motivation: Deep Learning

DeepMind's latest technique uses external memory to **solve tasks that require logic and reasoning — a step toward more human-like AI.**

… but …



optimal planner recalculating a shortest path to the end node. To ensure that the network always moved to a valid node, the output distribution was renormalized over the set of possible triples outgoing from the current node. The performance

it also received input triples during the answer phase, indicating the actions chosen on the previous time-step. This makes the problem a 'structured prediction'

[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.]

Knowledge vs. Data

- Where did the world knowledge go?
  – Python scripts
    - Decode/encode cleverly
    - Fix inconsistent beliefs
  – Rule-based decision systems
  – Dataset design
  – "a big hack" (with author's permission)
- In some sense we went backwards

  Less principled, scientific, and intellectually satisfying ways of incorporating knowledge

Learning with Symbolic Knowledge

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)

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

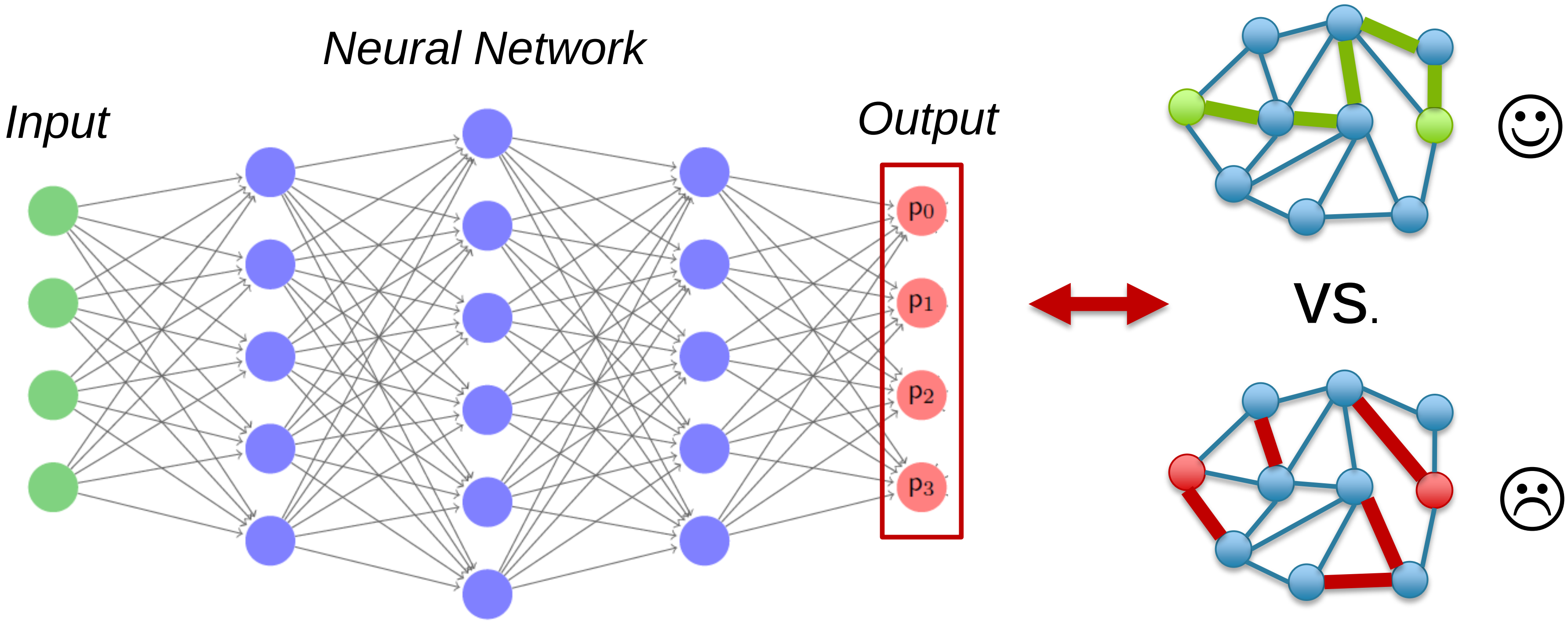
Learn

ML Model

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

Deep Learning with Symbolic Knowledge

cf. Nature paper



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

Semantic Loss

*<u>Q</u>: How close is output **p** to satisfying constraint α?*

*<u>Answer</u>: Semantic loss function L(α,**p**)*

- Axioms, for example:
  - If α constrains to one label, L(α,**p**) is cross-entropy
  - If α implies β then L(α,**p**) ≥ L(β,**p**)   (*α more strict*)

- Implied 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:

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

Simple Example: Exactly-One

- Data must have some label

  We agree this must be one of the 10 digits:

- Exactly-one constraint

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

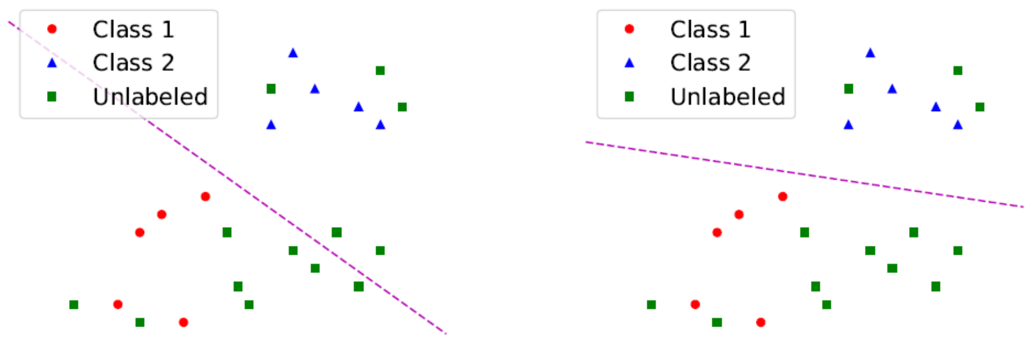
- Semantic loss:

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

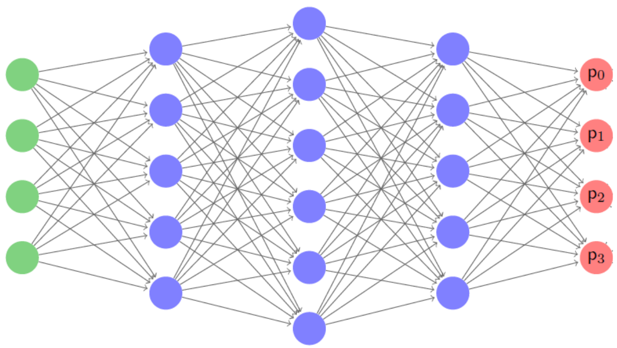
Semi-Supervised Learning

- Intuition: Unlabeled data must have some label

  Cf. entropy minimization, manifold learning



- Minimize exactly-one semantic loss on unlabeled data



Train with

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

Experimental Evaluation

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)	
Baseline: MLP with Entropy Regularizer	96.27 (±0.64)	98.32 (±0.34)	99.37 (±0.12)
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

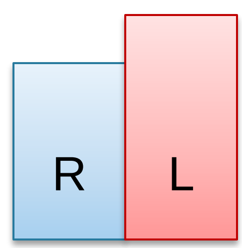
Accuracy % with # of used labels	100	500	1000	ALL
Ladder Net (Rasmus et al., 2015)	81.46 (±0.64)	85.18 (±0.27)	86.48 (±0.15)	90.46
Baseline: MLP, Gaussian Noise	69.45 (±2.03)	78.12 (±1.41)	80.94 (±0.84)	89.87
MLP with Semantic Loss	**86.74** (±0.71)	**89.49** (±0.24)	**89.67** (±0.09)	89.81

Outperforms SoA!

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

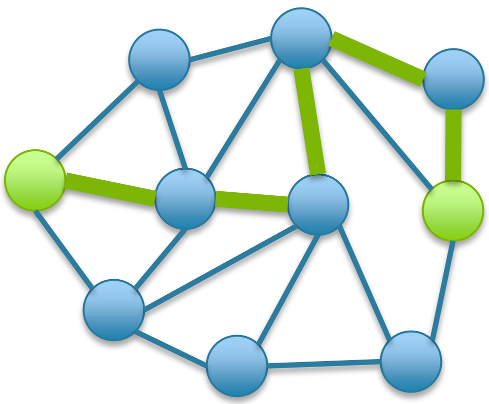
Efficient Reasoning During Learning



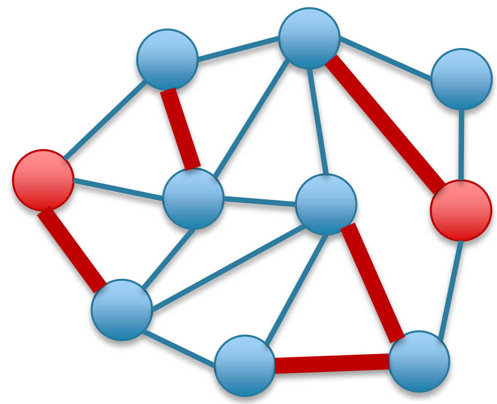
But what about *real* constraints?

cf. Nature paper

- Path constraint



vs.



- Example: 4x4 grids

  2^{24} = 184 paths + 16,777,032 non-paths

- Easily encoded as logical constraints ☺

[Nishino et al., Choi et al.]

A Semantic Loss Function

$$\mathrm{L}^{\mathrm{s}}(\alpha, \mathsf{p}) \propto -\log \underbrace{\sum_{\mathbf{x} \models \alpha} \prod_{i:\mathbf{x} \models X_i} \mathsf{p}_i \prod_{i:\mathbf{x} \models \neg X_i} (1 - \mathsf{p}_i)}$$

Probability of satisfying α after flipping coins with probabilities **p**

In general: #P-hard ☹

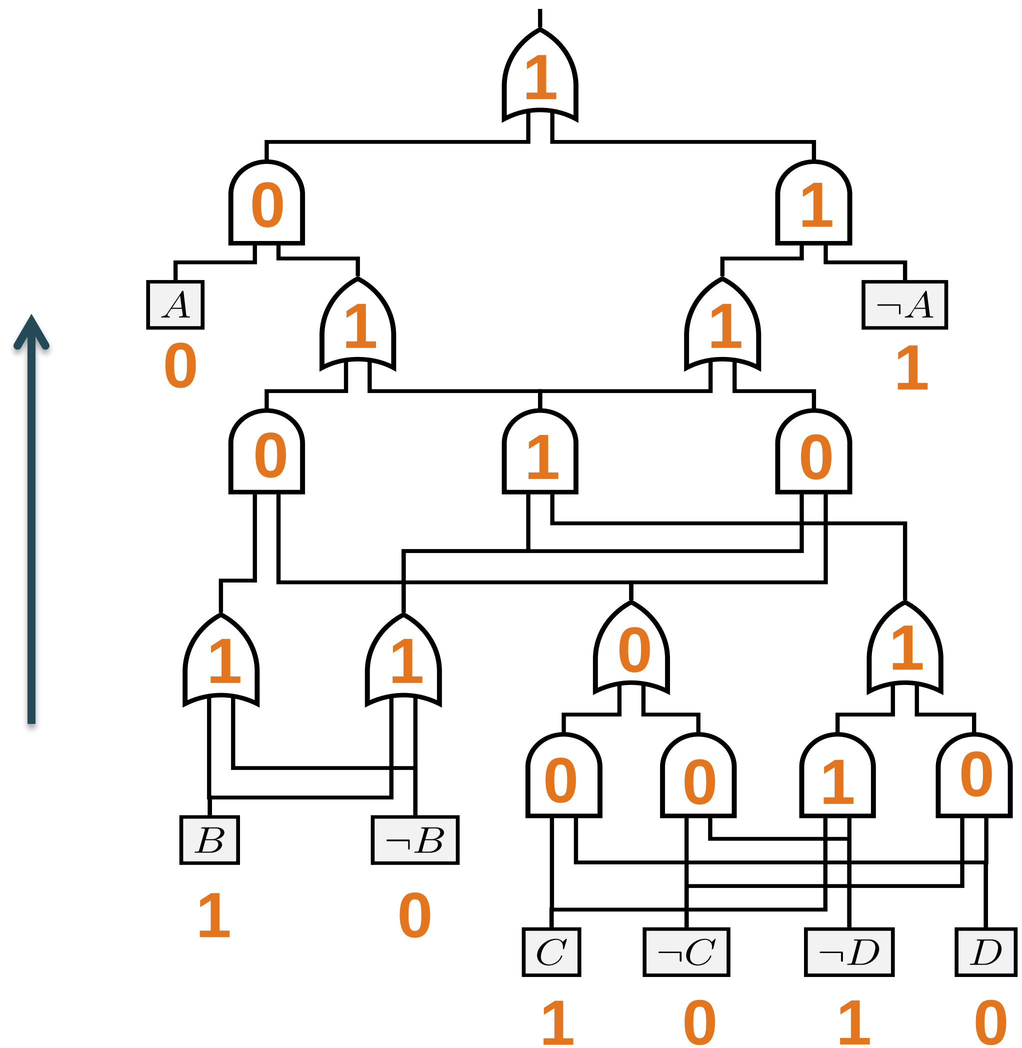
How to do this reasoning during learning?

Reasoning Tool: Logical Circuits

Representation of logical sentences:

Input:

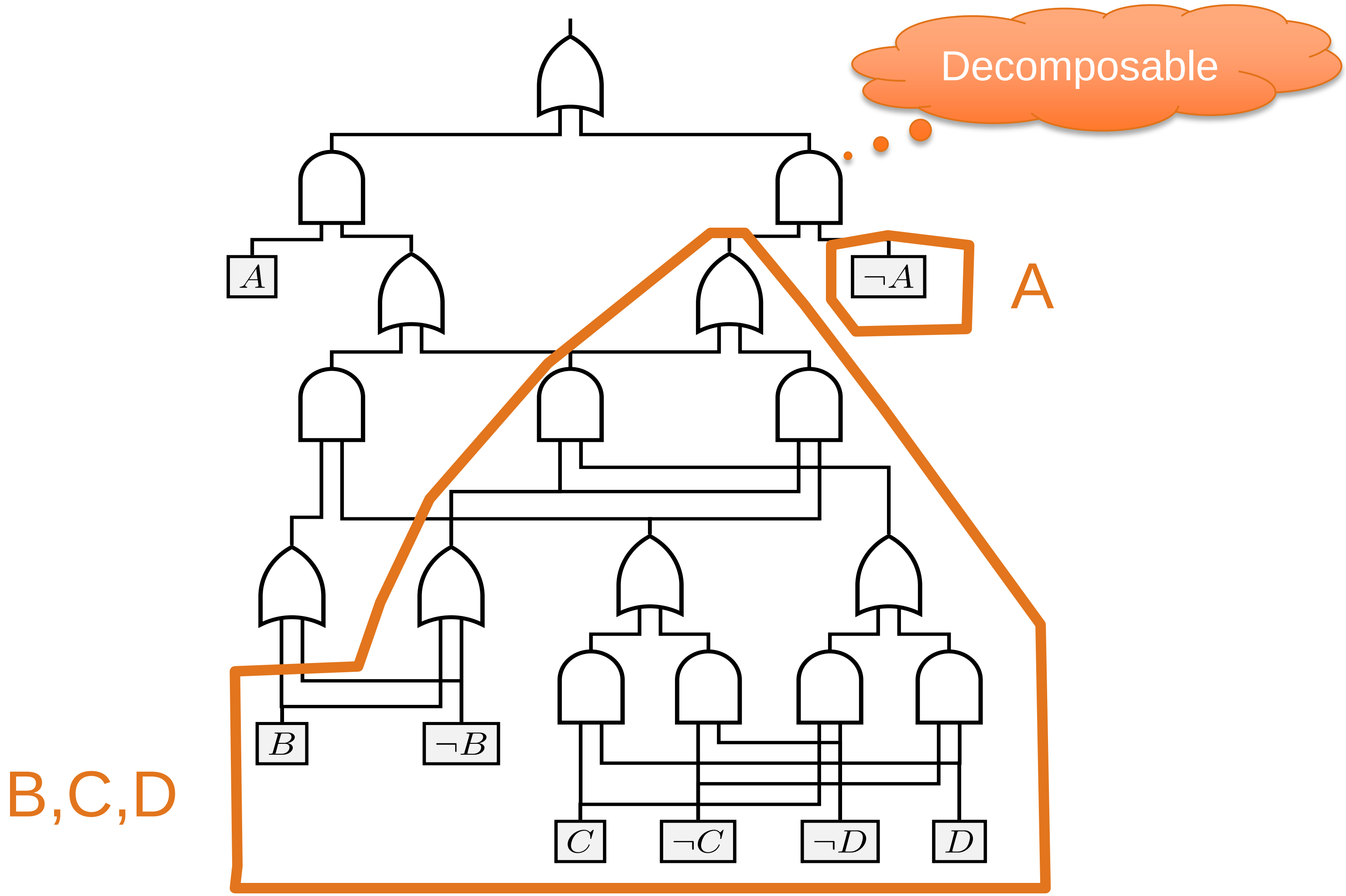
A	B	C	D
0	1	1	0



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 ***???***

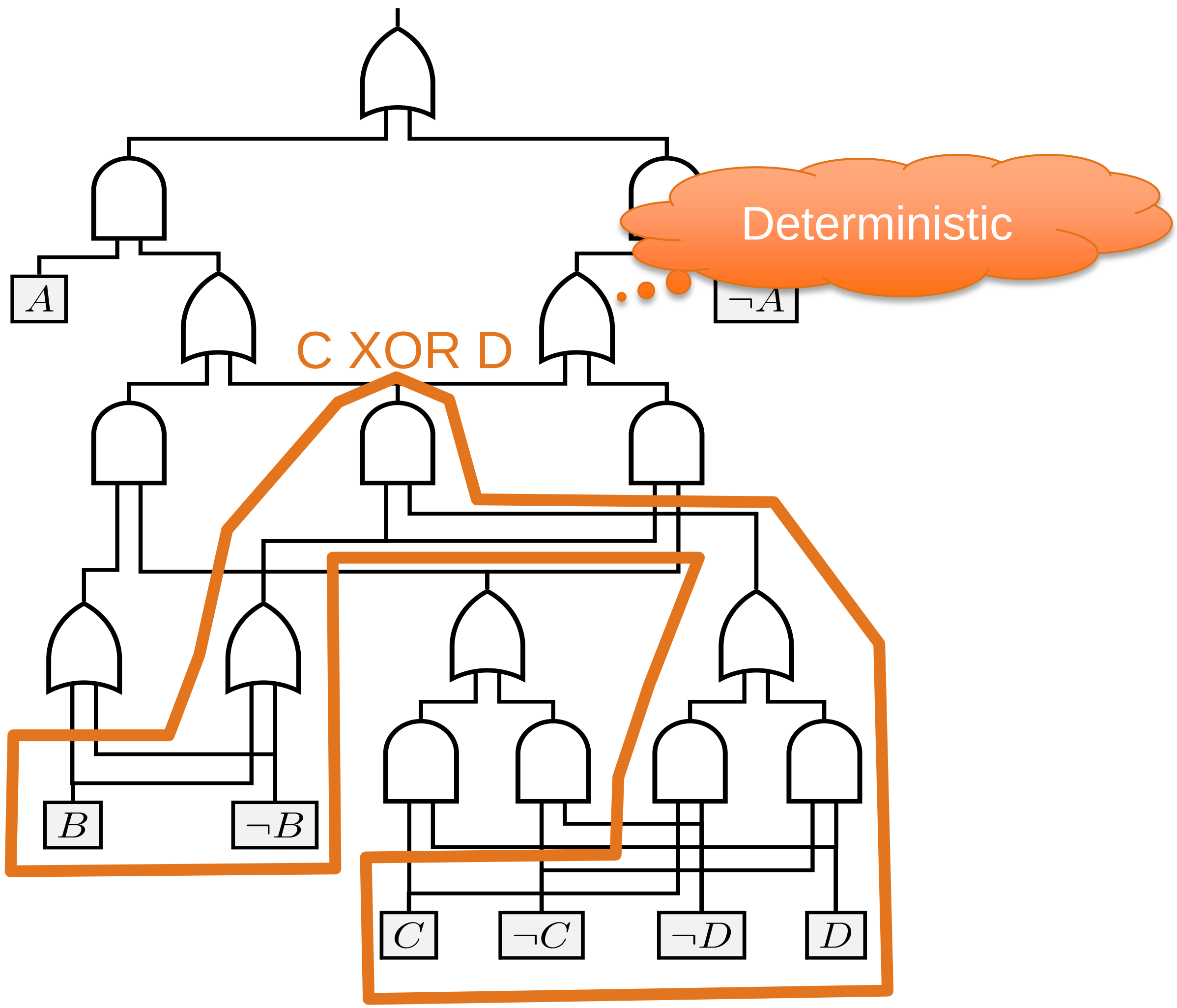
Decomposable Circuits



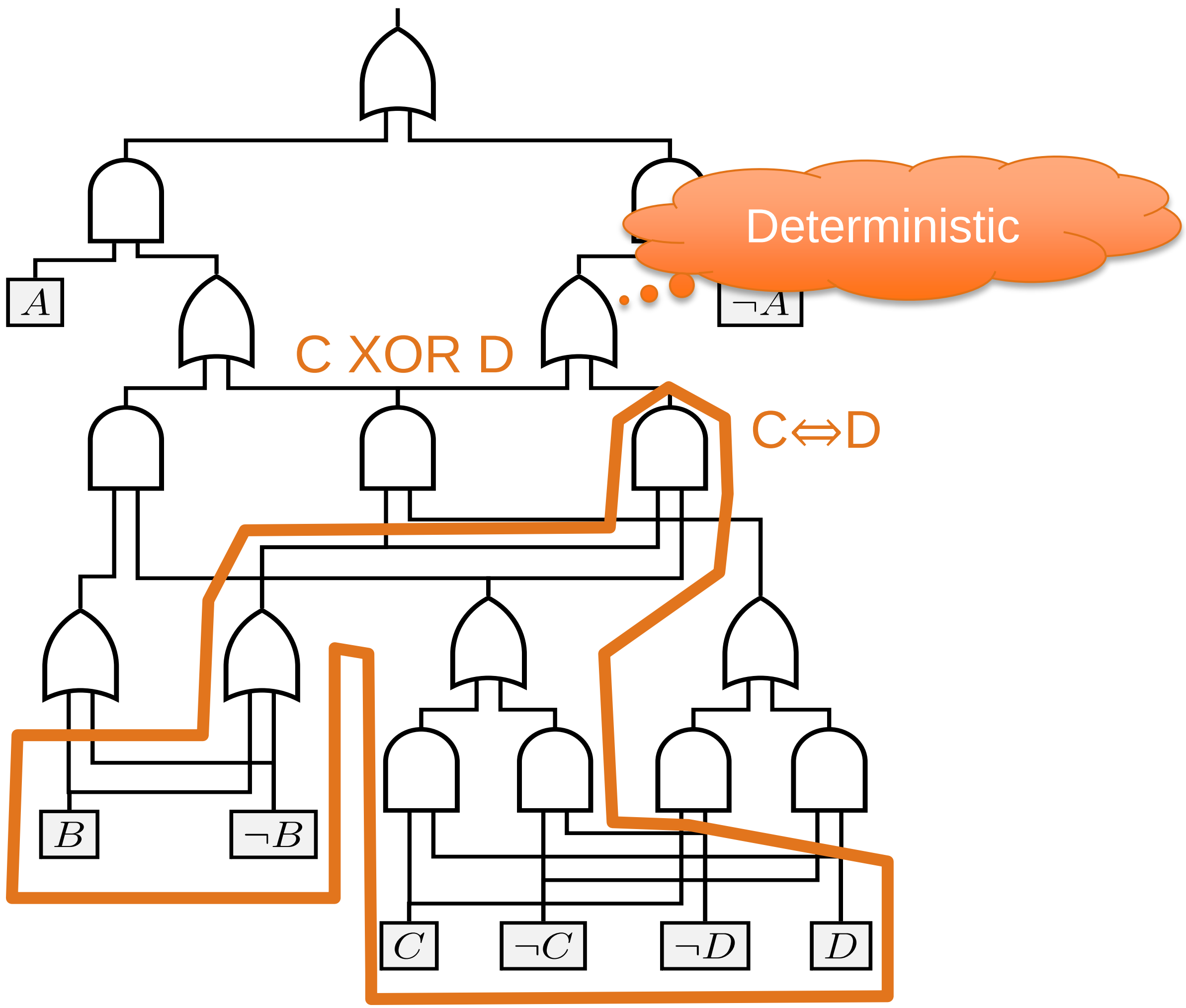
Tractable for Logical Inference

- 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 ☺

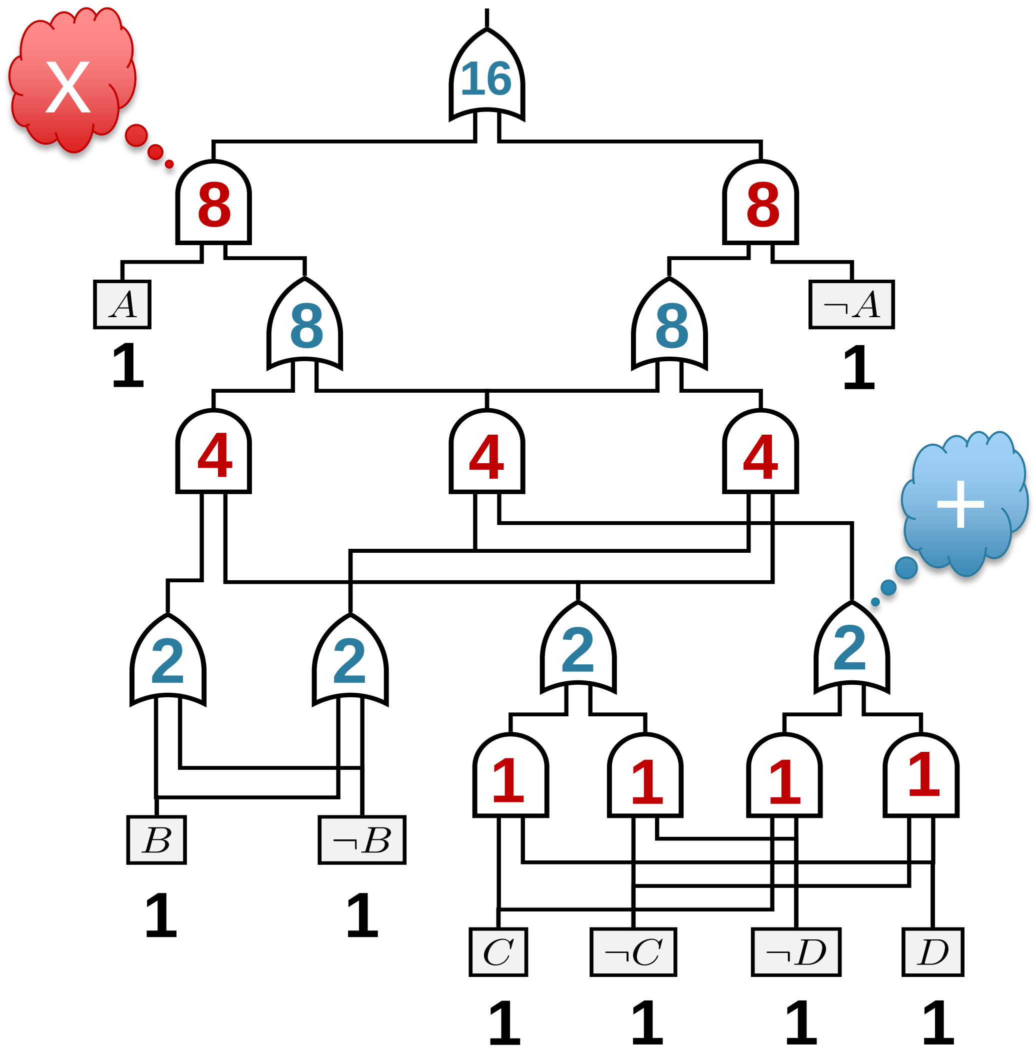
Deterministic Circuits



Deterministic Circuits

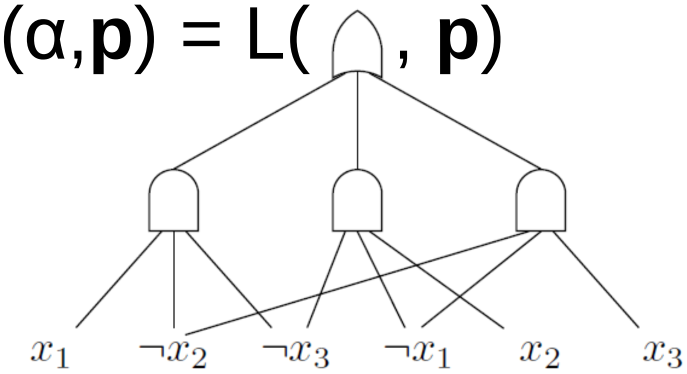


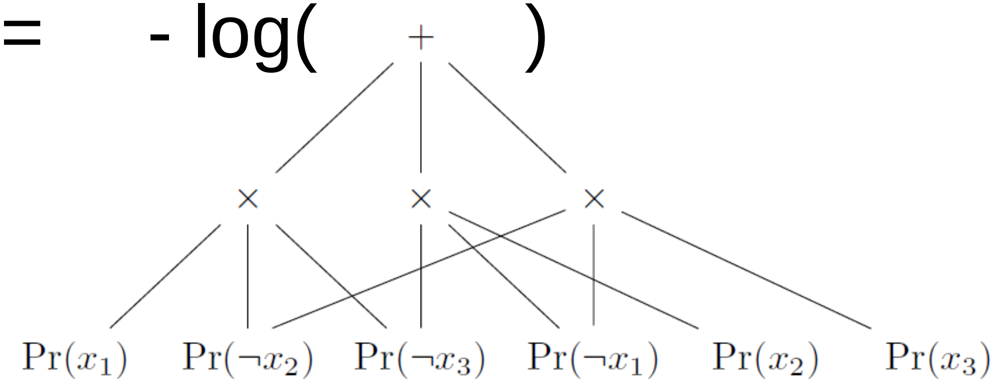
How many solutions are there? (#SAT)



Tractable for Inference

- Is there a solution? (SAT) ✓
- How many solutions are there? (#SAT) ✓
- And also semantic loss becomes tractable ✓

L(α,**p**) = L(

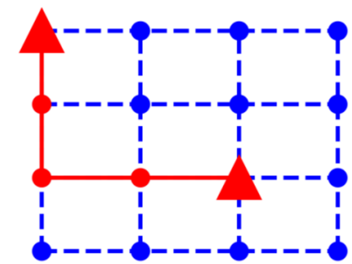
, **p**) = - log(

)

- Compilation into circuit by SAT solvers
- Add circuit to neural network output in tensorflow

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?

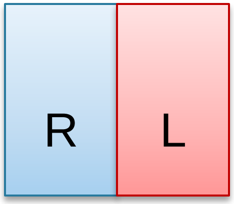
Is output a path?

(same conclusion for predicting sushi preferences, see paper)

Early Conclusions

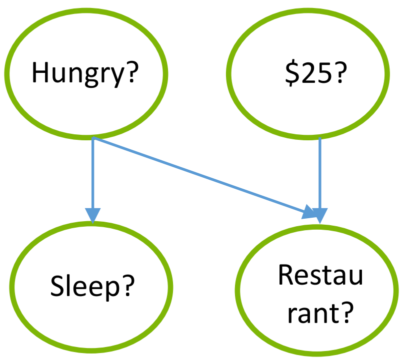
- Knowledge is (hidden) everywhere in ML
- Semantic loss makes logic differentiable
- Performs well semi-supervised
- Requires hard reasoning in general
  - Reasoning can be encapsulated in a circuit
  - No overhead during learning
- Performs well on structured prediction
- A little bit of reasoning goes a long way!

Probabilistic and Logistic Circuits

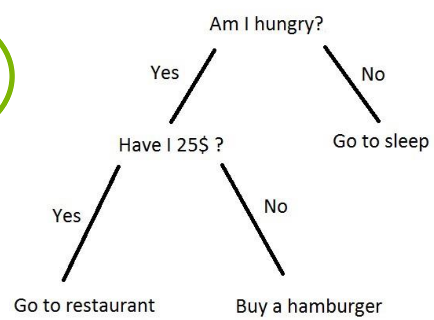


Another False Dilemma?

Classical AI Methods

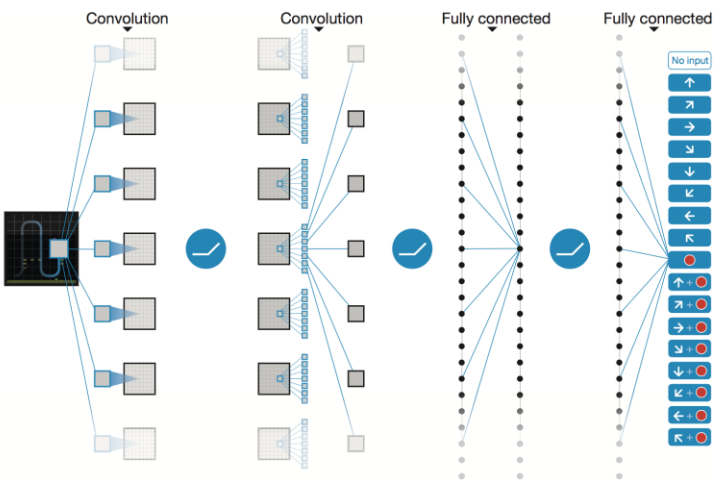


...



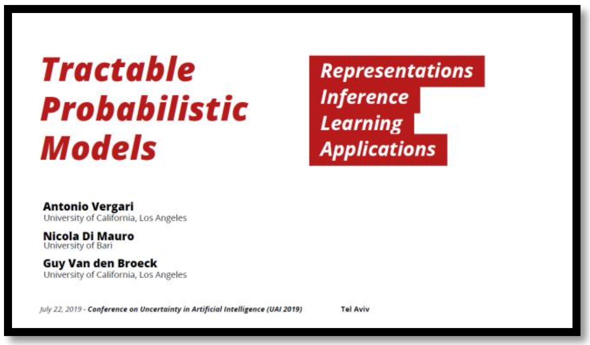
Clear Modeling Assumption
Well-understood

Neural Networks



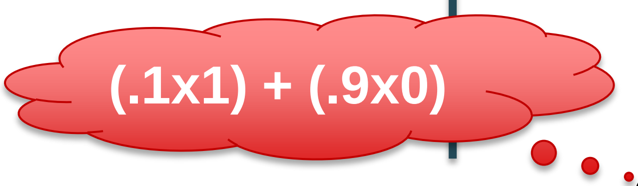
“Black Box”
Empirical performance

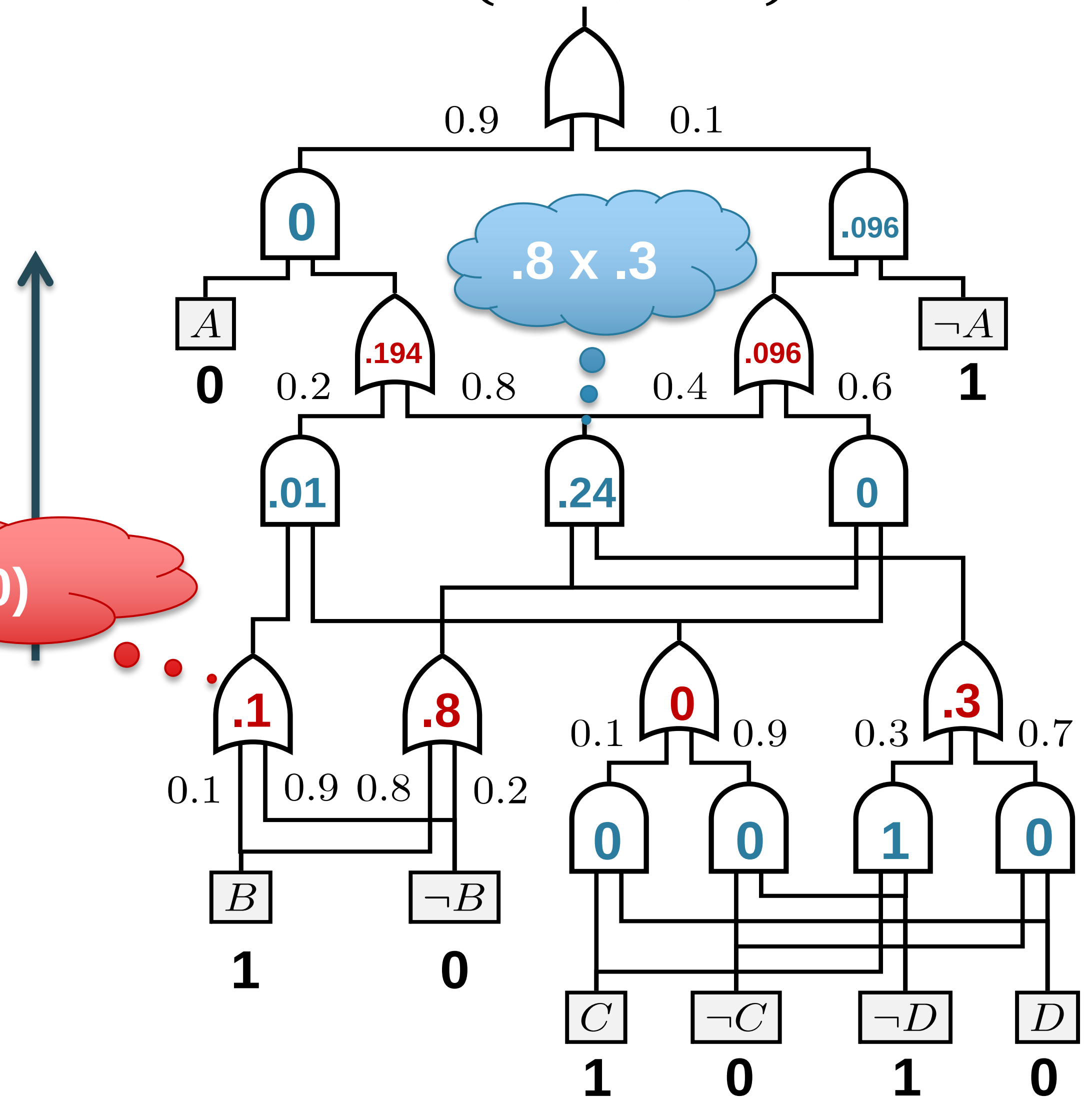
Probabilistic Circuits



SPNs, ACs
PSDDs, CNs

$$\mathbf{Pr}(A, B, C, D) = 0.096$$





Input:

A	B	C	D
0	1	1	0

Properties, Properties, Properties!

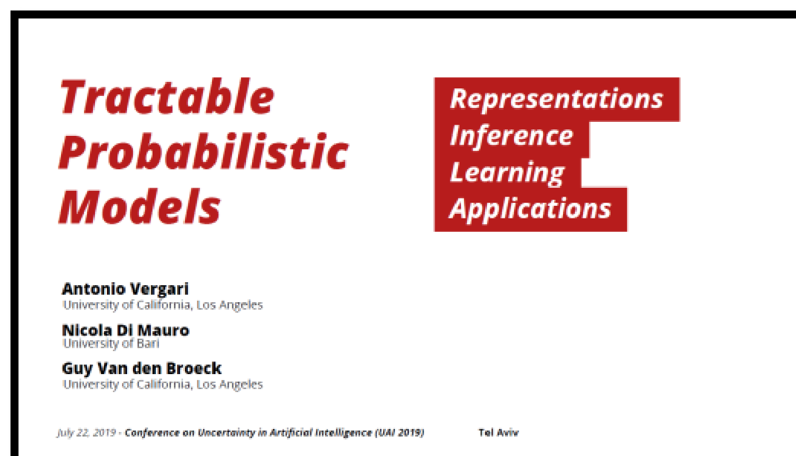
- Read conditional independencies from structure
- Interpretable parameters (XAI)
  (conditional probabilities of logical sentences)
- Closed-form parameter learning
- Efficient reasoning (linear ☺)
  - Computing **conditional probabilities** Pr(x|y)
  - **MAP inference**: most-likely assignment to x given y
  - Even much harder tasks: expectations, KLD, entropy, logical queries, decision making queries, etc.

Probabilistic Circuits: Performance

Density estimation benchmarks: tractable vs. intractable

Dataset	best circuit	BN	MADE	VAE
nltcs	**-5.99**	-6.02	-6.04	**-5.99**
msnbc	**-6.04**	**-6.04**	-6.06	-6.09
kdd2000	-2.12	-2.19	**-2.07**	-2.12
plants	**-11.84**	-12.65	12.32	-12.34
audio	-39.39	-40.50	-38.95	**-38.67**
jester	-51.29	**-51.07**	-52.23	-51.54
netflix	-55.71	-57.02	-55.16	**-54.73**
accidents	-26.89	**-26.32**	-26.42	-29.11
retail	**-10.72**	-10.87	-10.81	-10.83
*pumbs**	-22.15	**-21.72**	-22.3	-25.16
dna	**-79.88**	-80.65	-82.77	-94.56
Kosarek	**-10.52**	-10.83	-	-10.64
Msweb	-9.62	-9.70	**-9.59**	-9.73

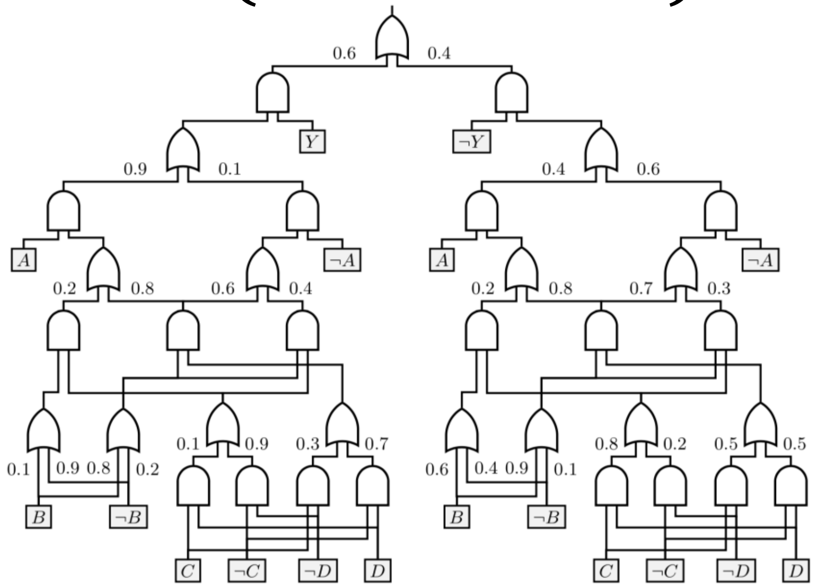
Dataset	best circuit	BN	MADE	VAE
Book	-33.82	-36.41	-33.95	**-33.19**
movie	-50.34	-54.37	-48.7	**-47.43**
webkb	-149.20	-157.43	-149.59	**-146.9**
cr52	-81.87	-87.56	-82.80	**-81.33**
c20ng	-151.02	-158.95	-153.18	**-146.90**
bbc	**-229.21**	-257.86	-242.40	-240.94
ad	-14.00	-18.35	**-13.65**	-18.81



But what if I only want to classify?

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

$$\cancel{\Pr(Y,A,B,C,D)}$$

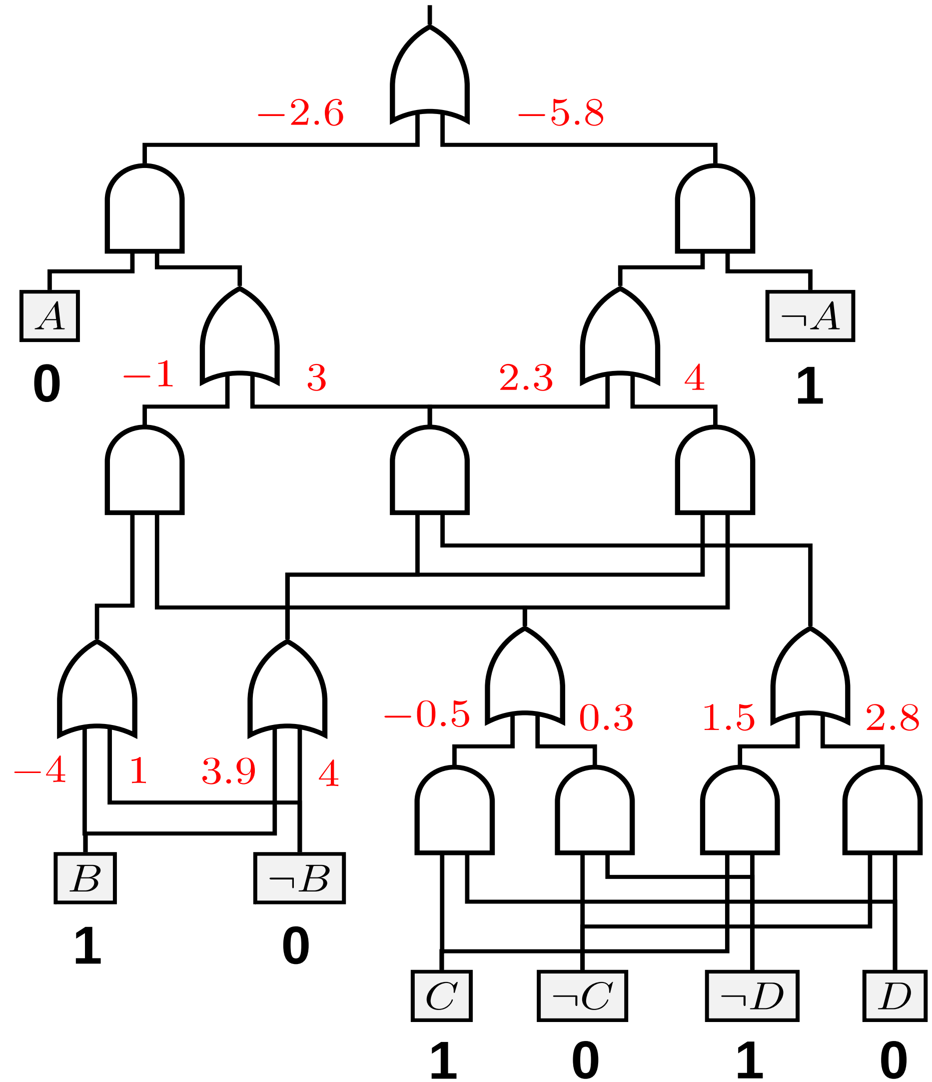


Learn a logistic circuit from data

Logistic Circuits

$$\Pr(Y = 1 \mid A, B, C, D)$$

$$= \frac{1}{1 + exp(-1.9)} = 0.869$$



Input:

A	B	C	D	$\Pr(Y \mid A, B, C, D)$
0	1	1	0	?

Learning Logistic Circuits

Parameter learning reduces 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!

Greedy structure learning (cf. decision trees)

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

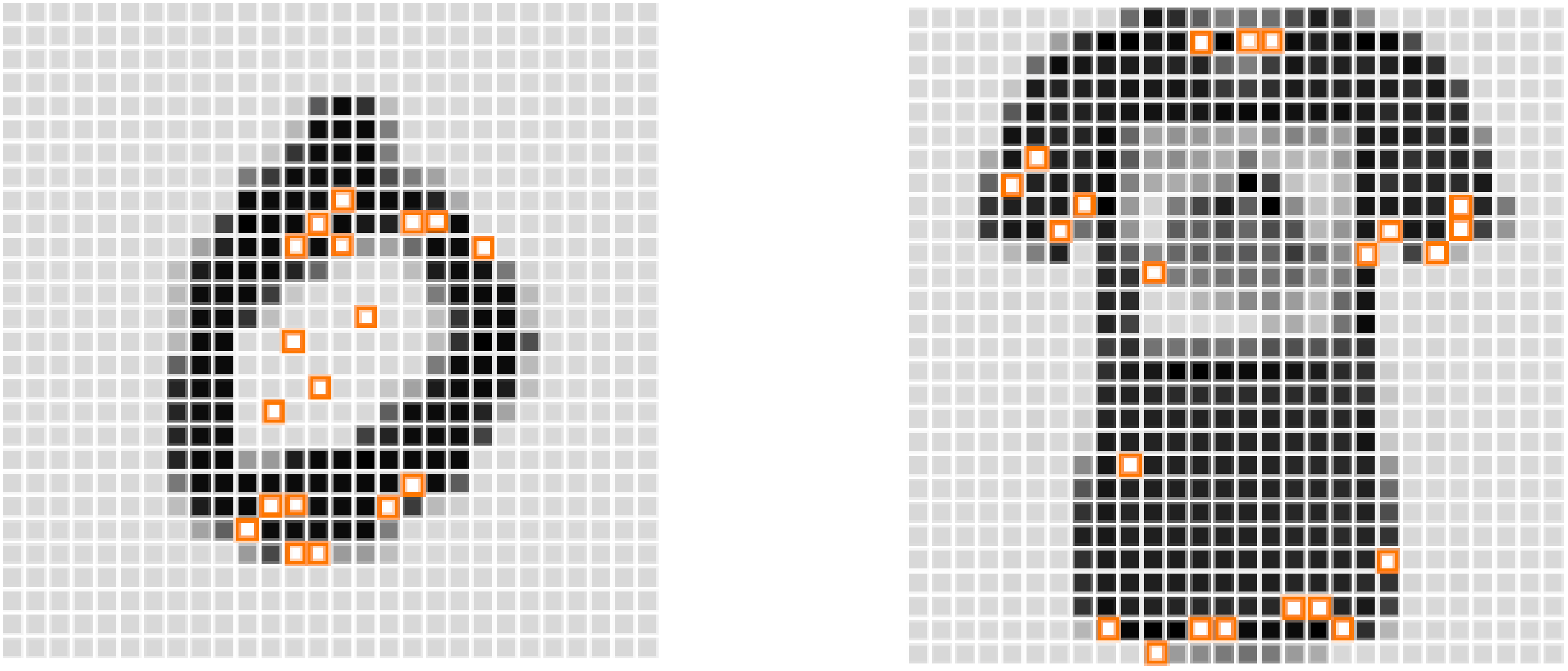
Significantly Smaller in Size

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

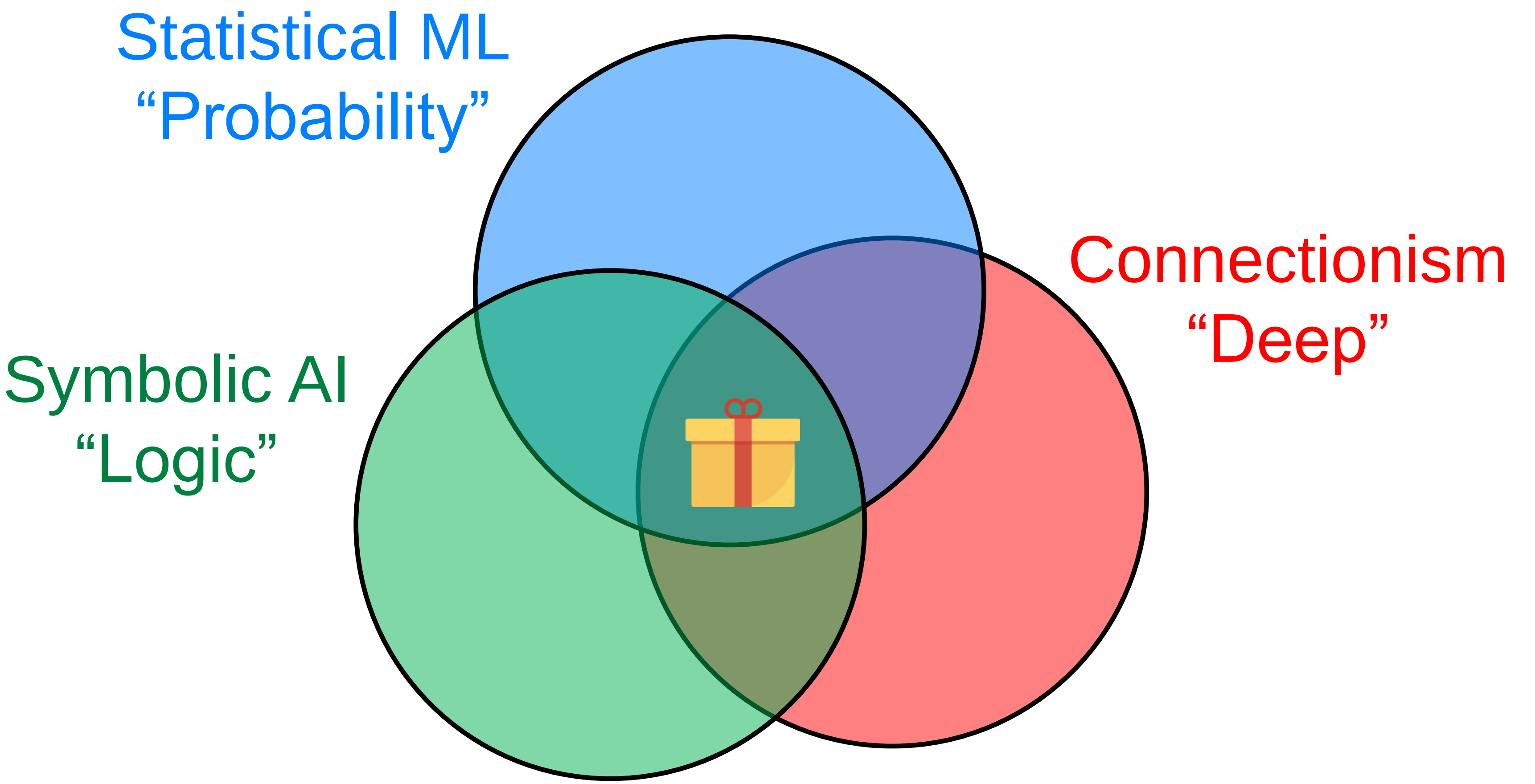
Better Data Efficiency

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?



Probabilistic & Logistic Circuits



Reasoning about
World Model + Classifier

"*Pure learning is brittle*"

bias, algorithmic fairness, interpretability, explainability, adversarial attacks, unknown unknowns, calibration, verification, missing features, missing labels, data efficiency, shift in distribution, general robustness and safety

fails to incorporate a sensible model of the world

- Given a learned predictor F(x)
- Given a probabilistic world model P(x)
- How does the world act on learned predictors?

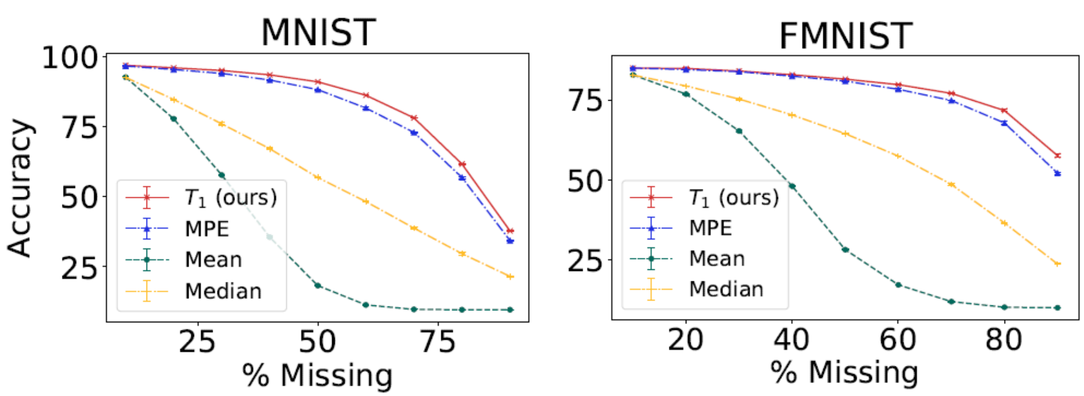
Can we solve these hard problems?

What to expect of classifiers?

- Missing features at prediction time
- What is expected prediction of F(x) in P(x)?

$$E_{\mathcal{F},P}(\mathbf{y}) = \mathop{\mathbb{E}}_{\mathbf{m}\sim P(\mathbf{M}|\mathbf{y})} [\mathcal{F}(\mathbf{ym})]$$

M: Missing features
y: Observed Features

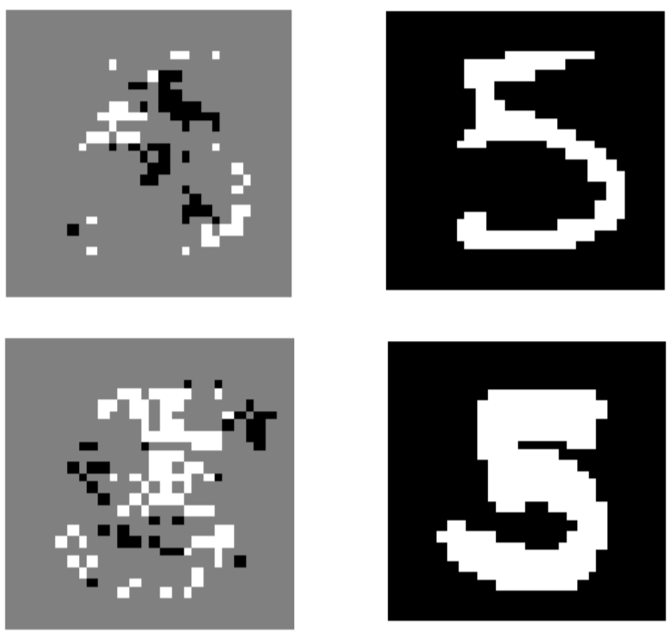


Explaining classifiers on the world

If the world looks like P(x),

then what part of the data is *sufficient* for

F(x) to make the prediction it makes?



Conclusions

Pure Logic **Probabilistic World Models** **Pure Learning**

Bring high-level representations, general knowledge, and efficient high-level reasoning to probabilistic models (*Weighted Model Integration, Probabilistic Programming*)

Bring back models of the world, supporting new tasks, and reasoning about what we have learned, without compromising learning performance

Conclusions

- There is a lot of value in working on pure logic, pure learning
- But we can do more
  by finding a synthesis, a confluence

Let’s get rid of this false dilemma…

Advertisements

- *Juice.jl* library for circuits and ML
  - Structure and parameter learning algorithms
  - Advanced reasoning algorithms with probabilistic and logical circuits
  - Scalable implementation in Julia

- AAAI 2020 Tutorial on Probabilistic Circuits

- Special Session for KR & ML at KR 2020
  - Submit in March! Go to Rhodes, Greece.

Thanks