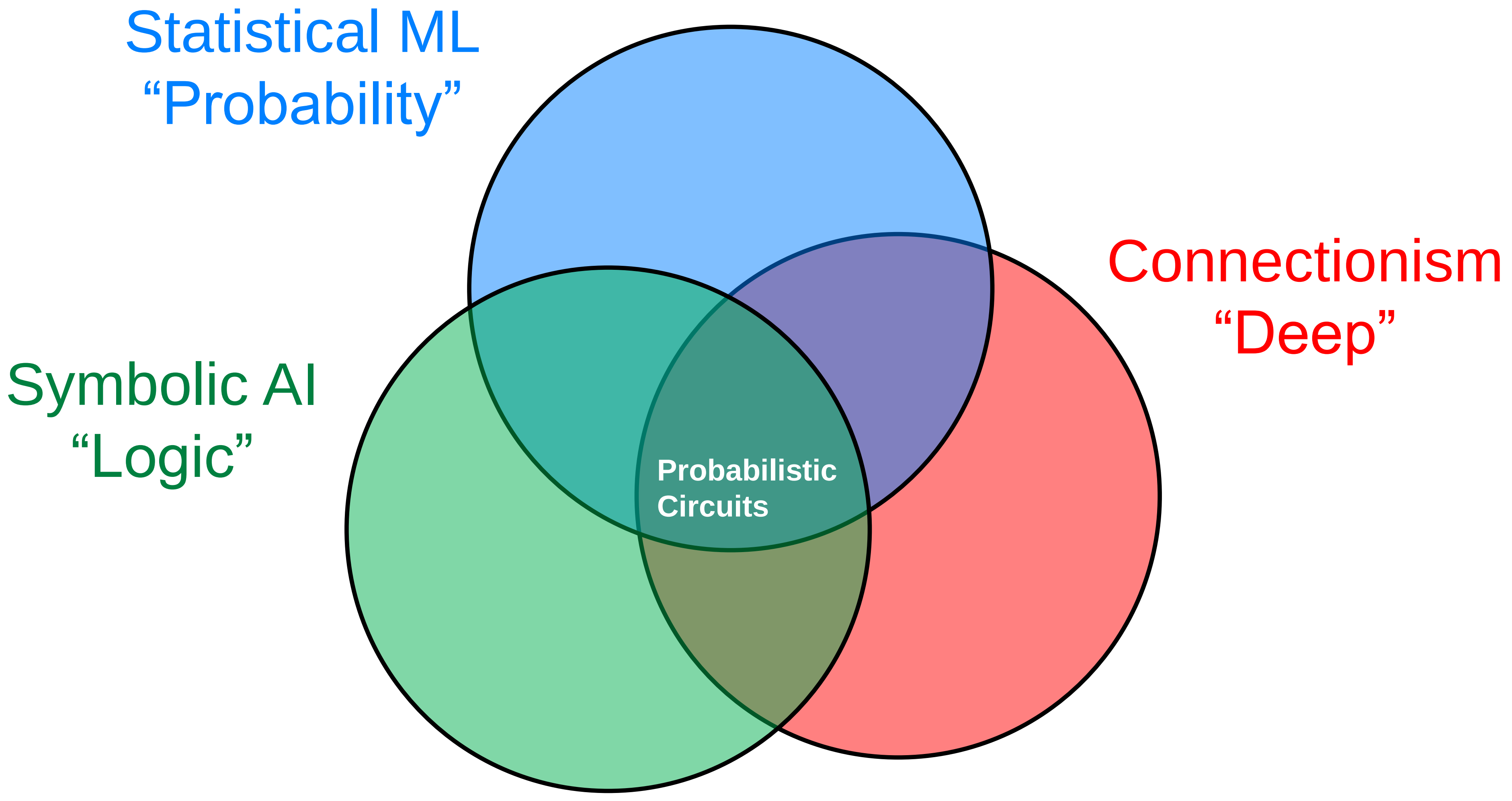
Probabilistic Circuits: A New Synthesis of Logic and Machine Learning

Guy Van den Broeck

UCSD

May 14, 2018

Overview


Statistical ML
“Probability”
Connectionism
“Deep”
Symbolic AI
“Logic”
Probabilistic
Circuits

References

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 International Conference on Machine Learning (ICML)*, 2018.

YooJung Choi and Guy Van den Broeck. **On Robust Trimming of Bayesian Network Classifiers**, *In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang and Guy Van den Broeck. **A Semantic Loss Function for Deep Learning Under Weak Supervision**, *In NIPS 2017 Workshop on Learning with Limited Labeled Data: Weak Supervision and Beyond*, 2017.

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.

YooJung Choi, Adnan Darwiche and Guy Van den Broeck. **Optimal Feature Selection for Decision Robustness in Bayesian Networks**, *In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

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.

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.

Arthur Choi, Guy Van den Broeck and Adnan Darwiche. **Probability Distributions over Structured Spaces**, *In Proceedings of the AAAI Spring Symposium on KRR*, 2015.

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.

Doga Kisa, Guy Van den Broeck, Arthur Choi and Adnan Darwiche. **Probabilistic sentential decision diagrams: Learning with massive logical constraints**, *In ICML Workshop on Learning Tractable Probabilistic Models (LTPM)*, 2014.

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.

(… and ongoing work by Tal Friedman, YooJung Choi, and Yitao Liang)

Structured Spaces

Running Example

Courses:

- Logic (L)
- Knowledge Representation (K)
- Probability (P)
- Artificial Intelligence (A)

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

Running Example

Courses:

- Logic (L)
- Knowledge Representation (K)
- Probability (P)
- Artificial Intelligence (A)

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

Constraints

- Must take at least one of Probability or Logic.
- Probability is a prerequisite for AI.
- The prerequisites for KR is either AI or Logic.

Structured Space

unstructured

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Structured Space

unstructured

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

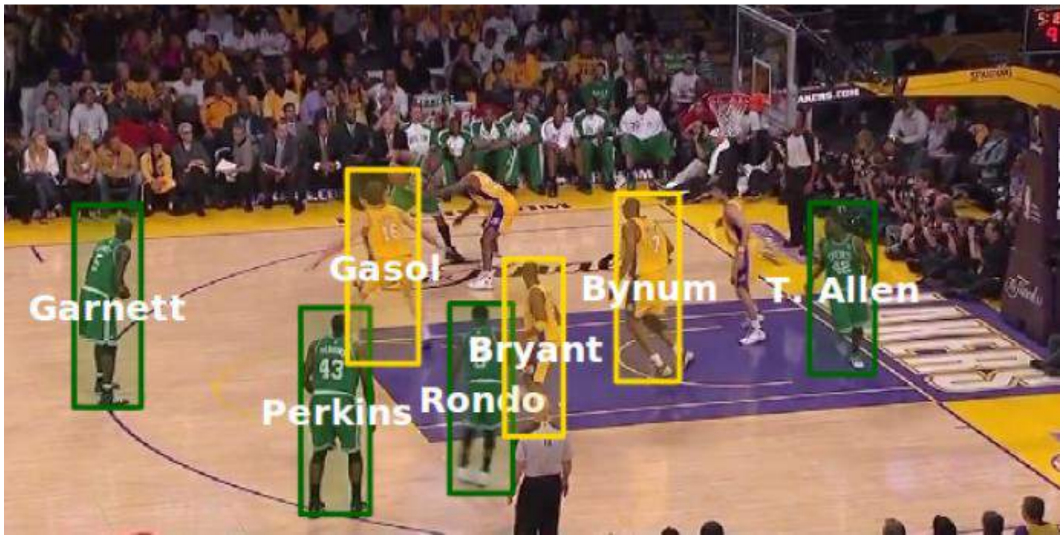
- Must take at least one of Probability (**P**) or Logic (**L**).
- Probability is a prerequisite for AI (**A**).
- The prerequisites for KR (**K**) is either AI or Logic.

7 out of 16 instantiations are impossible

structured

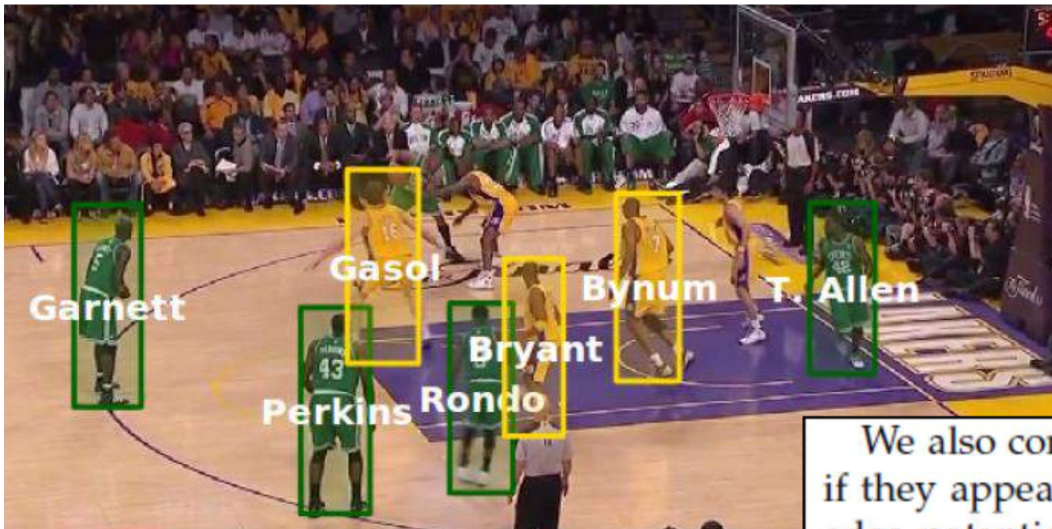
L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Example: Video



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

Example: Video



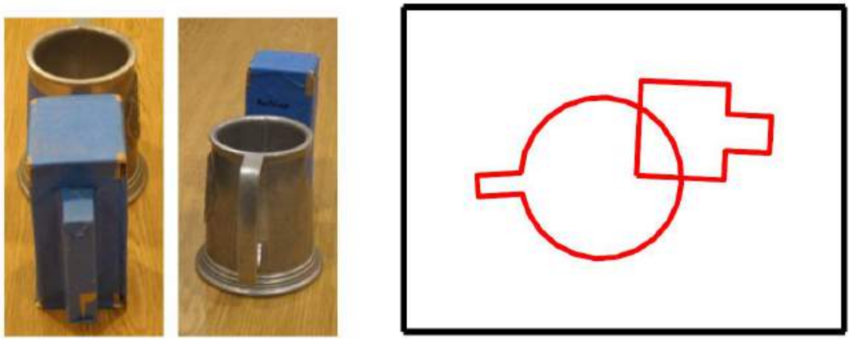
We also connect all pairs of identity nodes $y_{t,i}$ and $y_{t,j}$ if they appear in the same time t. We then introduce an edge potential that enforces mutual exclusion:

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

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.

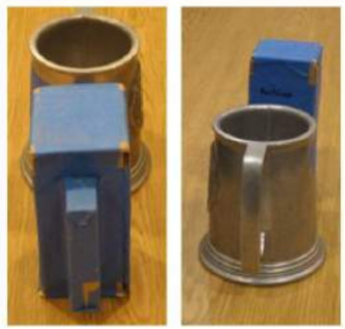
[Lu, W. L., Ting, J. A., Little, J. J., & Murphy, K. P. (2013). Learning to track and identify players from broadcast sports videos.]

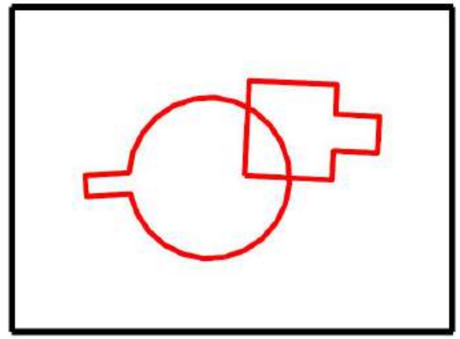
Example: Robotics

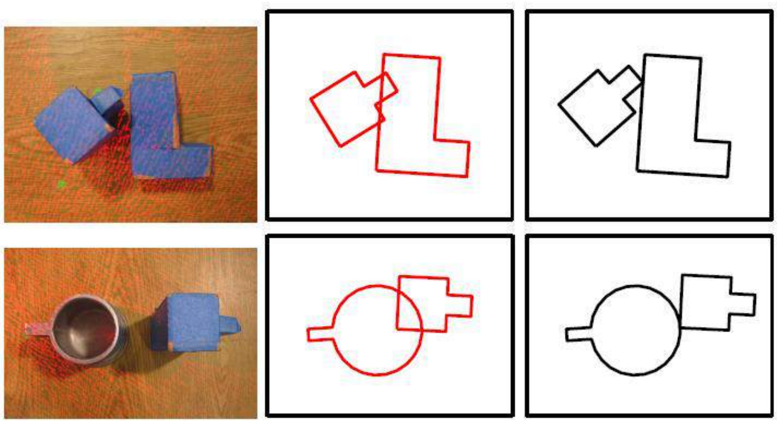


[Wong, L. L., Kaelbling, L. P., & Lozano-Perez, T., Collision-free state estimation. ICRA 2012]

Example: 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]

Example: Language

- Non-local dependencies:

  At least one verb in each sentence

[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]

Example: Language

- Non-local dependencies:

  At least one verb in each sentence

- Sentence compression

  If a modifier is kept, its subject is also kept

[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]

Example: 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

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]

Example: Language

- Non-local dependencies:

  At least one verb in each sentence

- Sentence compression

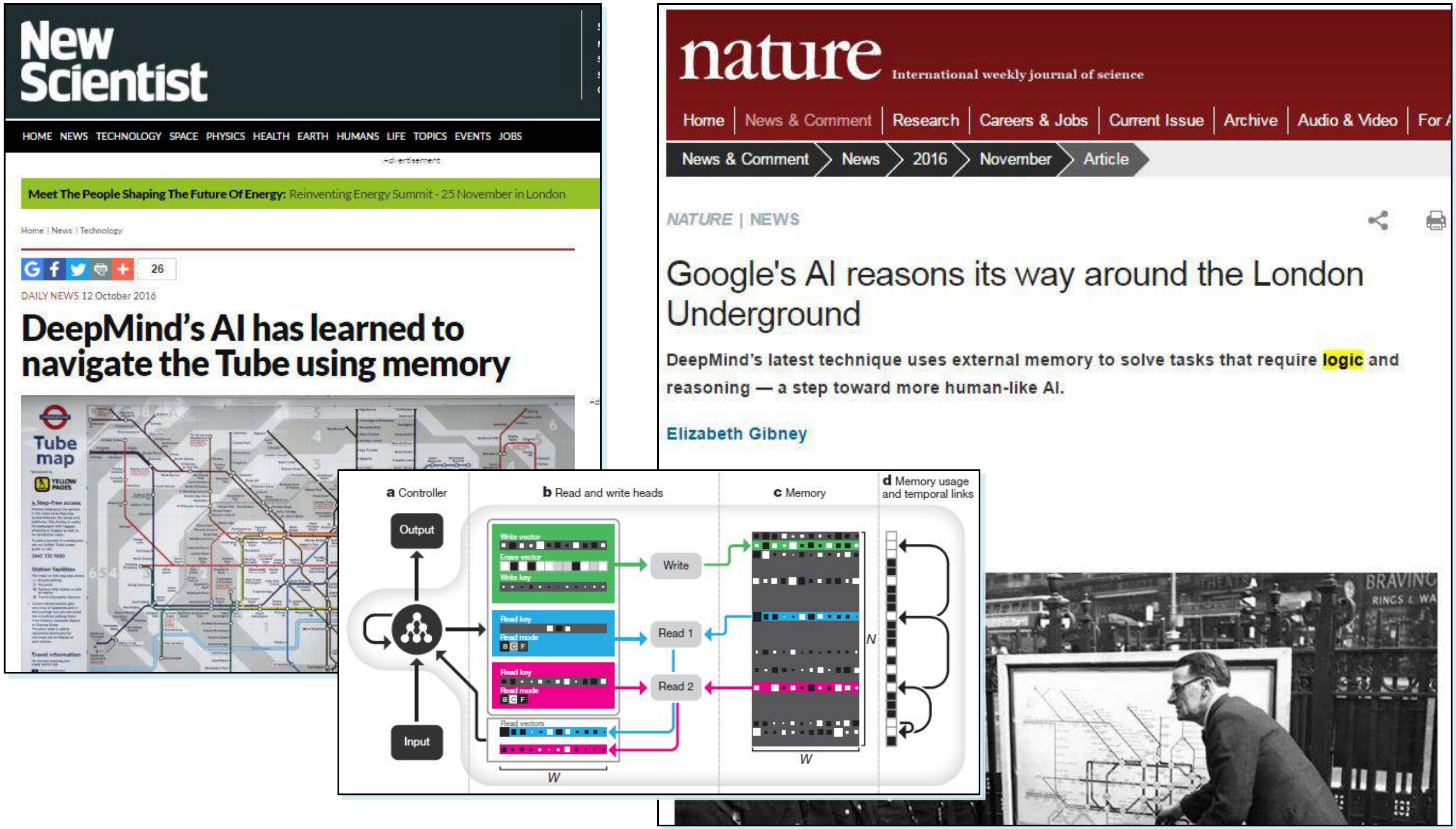
  If a modifier is kept, its subject is also kept

- 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]

Example: Deep Learning

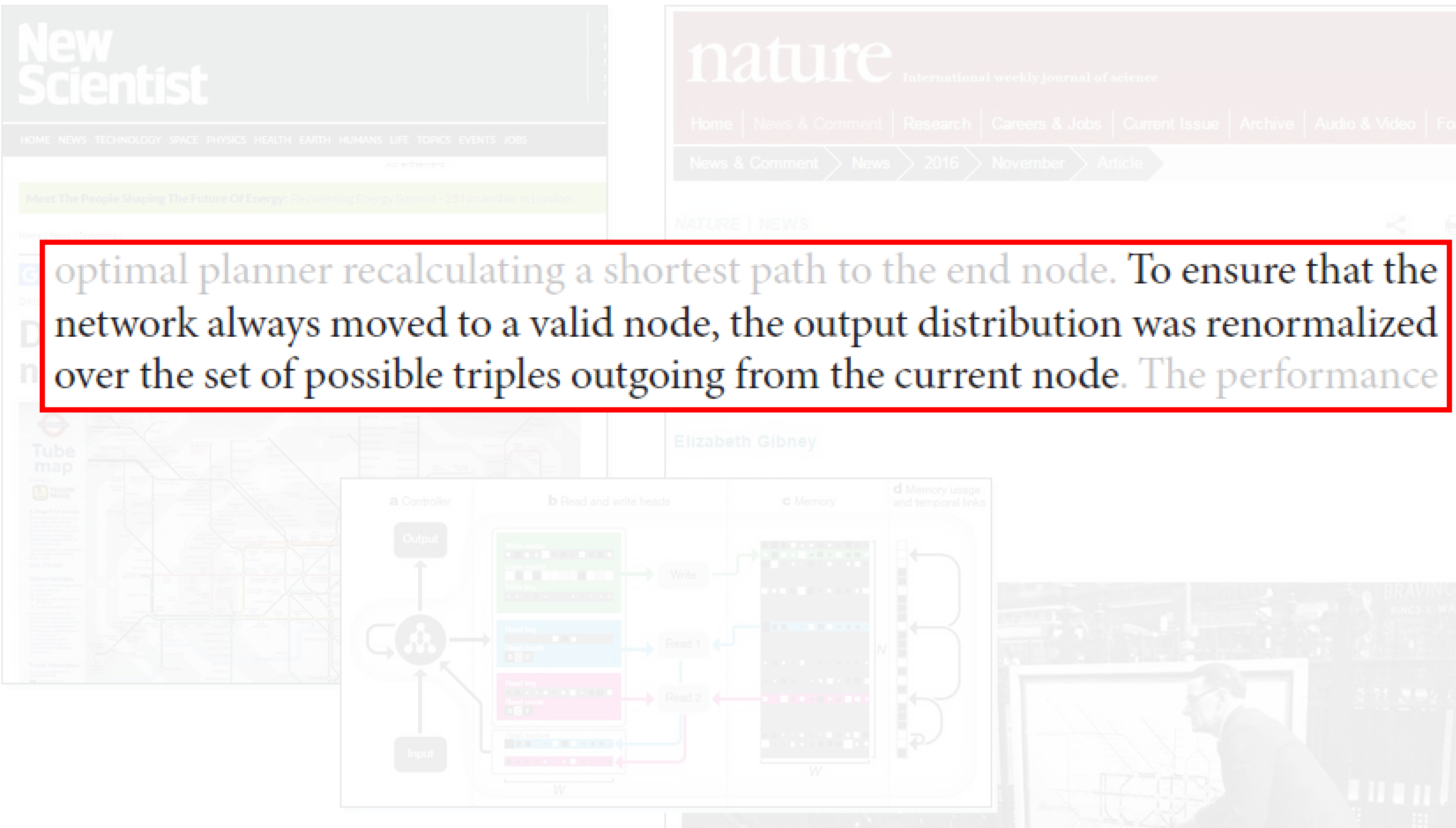


[Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al.. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.]

Example: Deep Learning

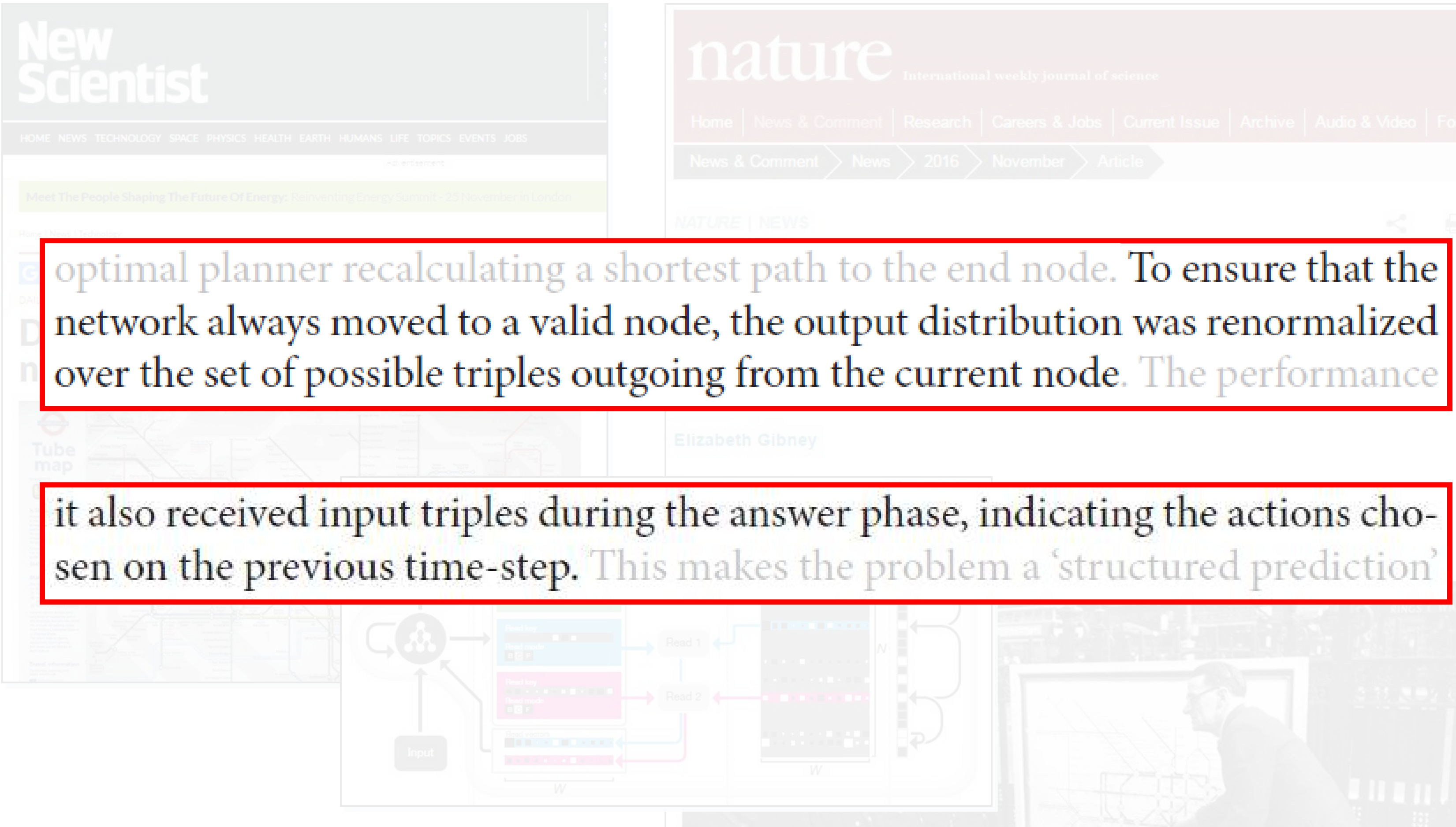
[Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al.. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.]

Example: 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.]

Example: 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

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)

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

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

(Distribution) (Neural Network)

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
(Distribution) (Neural Network)

Statistical ML tools don't take constraints as input! ☹

Specification Language: Logic

Structured Probability Space

unstructured

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

- Must take at least one of Probability or Logic.
- Probability is a prerequisite for AI.
- The prerequisites for KR is either AI or Logic.

7 out of 16 instantiations are impossible

structured

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Boolean Constraints

unstructured

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

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

7 out of 16 instantiations are impossible

structured

L	K	P	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Combinatorial Objects: Rankings

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

rank	sushi
1	shrimp
2	sea urchin
3	salmon roe
4	fatty tuna
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

10 items:
3,628,800
rankings

20 items:
2,432,902,008,176,640,000
rankings

Combinatorial Objects: Rankings

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

rank	sushi
1	shrimp
2	sea urchin
3	salmon roe
4	fatty tuna
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

A_{ij} item i at position j (n items require n^2 Boolean variables)

Combinatorial Objects: Rankings

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

rank	sushi
1	shrimp
2	sea urchin
3	salmon roe
4	fatty tuna
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

A_{ij} item i at position j (n items require n^2 Boolean variables)

An item may be assigned to more than one position

A position may contain more than one item

Encoding Rankings in Logic

A_{ij} : item i at position j

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

Encoding Rankings in Logic

A_{ij} : item i at position j

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

constraint: each item i assigned to a unique position (n constraints)

$$\bigvee_j A_{ij} \wedge \left(\bigwedge_{k \neq j} \neg A_{ik} \right)$$

Encoding Rankings in Logic

A_{ij} : item i at position j

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

constraint: each item i assigned to a unique position (n constraints)

$$\bigvee_j A_{ij} \wedge \left(\bigwedge_{k \neq j} \neg A_{ik} \right)$$

constraint: each position j assigned a unique item (n constraints)

$$\bigvee_i A_{ij} \wedge \left(\bigwedge_{k \neq i} \neg A_{kj} \right)$$

Encoding Rankings in Logic

A_{ij} : item *i* at position *j*

	pos 1	pos 2	pos 3	pos 4
item 1	A_{11}	A_{12}	A_{13}	A_{14}
item 2	A_{21}	A_{22}	A_{23}	A_{24}
item 3	A_{31}	A_{32}	A_{33}	A_{34}
item 4	A_{41}	A_{42}	A_{43}	A_{44}

constraint: each item *i* assigned to a unique position (*n* constraints)

$$\bigvee_j A_{ij} \wedge \left(\bigwedge_{k \neq j} \neg A_{ik} \right)$$

constraint: each position *j* assigned a unique item (*n* constraints)

$$\bigvee_i A_{ij} \wedge \left(\bigwedge_{k \neq i} \neg A_{kj} \right)$$

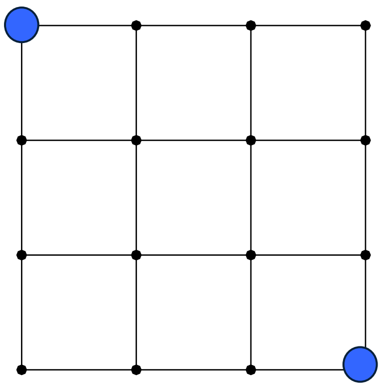
total constraints	$2n$
unstructured space	2^{n^2}
structured space	$n!$

Structured Space for Paths

cf. Nature paper

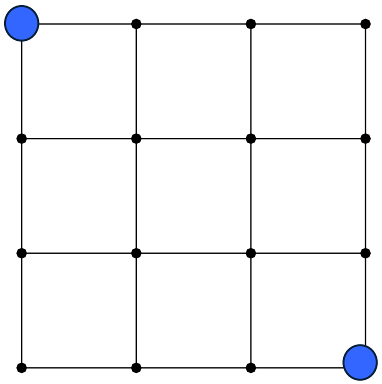

nature

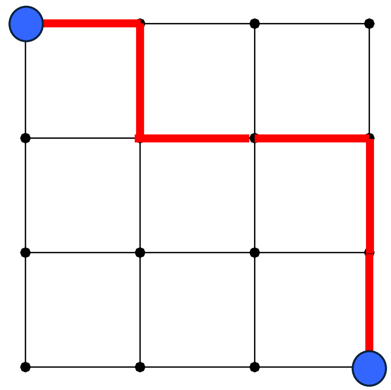
Google's AI reasons its way around the London Underground



Structured Space for Paths

cf. Nature paper





Good variable assignment (represents route)

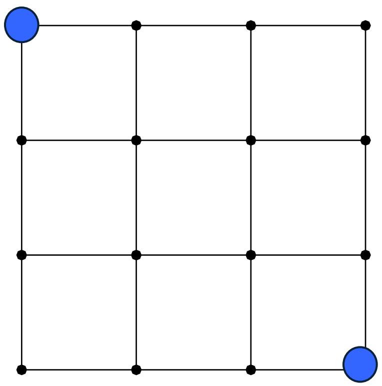
184

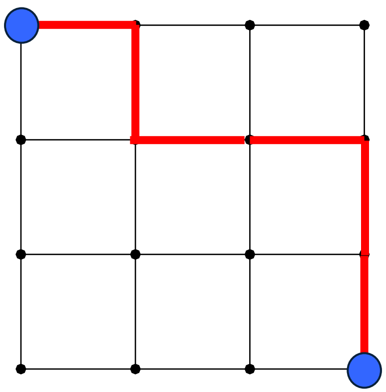
Structured Space for Paths

cf. Nature paper


nature

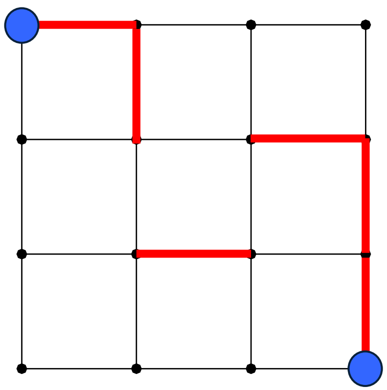
Google's AI reasons its way around the London Underground





Good variable assignment
(represents route)

184

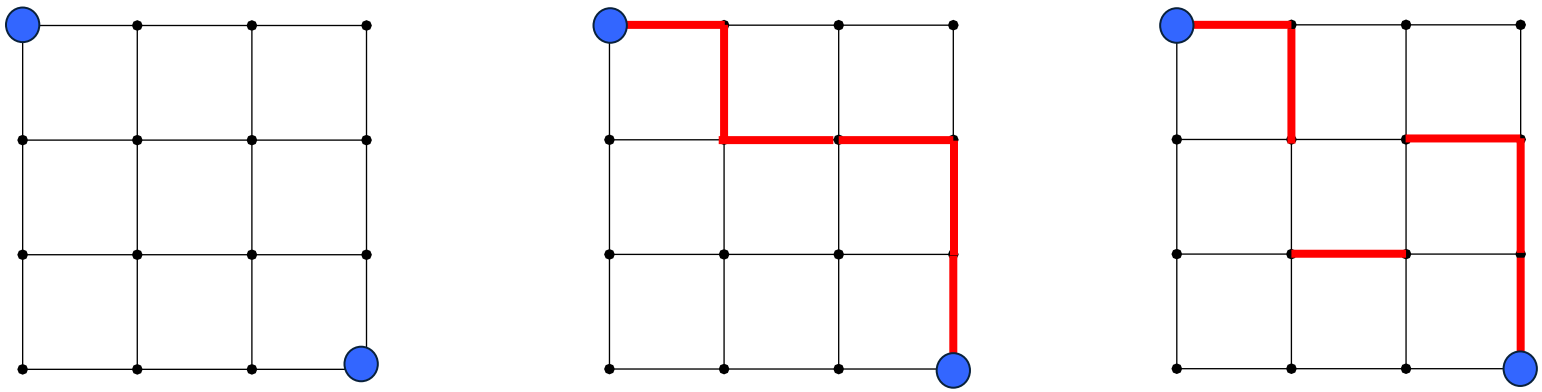


Bad variable assignment
(does not represent route)

16,777,032

Structured Space for Paths

cf. Nature paper



Good variable assignment (represents route)

184

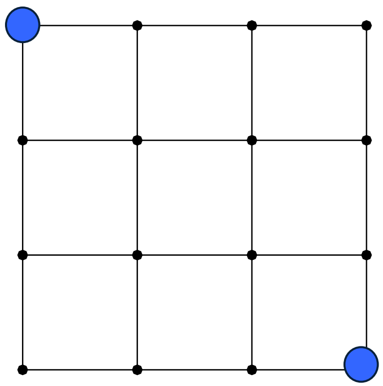
Bad variable assignment (does not represent route)

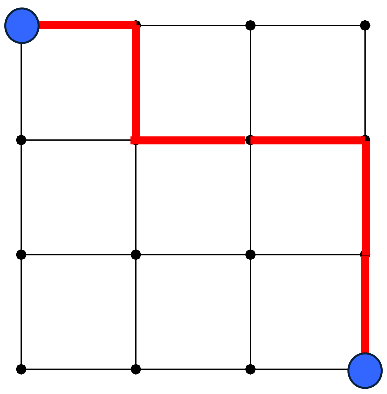
16,777,032

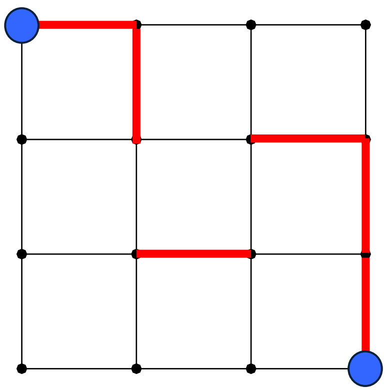
Space easily encoded in logical constraints ☺ [Nishino et al.]

Structured Space for Paths

cf. Nature paper







Good variable assignment (represents route)	**Bad variable assignment (does not represent route)**
184	**16,777,032**

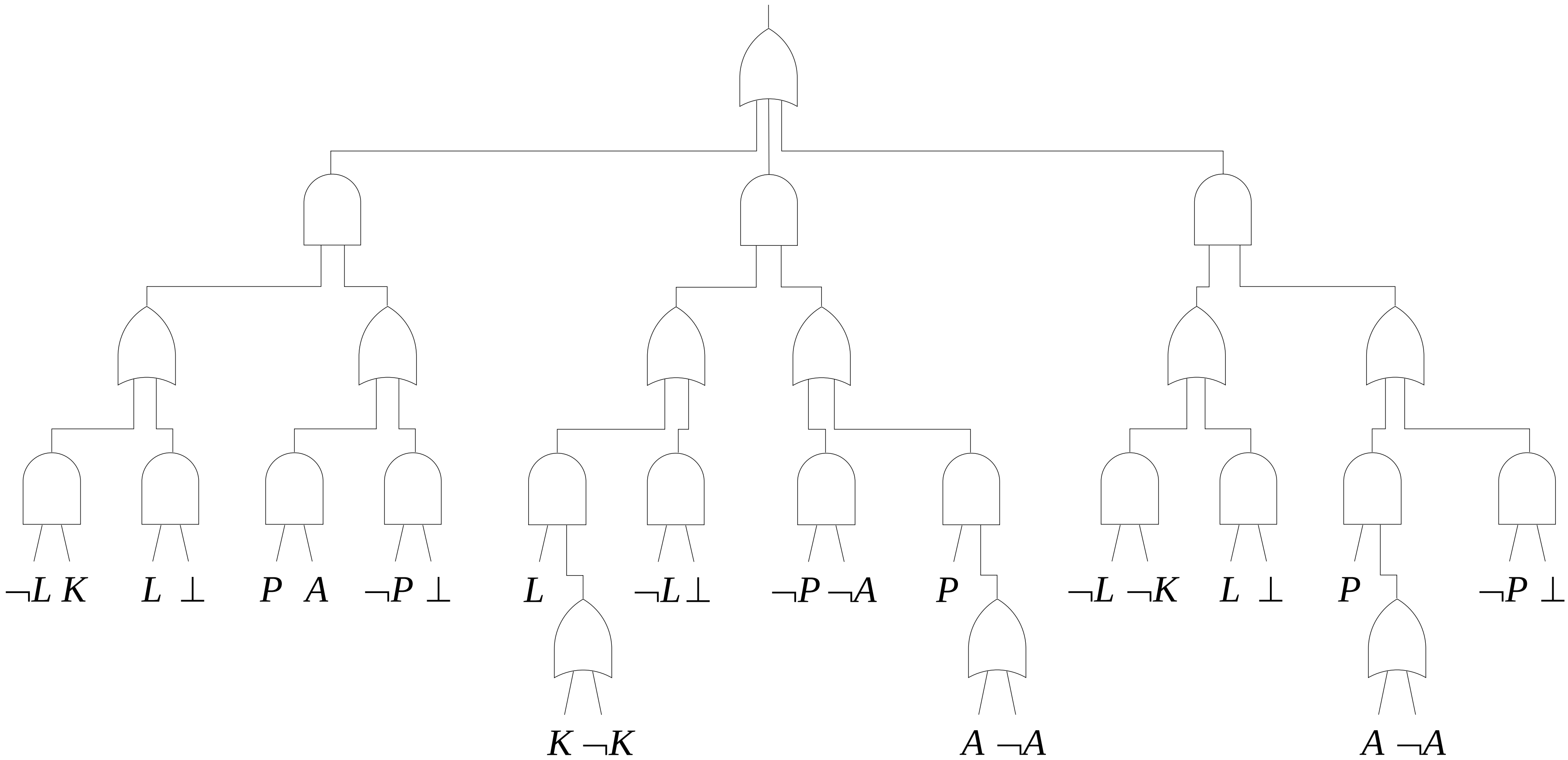
Space easily encoded in logical constraints ☺ [Nishino et al.]

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

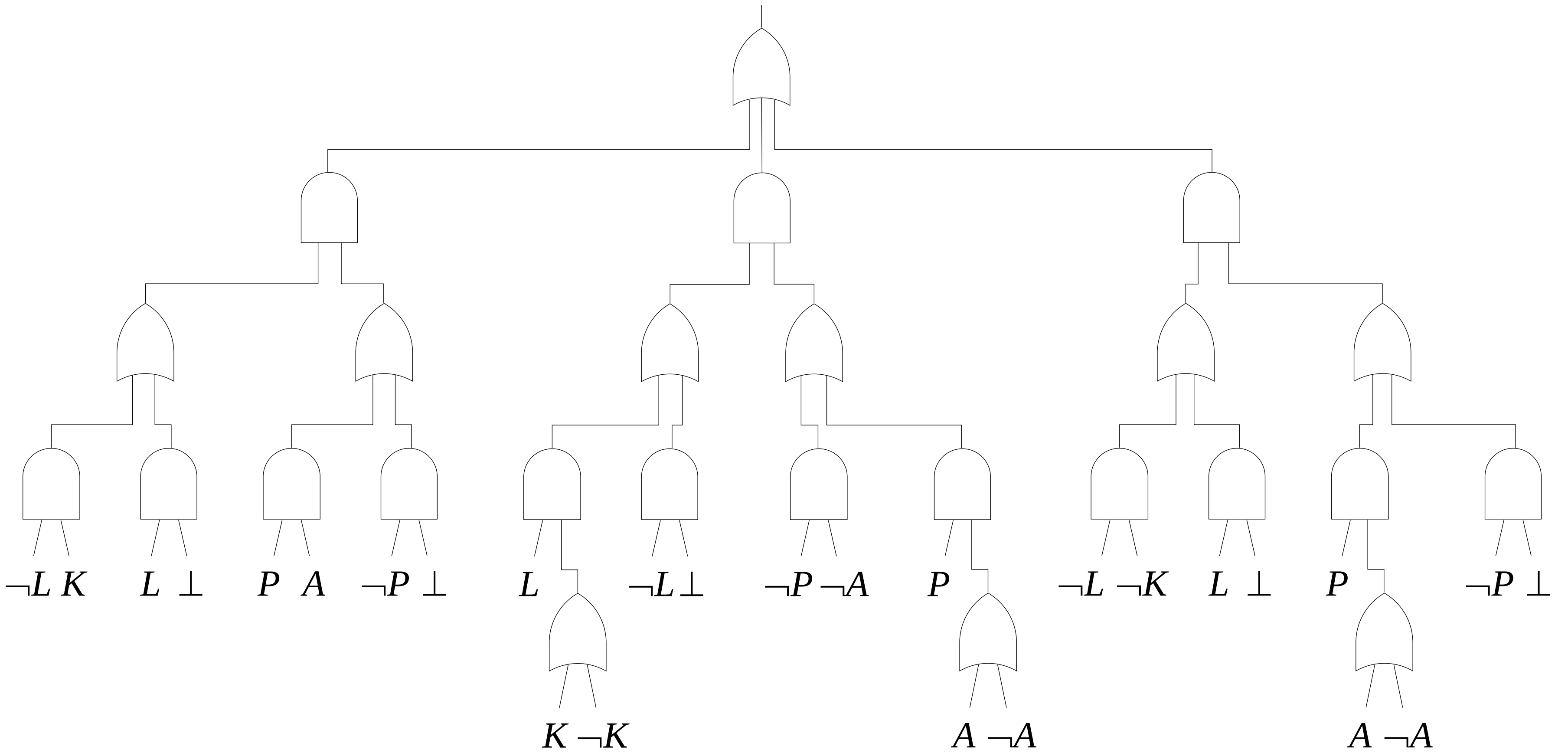
Logical Circuits

Logical Circuits

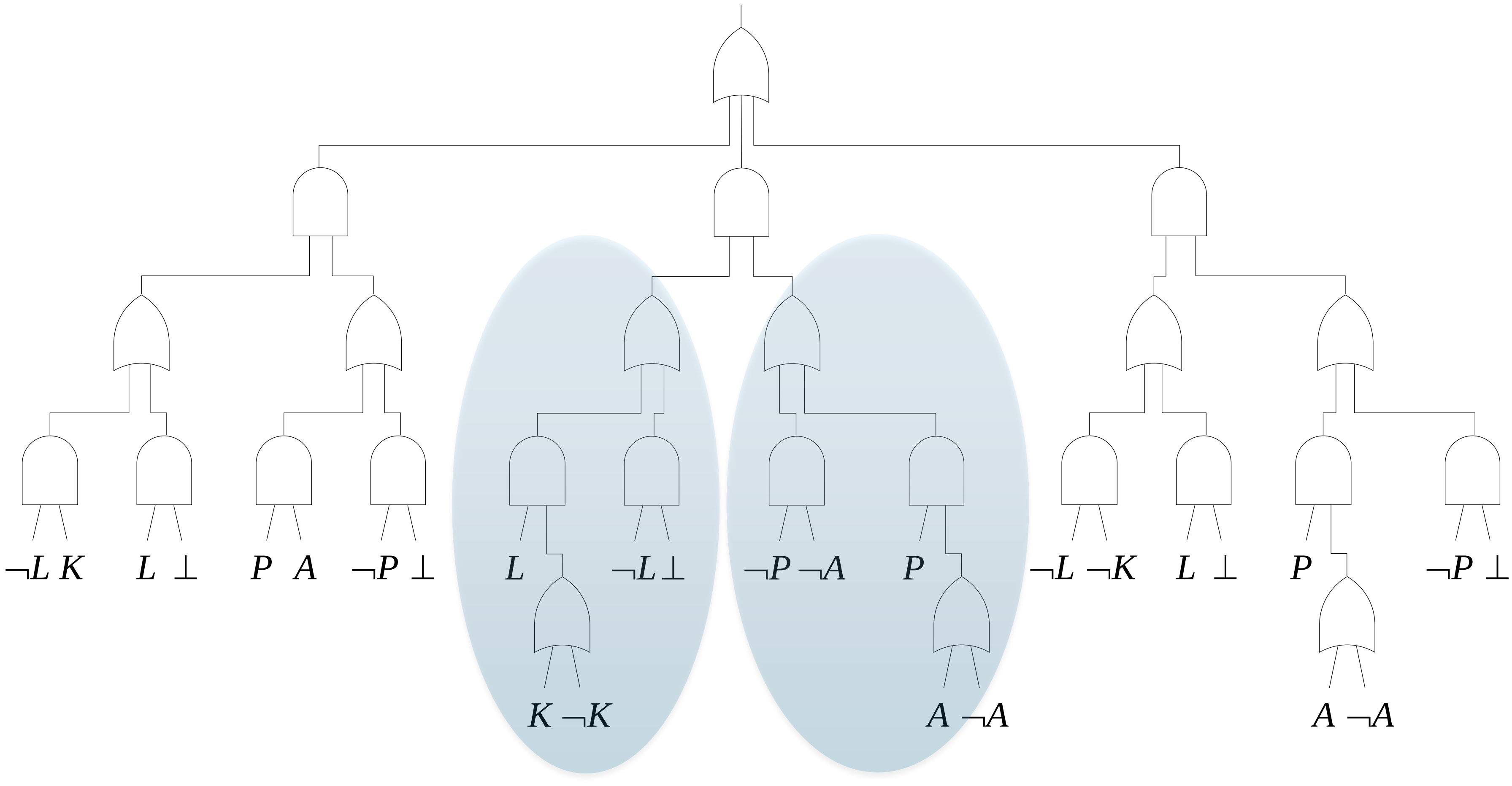
$P \vee L$
$A \Rightarrow P$
$K \Rightarrow (P \vee L)$



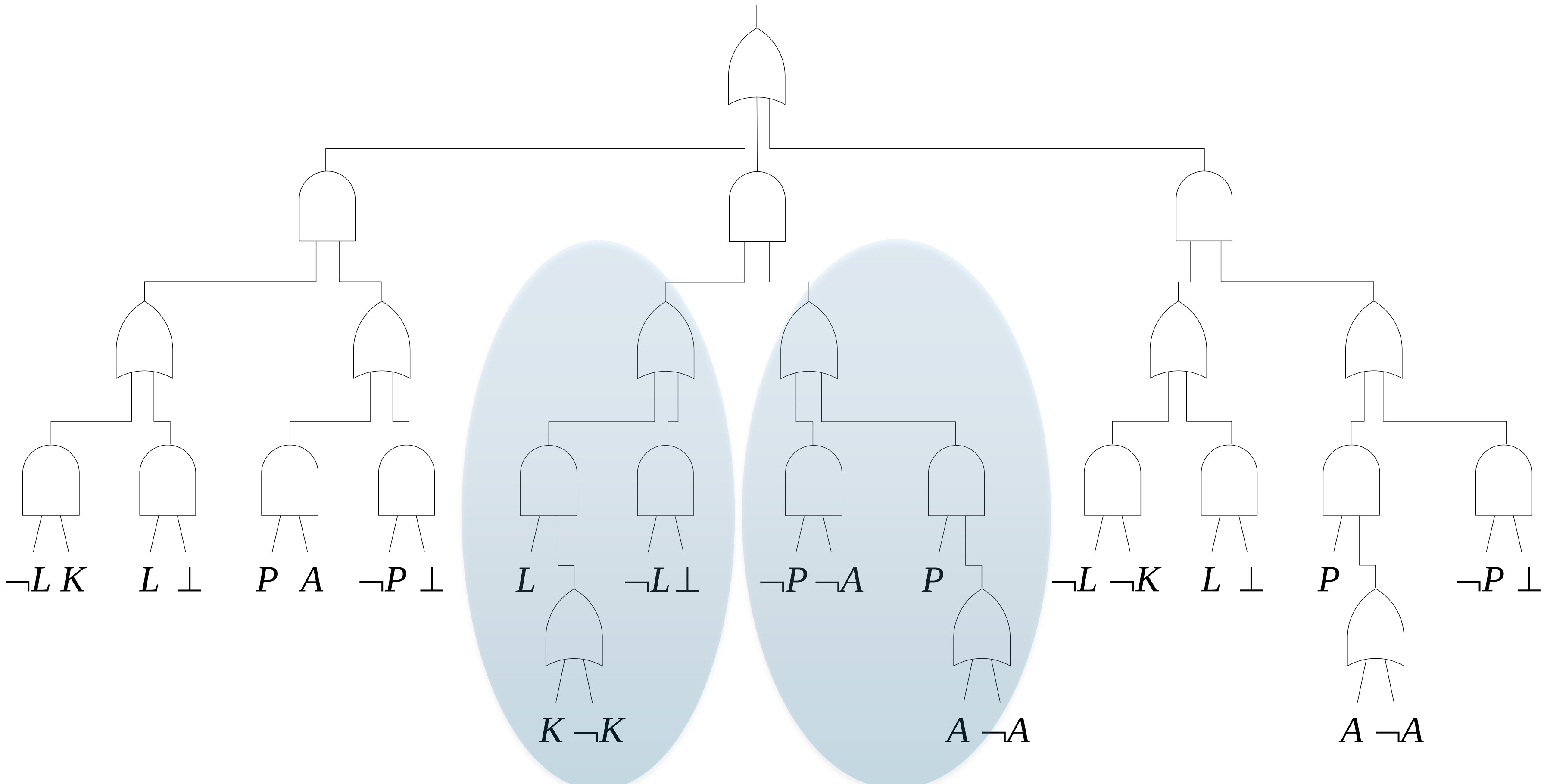
Property: Decomposability



Property: Decomposability

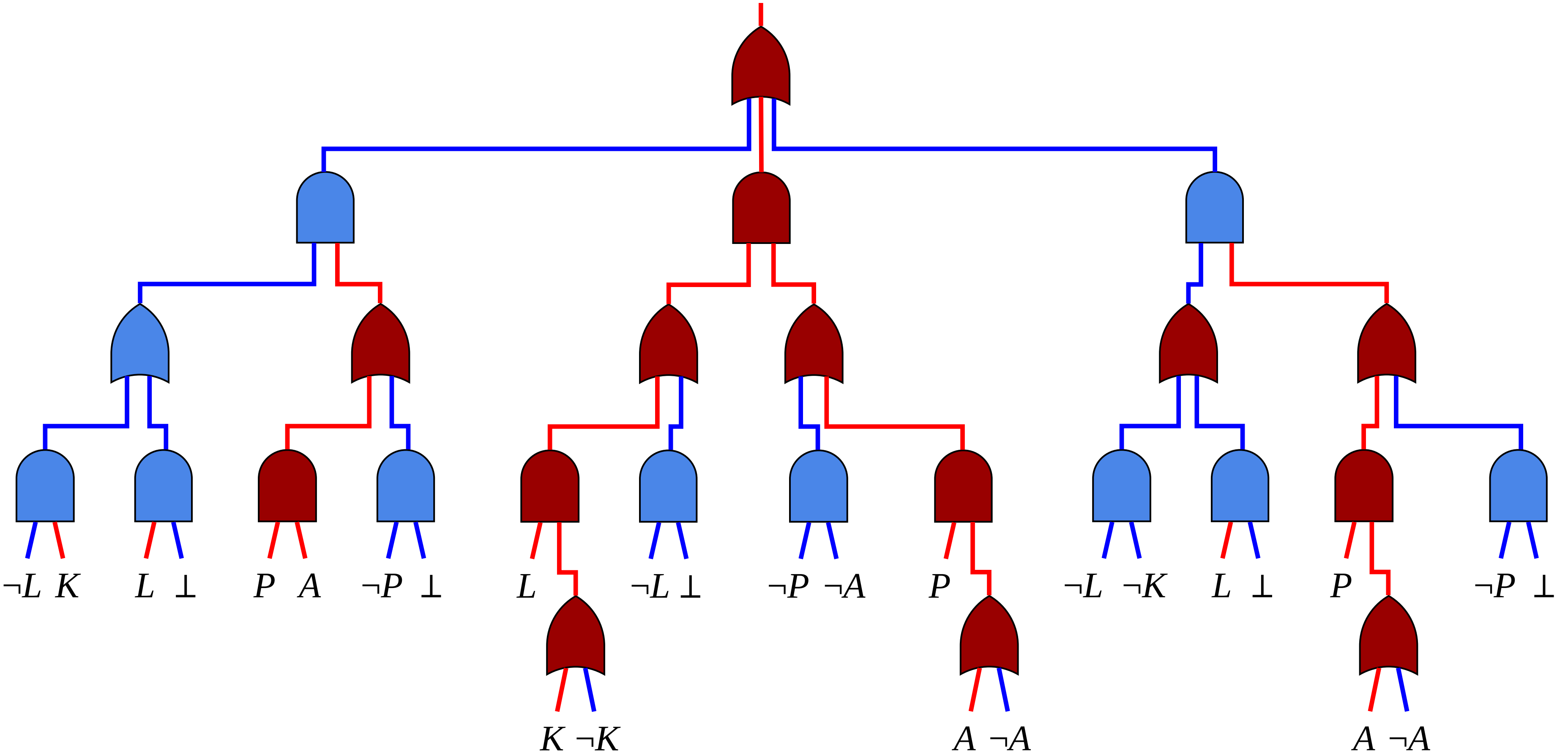


Property: Decomposability



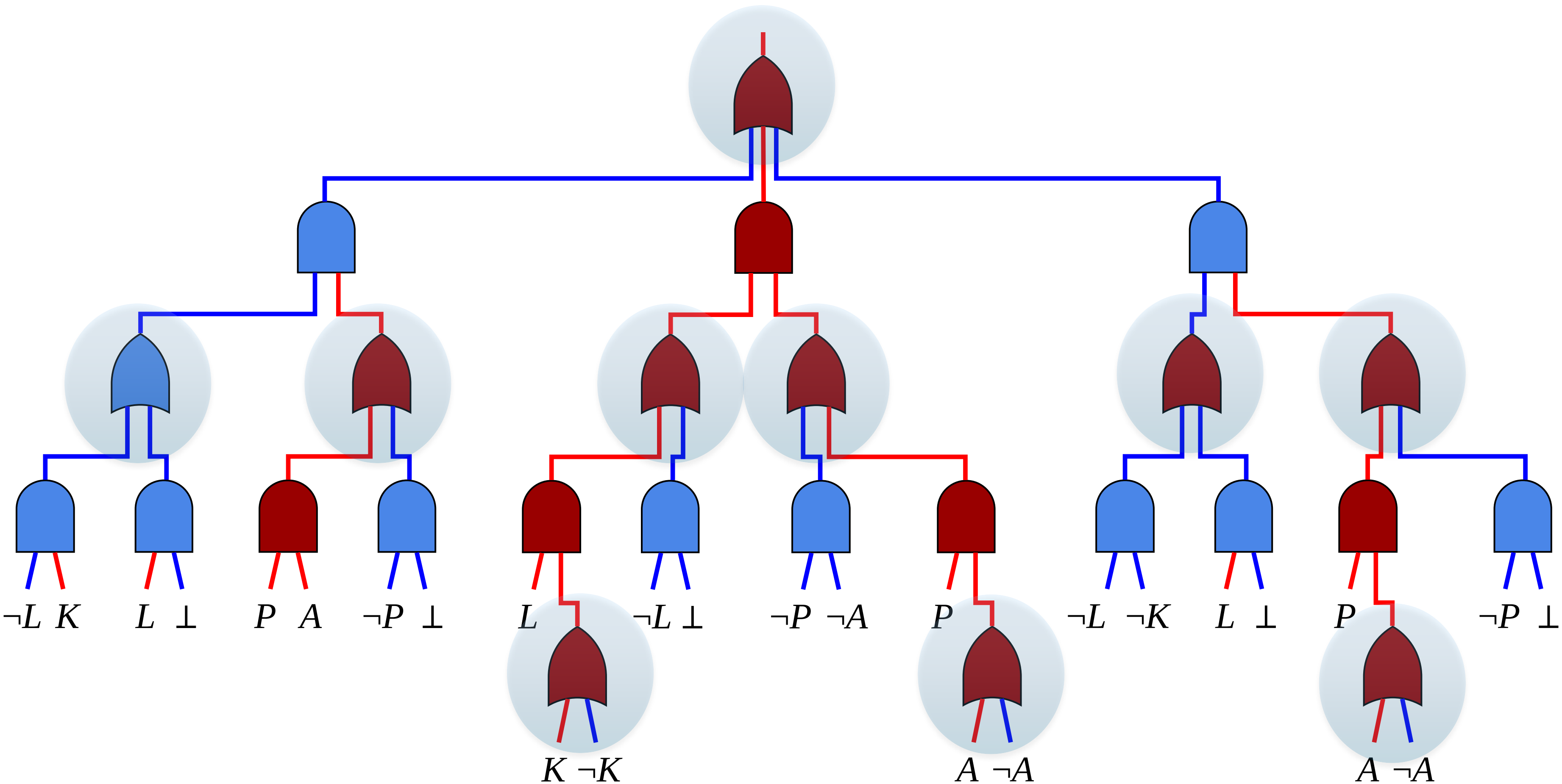
Property: AND gates have disjoint input circuits

Property: Determinism



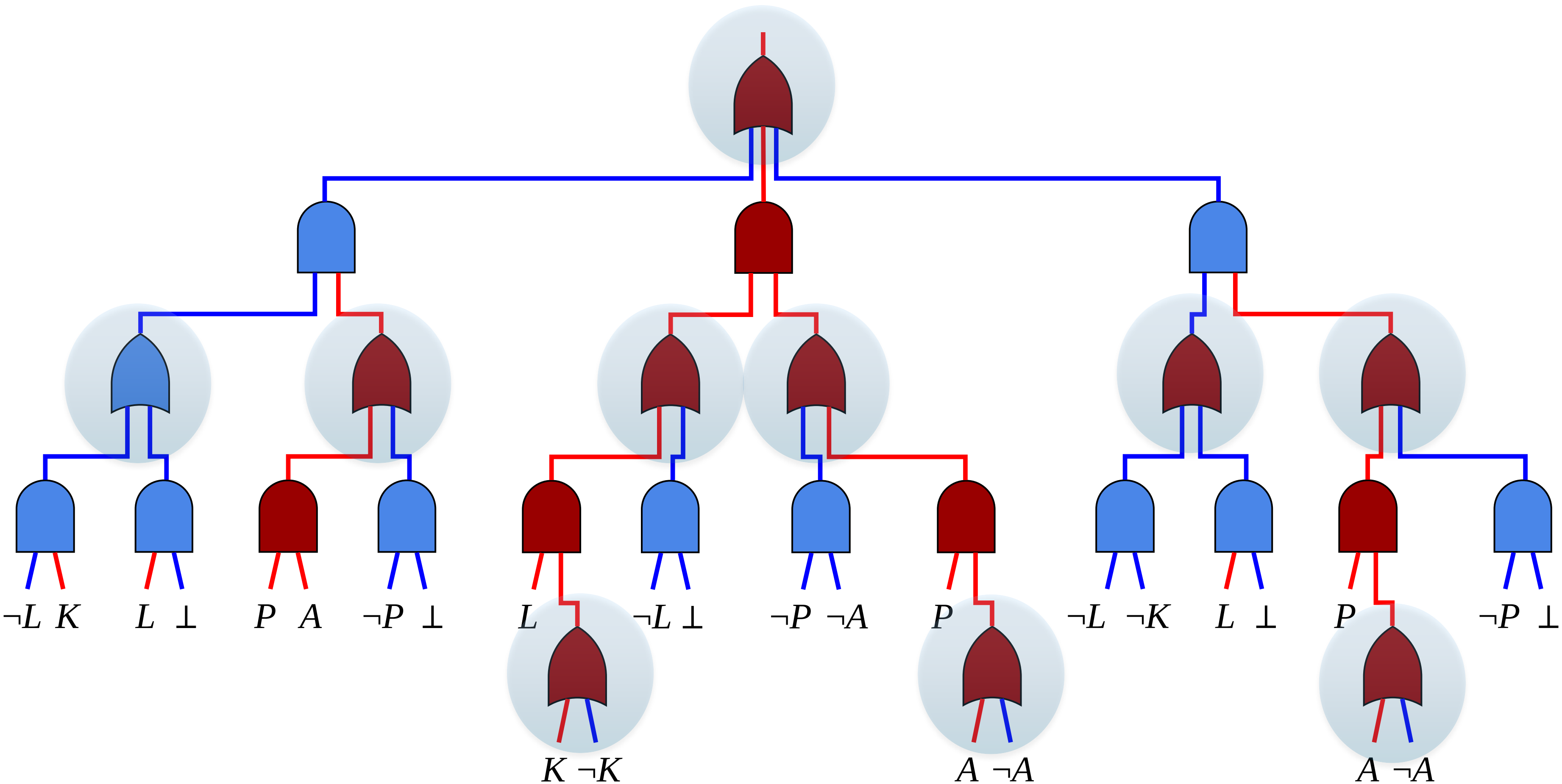
Input: L, K, P, A are true and ¬L, ¬K, ¬P, ¬A are false

Property: Determinism



Input: L, K, P, A are true and ¬L, ¬K, ¬P, ¬A are false

Property: Determinism



Input: L, K, P, A are true and ¬L, ¬K, ¬P, ¬A are false

Property: OR gates have at most one true input wire

Tractable for Logical Inference

- Is structured space empty? (SAT)
- Count size of structured space (#SAT)
- Check equivalence of spaces

Tractable for Logical Inference

- Is structured space empty? (SAT)
- Count size of structured space (#SAT)
- Check equivalence of spaces

- Algorithms linear in circuit size ☺
  (pass up, pass down, similar to backprop)

Tractable for Logical Inference

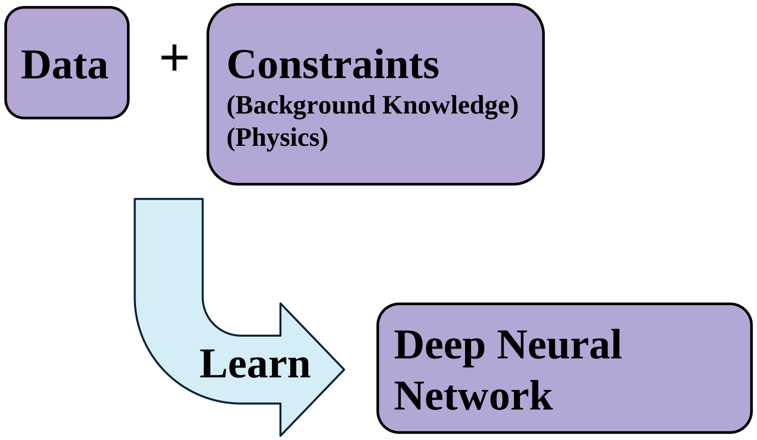
- Is structured space empty? (SAT)
- Count size of structured space (#SAT)
- Check equivalence of spaces

- Algorithms linear in circuit size ☺
(pass up, pass down, similar to backprop)

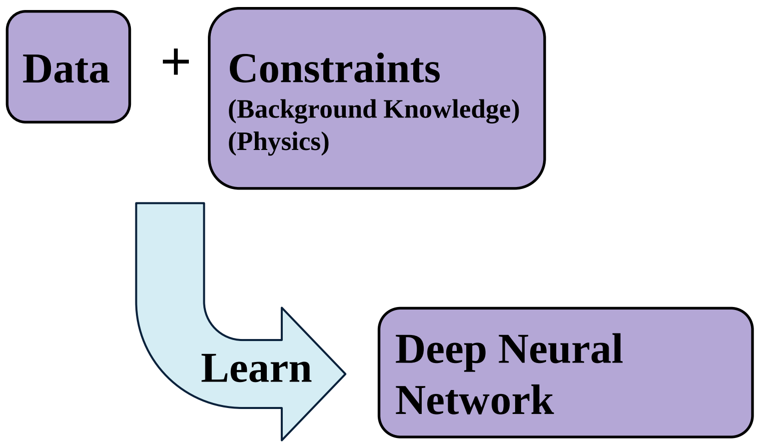
- Compilation by exhaustive SAT solvers

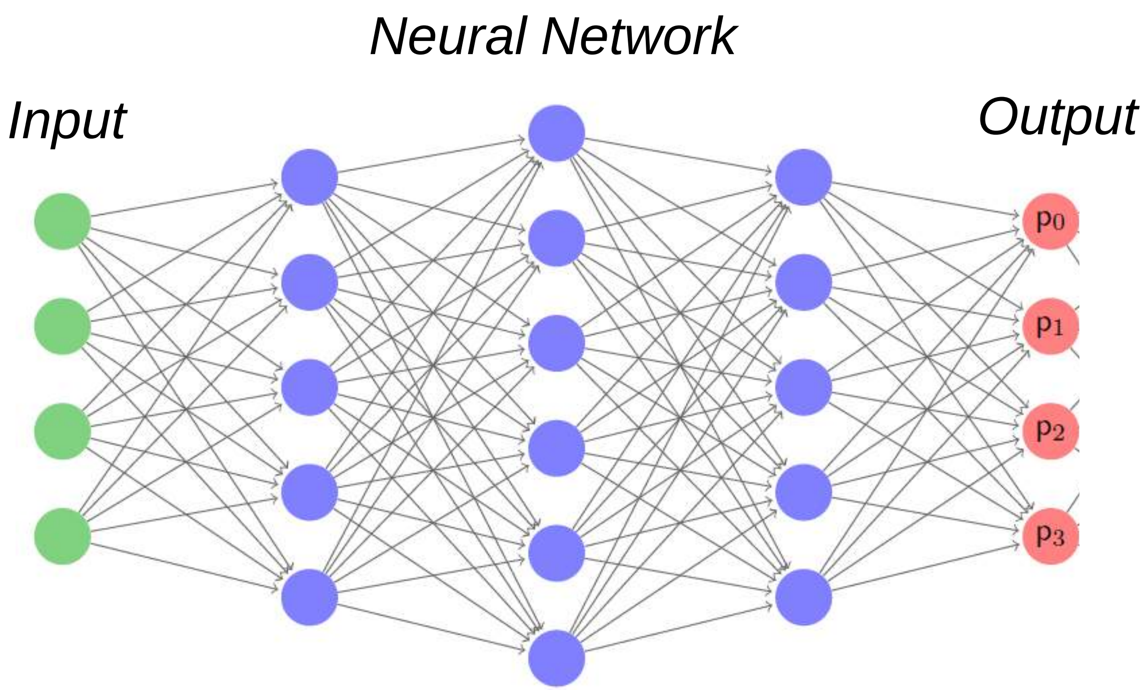
Semantic Loss for Deep Learning

Deep Structured Output Prediction

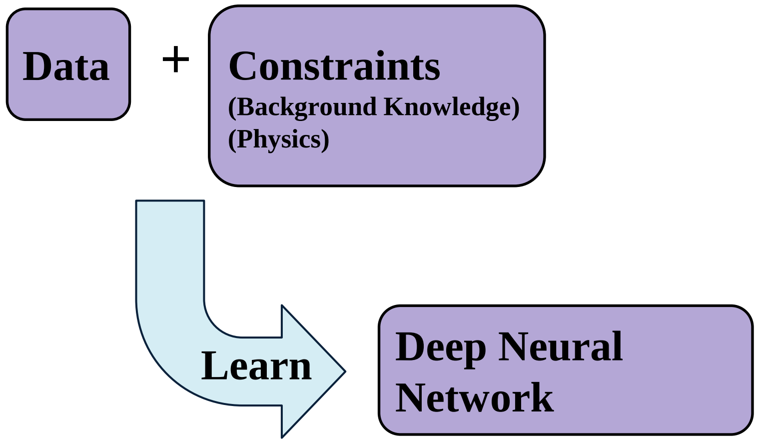


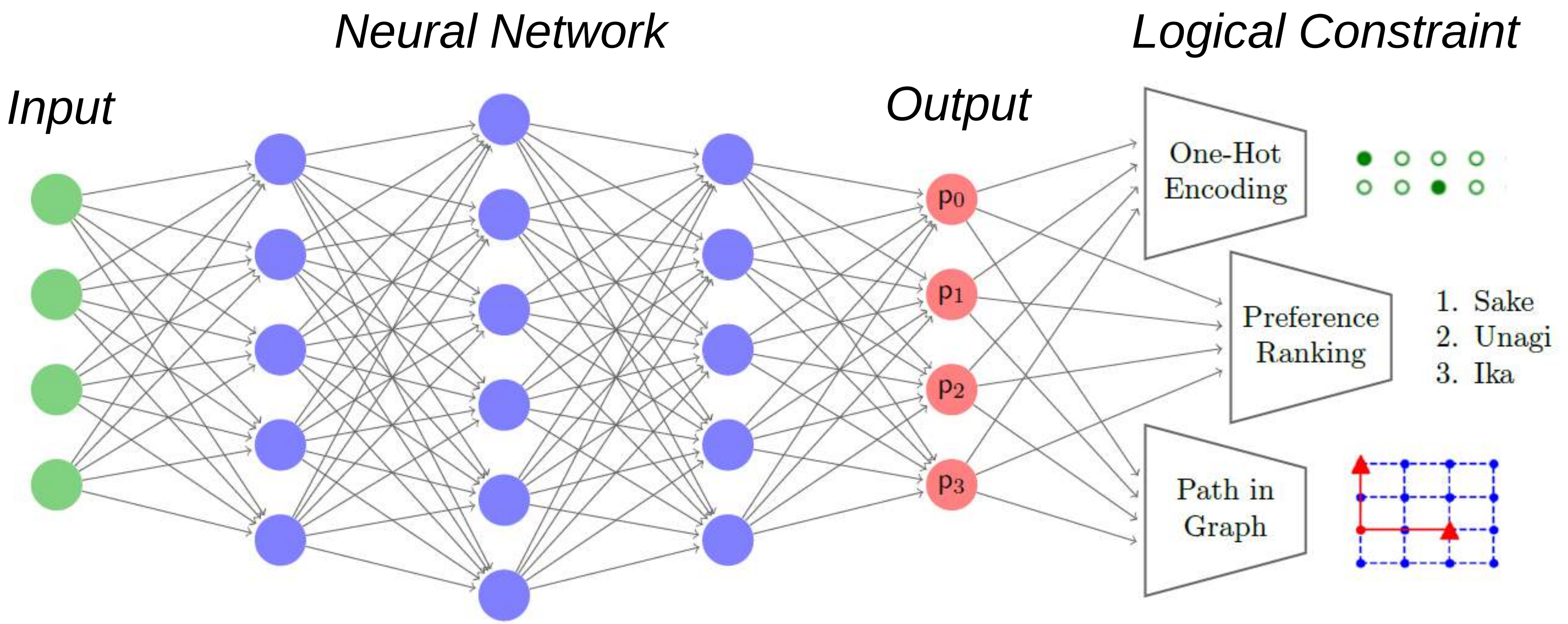
Deep Structured Output Prediction





Deep Structured Output Prediction





Semantic Loss

Semantic Loss

- Output is probability vector **p**, not logic!
  How close is output to satisfying constraint?

Semantic Loss

- Output is probability vector **p**, not logic!
  How close is output to satisfying constraint?
- Answer: Semantic loss function L(α,**p**)

Semantic Loss

- Output is probability vector **p**, not logic!
  How close is output to satisfying constraint?
- Answer: Semantic loss function L(α,**p**)
- Axioms, for example:

Semantic Loss

- Output is probability vector **p**, not logic!
  How close is output to satisfying constraint?
- Answer: Semantic loss function L(α,**p**)
- Axioms, for example:
  - If **p** is Boolean then L(**p,p**) = 0

Semantic Loss

- Output is probability vector **p**, not logic!
  How close is output 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**)

Semantic Loss

- Output is probability vector **p**, not logic!
  How close is output 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**)
- Properties:

Semantic Loss

- Output is probability vector **p**, not logic!
  How close is output to satisfying constraint?
- Answer: Semantic loss function $L(\alpha,\mathbf{p})$
- Axioms, for example:
  - If **p** is Boolean then $L(\mathbf{p},\mathbf{p}) = 0$
  - If α implies β then $L(\alpha,\mathbf{p}) \geq L(\beta,\mathbf{p})$
- Properties:
  - If α is equivalent to β then $L(\alpha,\mathbf{p}) = L(\beta,\mathbf{p})$

Semantic Loss

- Output is probability vector $\mathbf{p}$, not logic!
  How close is output to satisfying constraint?
- Answer: Semantic loss function $L(\alpha,\mathbf{p})$
- Axioms, for example:
  – If $\mathbf{p}$ is Boolean then $L(\mathbf{p},\mathbf{p}) = 0$
  – If α implies β then $L(\alpha,\mathbf{p}) \geq L(\beta,\mathbf{p})$
- Properties:
  – If α is equivalent to β then $L(\alpha,\mathbf{p}) = L(\beta,\mathbf{p})$
  – If $\mathbf{p}$ is Boolean and satisfies α then $L(\alpha,\mathbf{p}) = 0$

Semantic Loss

- Output is probability vector **p**, not logic!
  How close is output 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**)

SEMANTIC
Loss!

- 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: Definition

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

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

Probability of getting **x** after
flipping coins with prob. **p**

Probability of satisfying α after
flipping coins with prob. **p**

How to Compute Semantic Loss?

How to Compute Semantic Loss?

- In general: #P-hard ☹

How to Compute Semantic Loss?

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

How to Compute Semantic Loss?

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

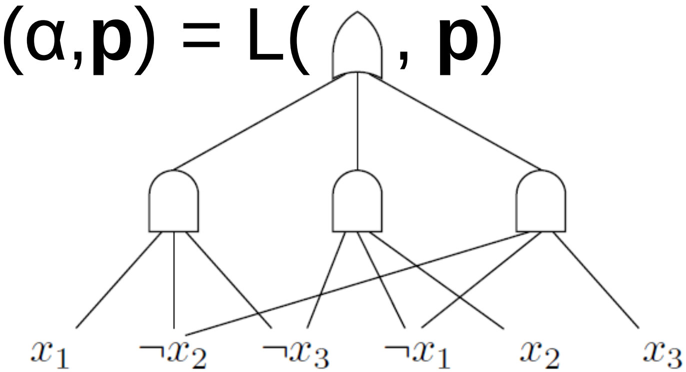
$$\begin{cases} \boldsymbol{x_1} \lor \boldsymbol{x_2} \lor \boldsymbol{x_3} \\ \neg \boldsymbol{x_1} \lor \neg \boldsymbol{x_2} \\ \neg \boldsymbol{x_2} \lor \neg \boldsymbol{x_3} \\ \neg \boldsymbol{x_1} \lor \neg \boldsymbol{x_3} \end{cases}$$

How to Compute Semantic Loss?

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

$$\begin{cases} x_1 \lor x_2 \lor x_3 \\ \neg x_1 \lor \neg x_2 \\ \neg x_2 \lor \neg x_3 \\ \neg x_1 \lor \neg x_3 \end{cases}$$

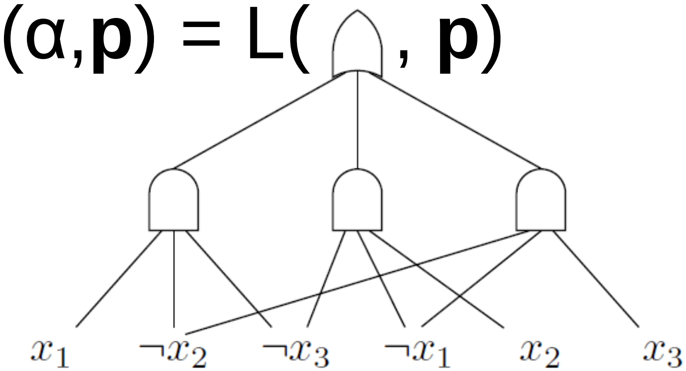
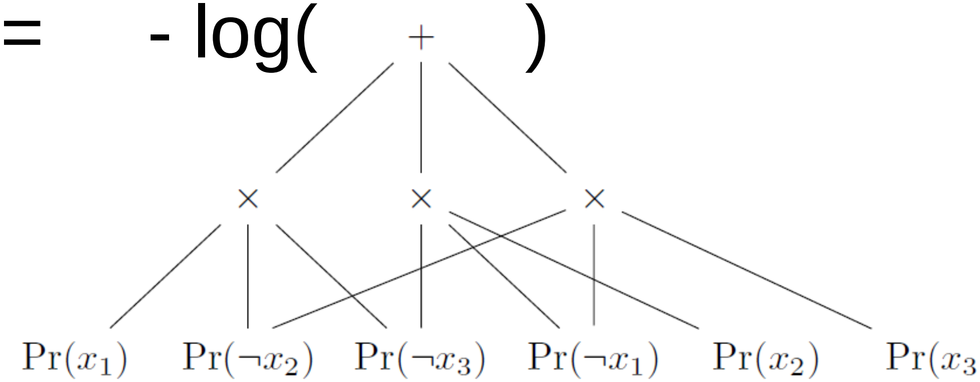
L(α,**p**) = L(, **p**)



How to Compute Semantic Loss?

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

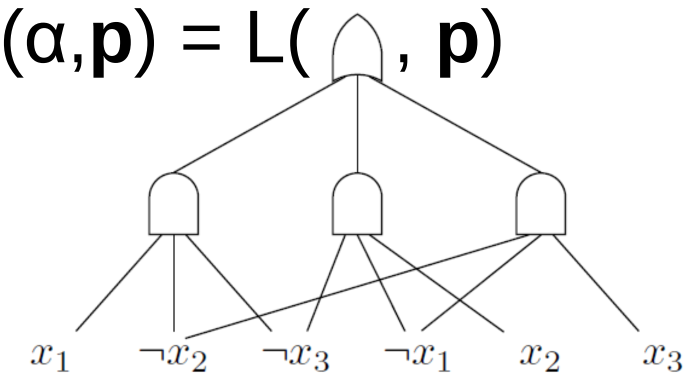
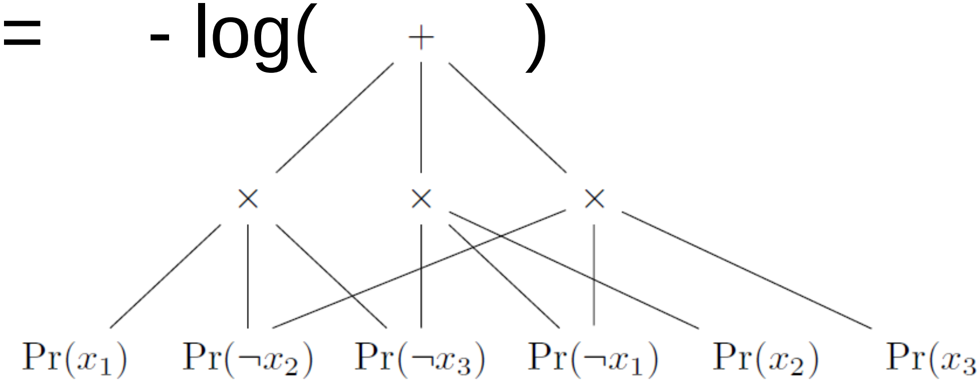
$$\begin{cases} \boldsymbol{x_1} \lor \boldsymbol{x_2} \lor \boldsymbol{x_3} \\ \neg \boldsymbol{x_1} \lor \neg \boldsymbol{x_2} \\ \neg \boldsymbol{x_2} \lor \neg \boldsymbol{x_3} \\ \neg \boldsymbol{x_1} \lor \neg \boldsymbol{x_3} \end{cases}$$

L(α,**p**) = L( , **p**) = - log()

How to Compute Semantic Loss?

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

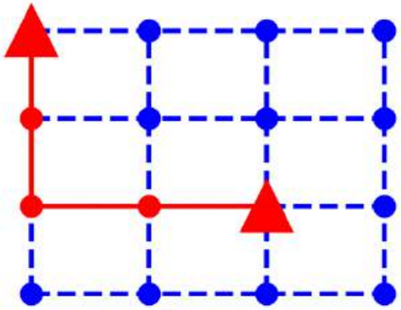
$$\begin{cases} x_1 \lor x_2 \lor x_3 \\ \neg x_1 \lor \neg x_2 \\ \neg x_2 \lor \neg x_3 \\ \neg x_1 \lor \neg x_3 \end{cases}$$

L(α,**p**) = L( , **p**) = - log()

- *Why?* Decomposability and determinism!

Supervised Learning

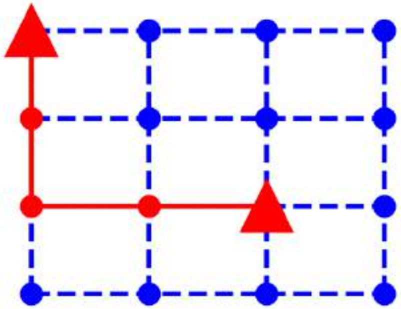
- Predict shortest paths
- Add semantic loss to objective



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

Supervised Learning

- Predict shortest paths
- Add semantic loss to objective

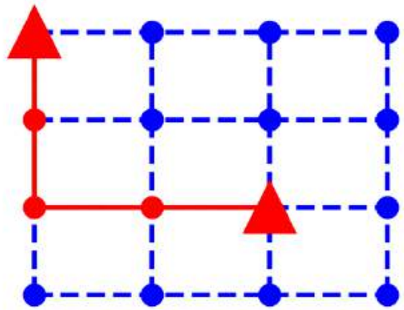


Test accuracy %	Coherent	Incoherent	Constraint
5-layer MLP	5.62	**85.91**	6.99
Semantic loss	**28.51**	83.14	**69.89**

Is output a path?

Supervised Learning

- Predict shortest paths
- Add semantic loss to objective



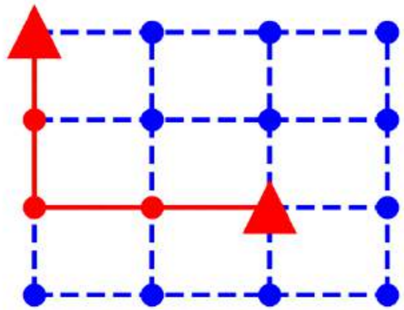
Test accuracy %	Coherent	Incoherent	Constraint
5-layer MLP	5.62	**85.91**	6.99
Semantic loss	**28.51**	83.14	**69.89**

Does output have true edges?

Is output a path?

Supervised Learning

- Predict shortest paths
- Add semantic loss to objective



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

Is output the true path?

Does output have true edges?

Is output a path?

Supervised Learning

- Predict sushi preferences
- Add semantic loss to objective

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

Test accuracy %	Coherent	Incoherent	Constraint
3-layer MLP	1.01	**75.78**	2.72
Semantic loss	**13.59**	72.43	**55.28**

Supervised Learning

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

- Predict sushi preferences
- Add semantic loss to objective

Test accuracy %	Coherent	Incoherent	Constraint
3-layer MLP	1.01	**75.78**	2.72
Semantic loss	**13.59**	72.43	**55.28**

Is output a ranking?

Supervised Learning

- Predict sushi preferences
- Add semantic loss to objective

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

Test accuracy %	Coherent	Incoherent	Constraint
3-layer MLP	1.01	**75.78**	2.72
Semantic loss	**13.59**	72.43	**55.28**

Does output correctly rank individual sushis?

Is output a ranking?

Supervised Learning

- Predict sushi preferences
- Add semantic loss to objective

rank	sushi
1	fatty tuna
2	sea urchin
3	salmon roe
4	shrimp
5	tuna
6	squid
7	tuna roll
8	see eel
9	egg
10	cucumber roll

Test accuracy %	Coherent	Incoherent	Constraint
3-layer MLP	1.01	**75.78**	2.72
Semantic loss	**13.59**	72.43	**55.28**

Is output the true ranking?

Does output correctly rank individual sushis?

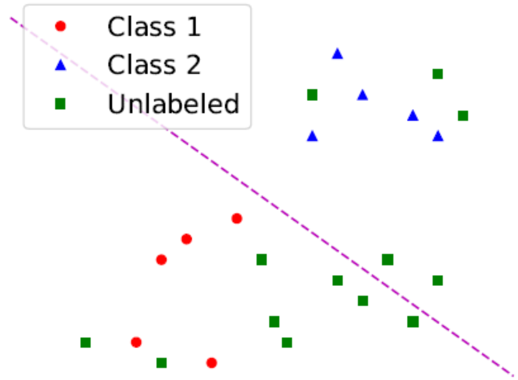
Is output a ranking?

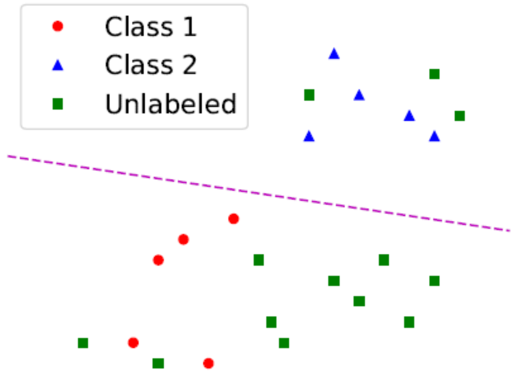
Semi-Supervised Learning

- Unlabeled data must have some label

Semi-Supervised Learning

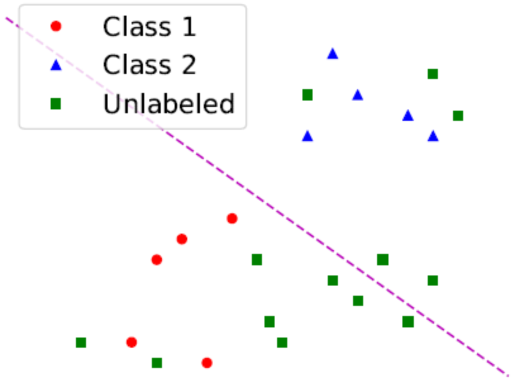
- Unlabeled data must have some label

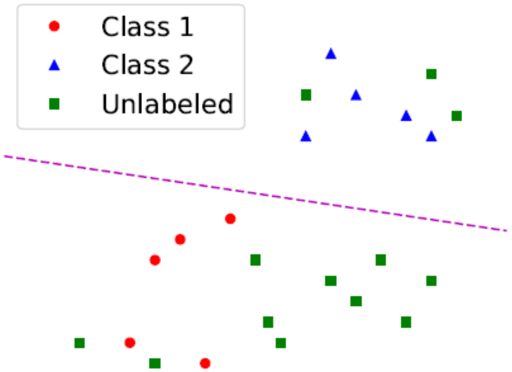




Semi-Supervised Learning

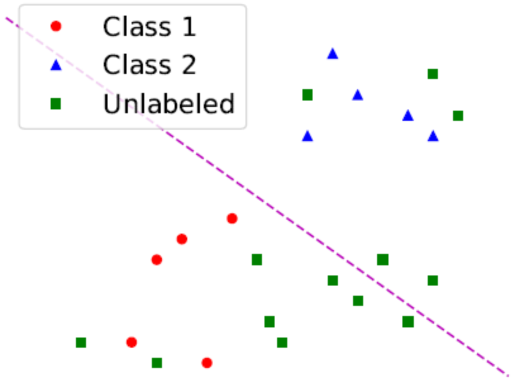
- Unlabeled data must have some label

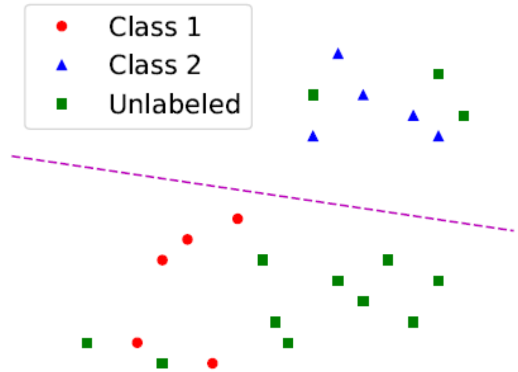




Semi-Supervised Learning

- Unlabeled data must have some label

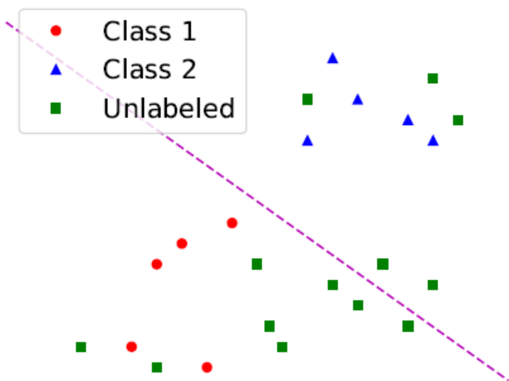


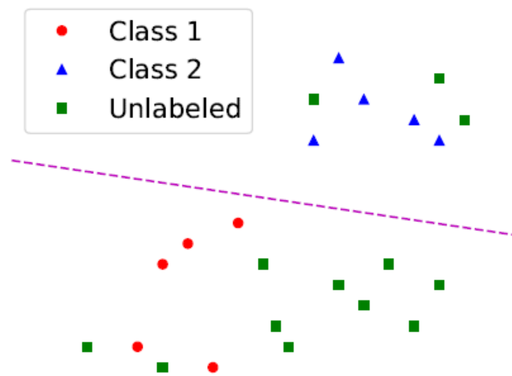


- Low semantic loss with exactly-one constraint

Semi-Supervised Learning

- Unlabeled data must have some label





- Low semantic loss with exactly-one constraint

$$\mathrm{L}^{s}(\text{exactly-one}, \mathsf{p}) \propto -\log \sum_{i=1}^{n} \mathsf{p}_i \prod_{j=1, j \neq i}^{n} (1 - \mathsf{p}_j)$$

MNIST

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)

FASHION

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

(a) Confidently Correct

(b) Unconfidently Correct

(c) Unconfidently Incorrect

(d) Confidently Incorrect

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

Semantic Loss Conclusions

- Cares about *meaning* not syntax
- Elegant axiomatic approach

Semantic Loss Conclusions

- Cares about *meaning* not syntax
- Elegant axiomatic approach

- If you have complex output constraints

  Use logical circuits to enforce them

Semantic Loss Conclusions

- Cares about *meaning* not syntax
- Elegant axiomatic approach

- If you have complex output constraints

  Use logical circuits to enforce them

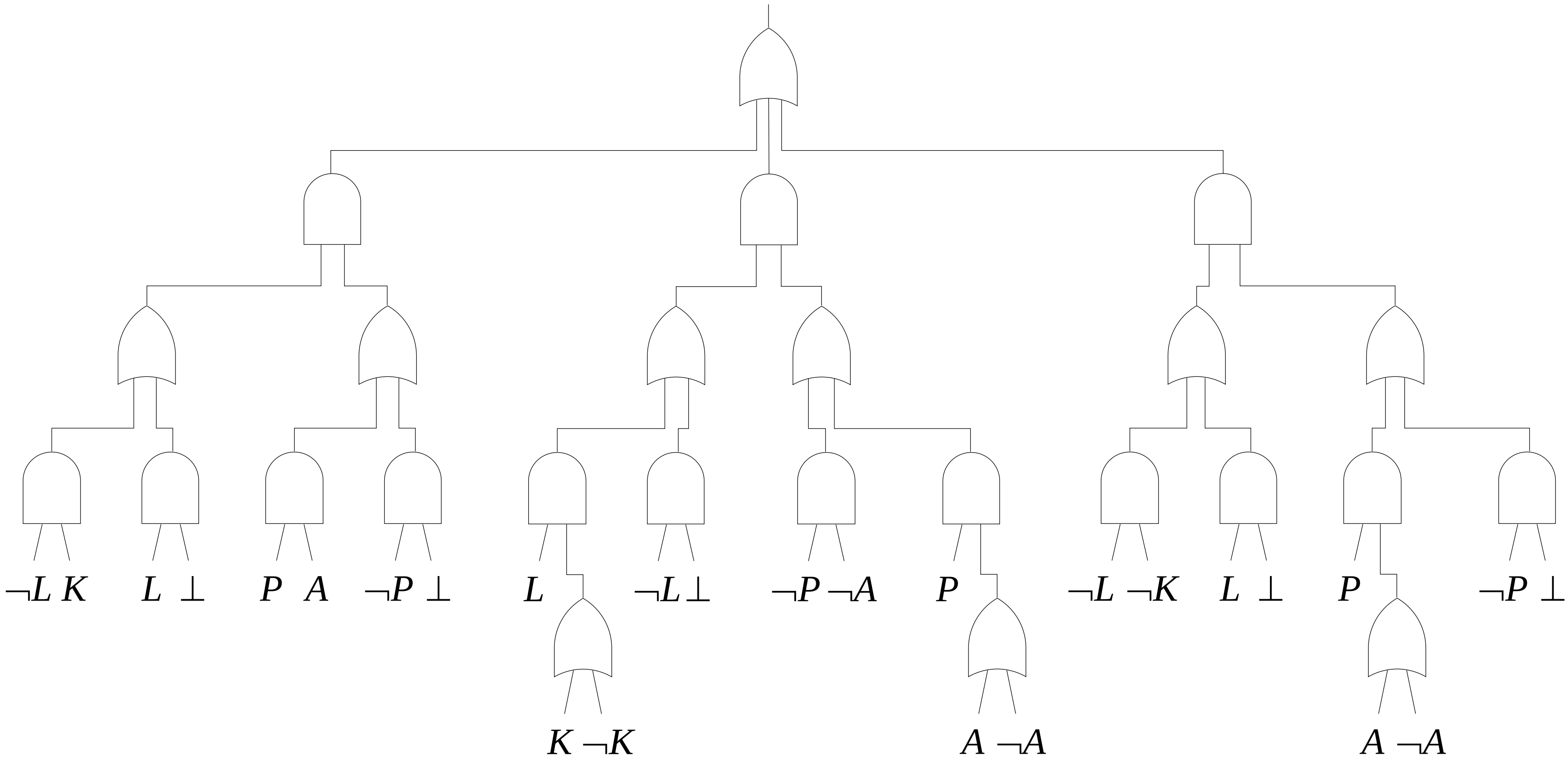
  If you have unlabeled data (no constraints)

  Get a lot of signal by minimizing
  semantic loss of exactly-one

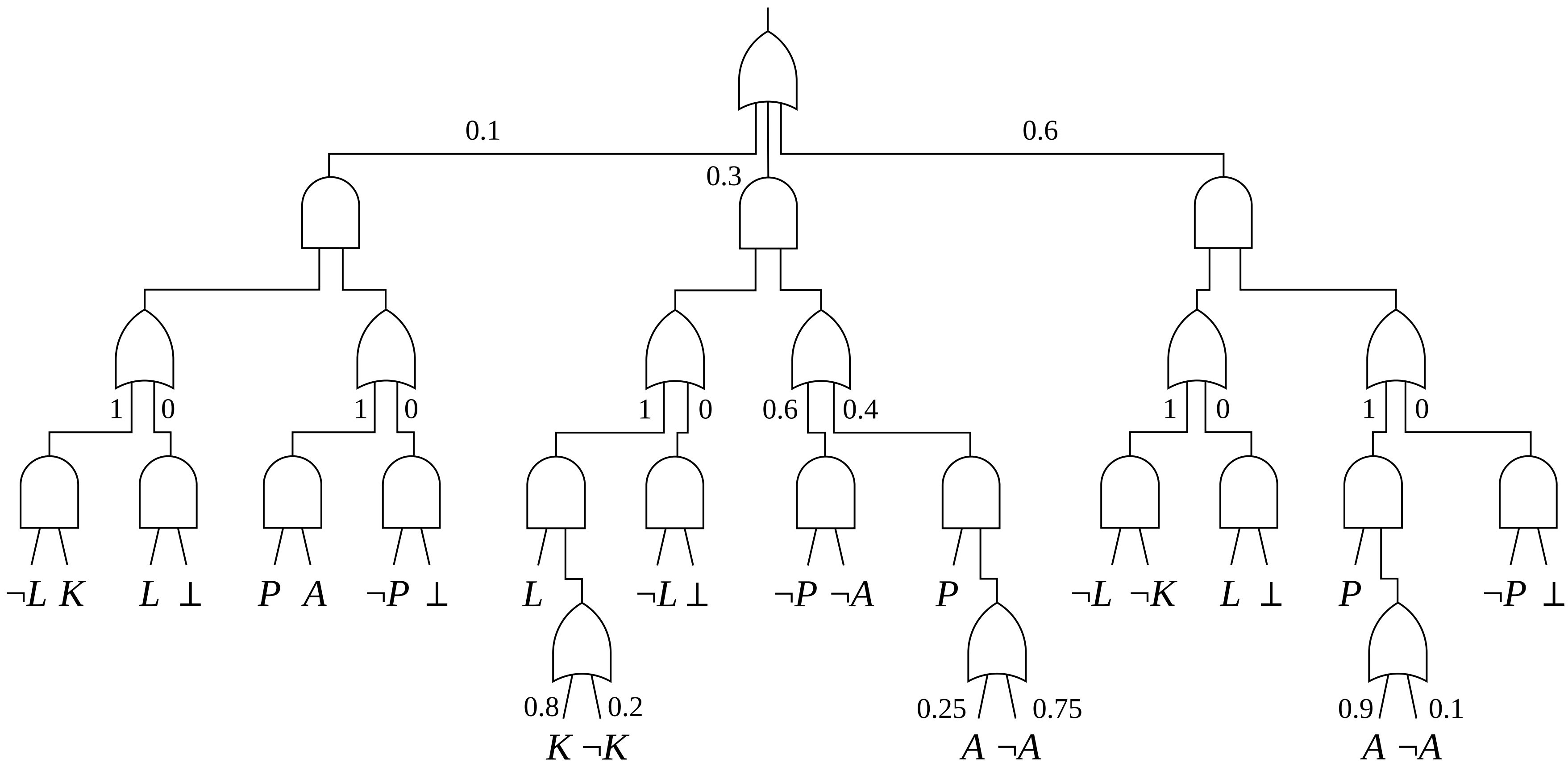
Probabilistic Circuits

Logical Circuits

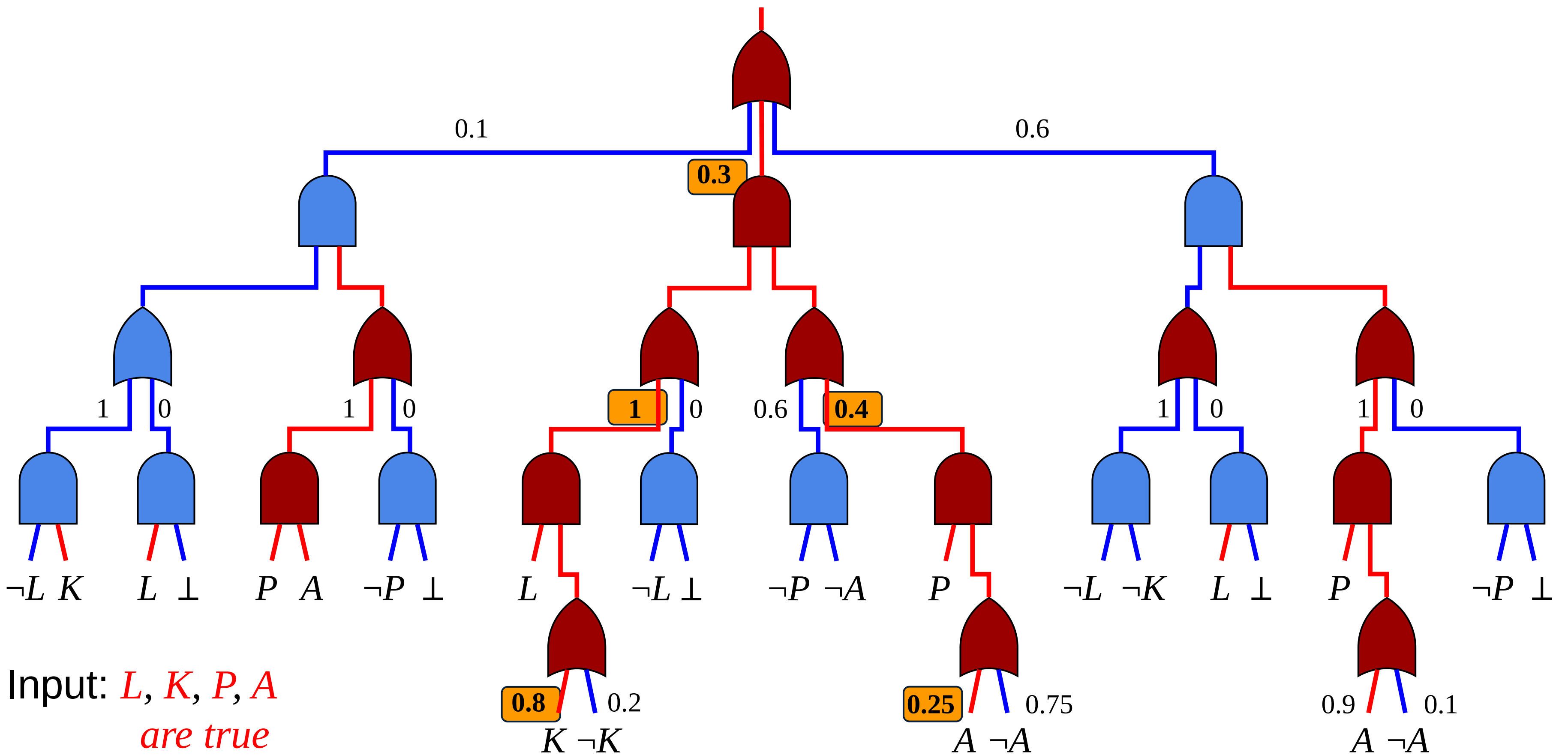
$P \vee L$
$A \Rightarrow P$
$K \Rightarrow (P \vee L)$



PSDD: Probabilistic SDD



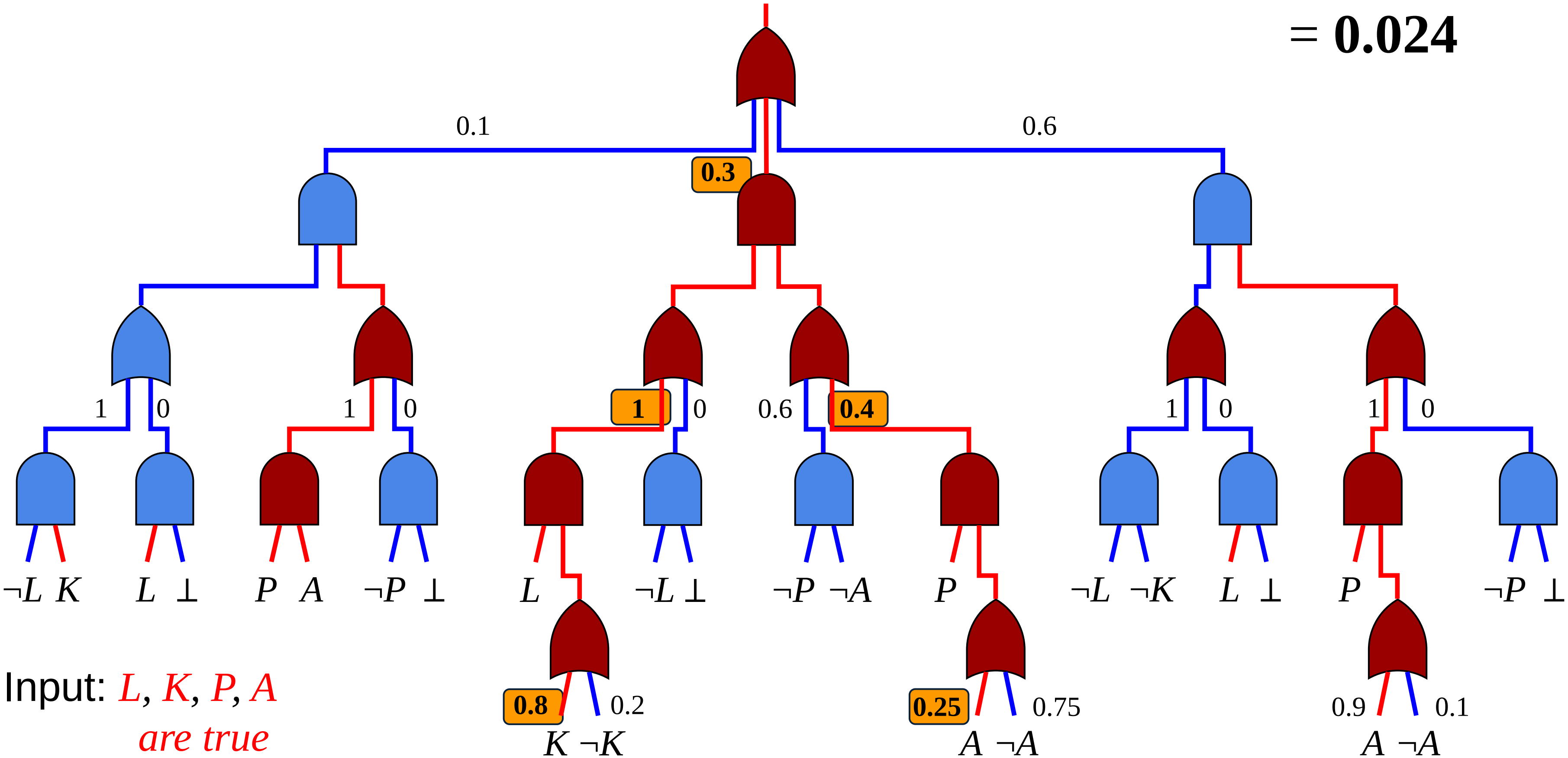
PSDD:
Probabilistic SDD



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

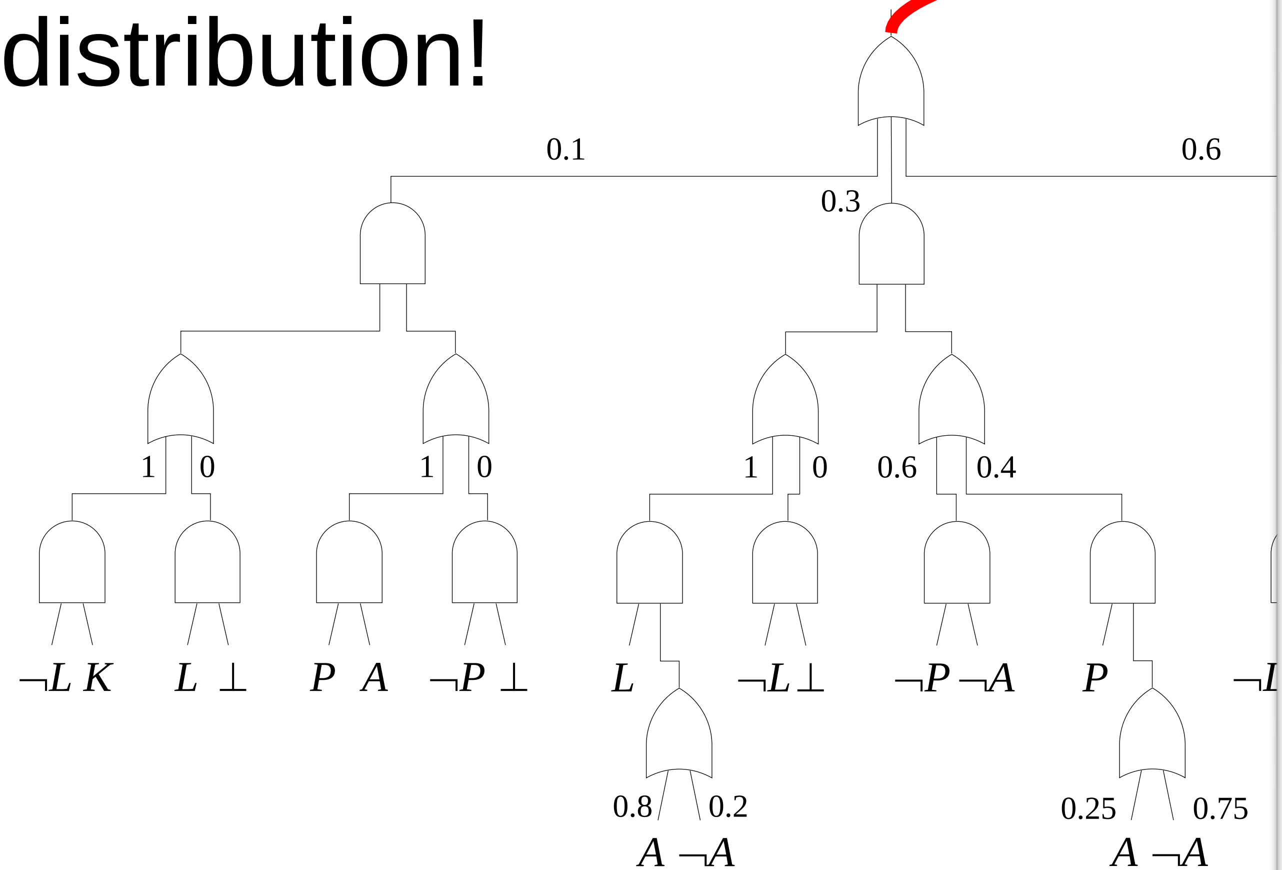
PSDD: Probabilistic SDD

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



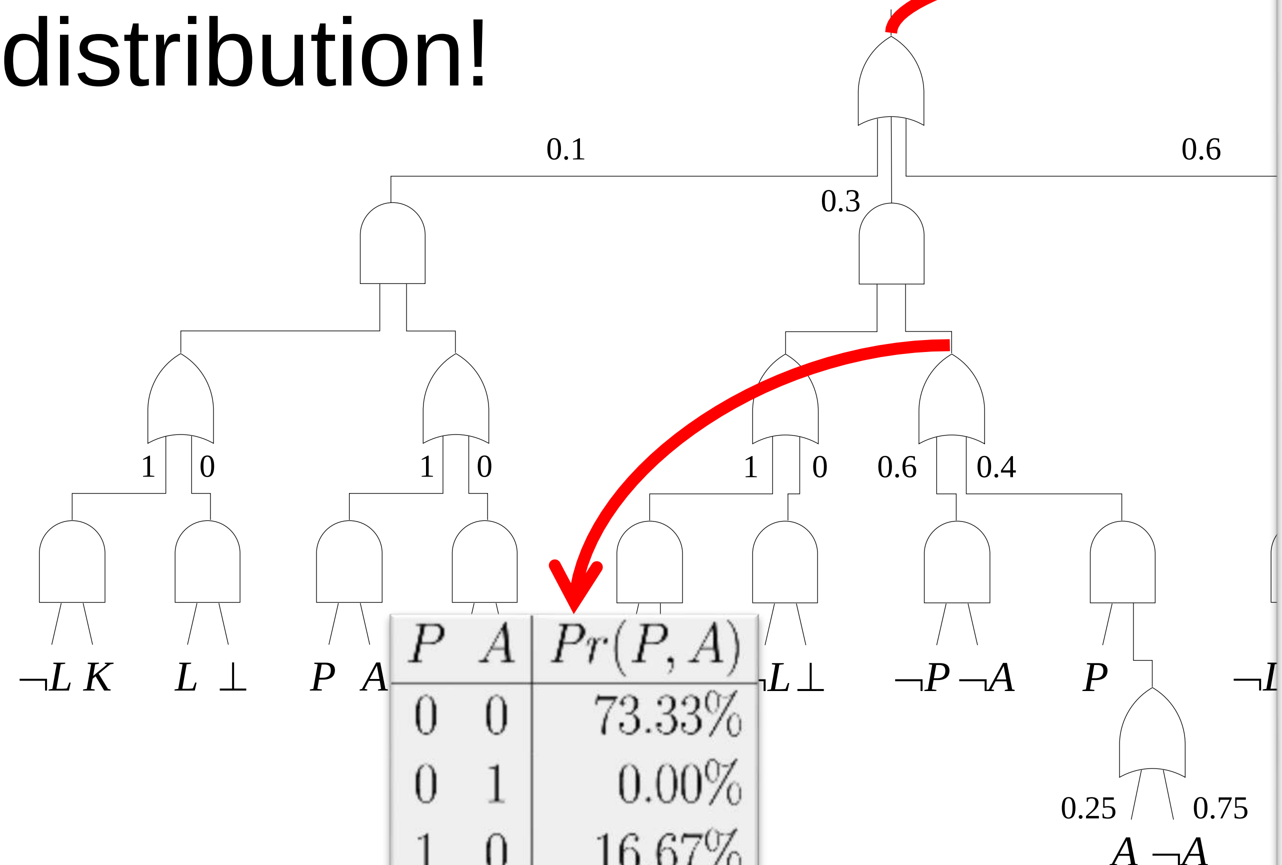
Input: *L*, *K*, *P*, *A* *are true*

PSDD nodes induce a normalized distribution!



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%

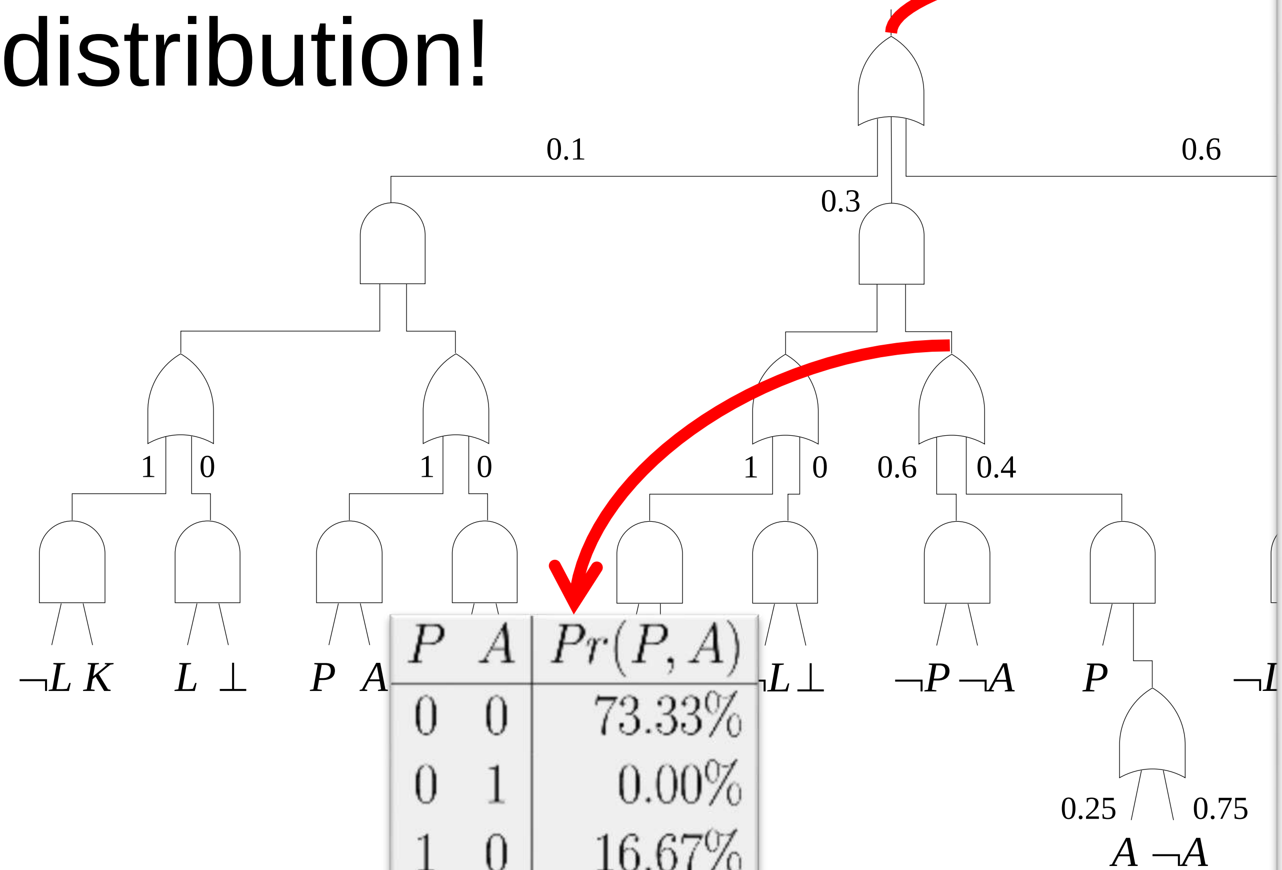
PSDD nodes induce a normalized distribution!



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%

PSDD nodes induce 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

Tractable for Probabilistic Inference

- **MAP inference**: Find most-likely assignment
  (otherwise NP-complete)
- Computing **conditional probabilities** Pr(x|y)
  (otherwise PP-complete)
- **Sample** from Pr(x|y)

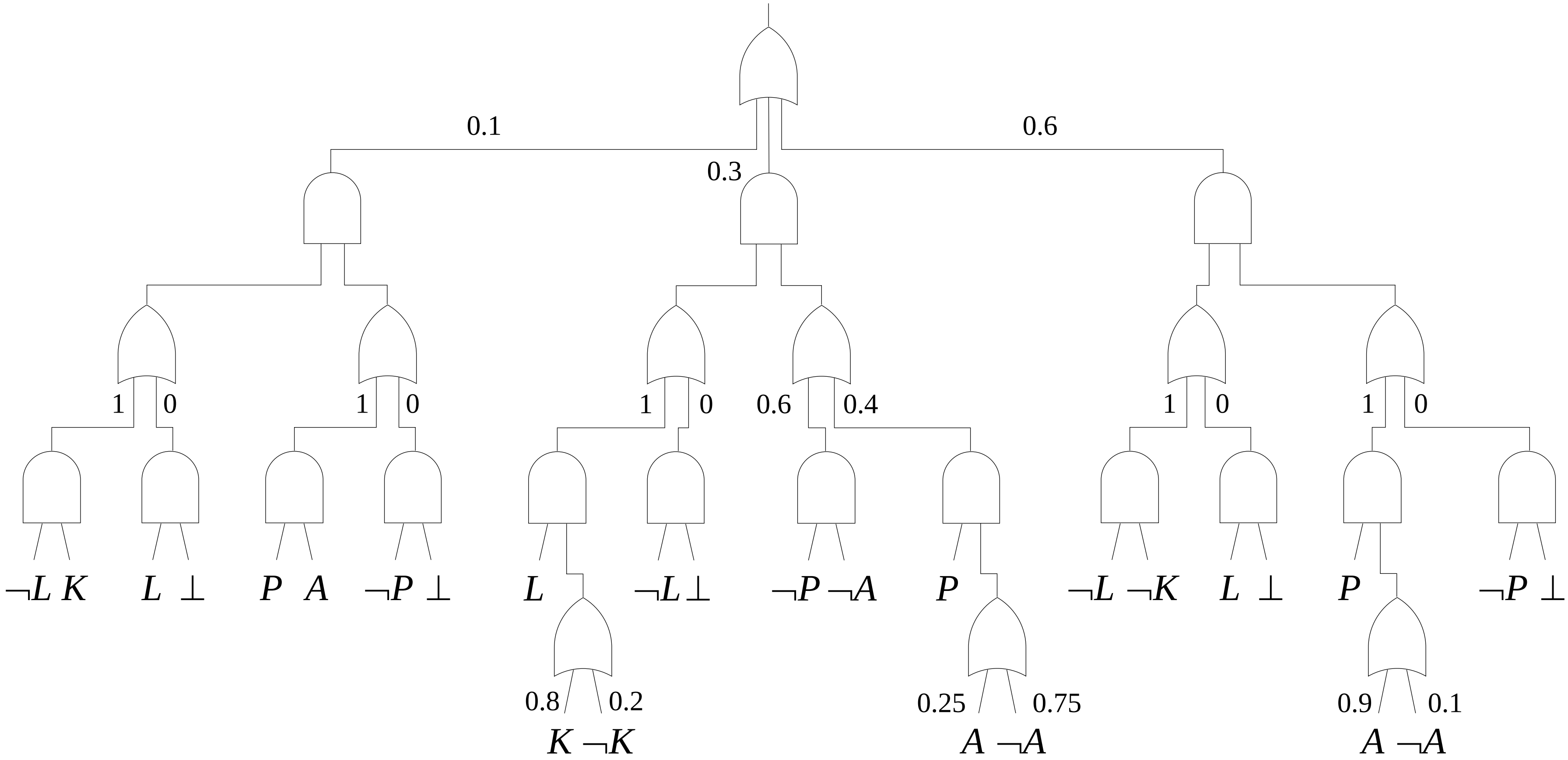
Tractable for Probabilistic Inference

- **MAP inference**: Find most-likely assignment (otherwise NP-complete)
- Computing **conditional probabilities** Pr(x|y) (otherwise PP-complete)
- **Sample** from Pr(x|y)

Algorithms linear in circuit size ☺
(pass up, pass down, similar to backprop)

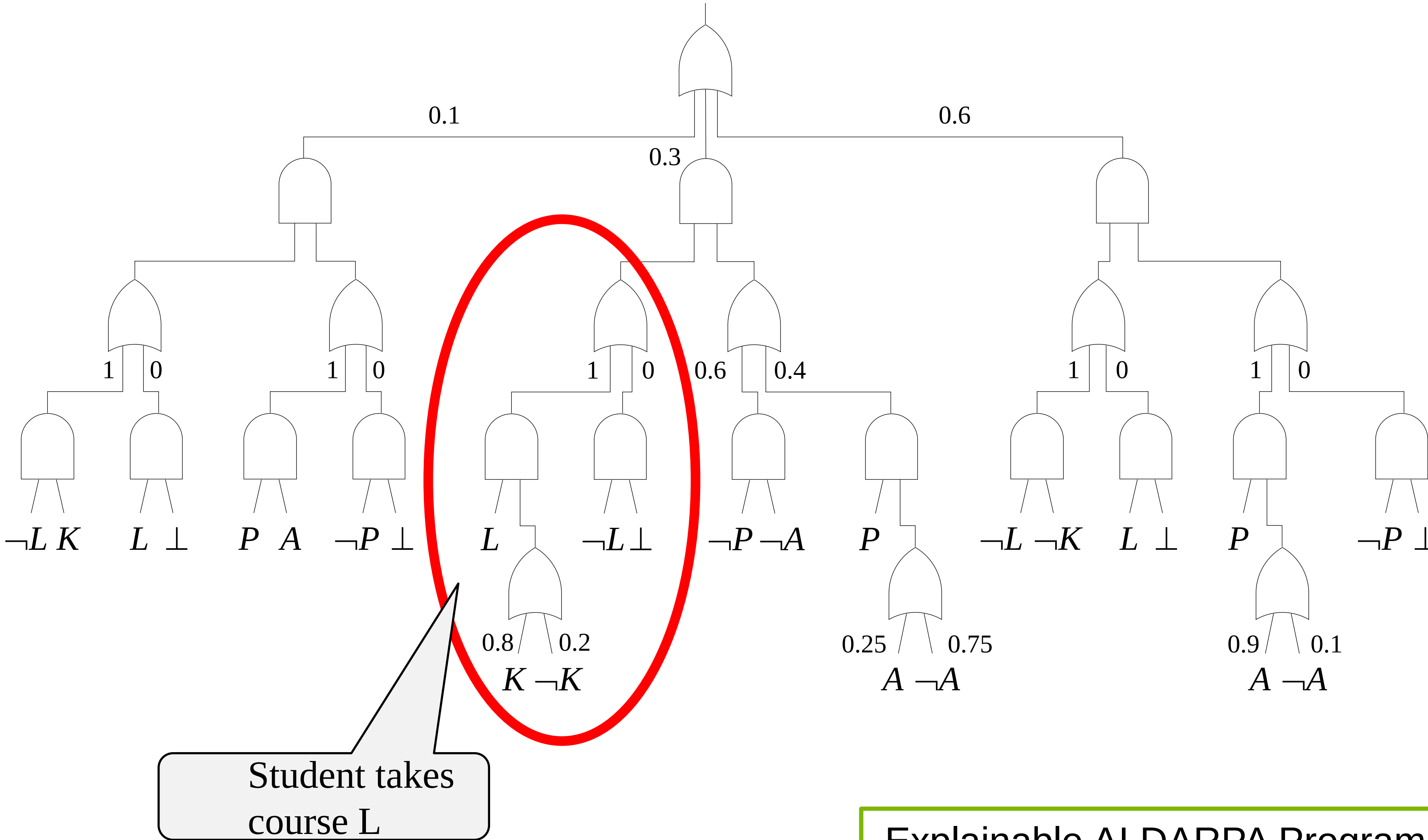
Learning Probabilistic Circuits

Parameters are Interpretable



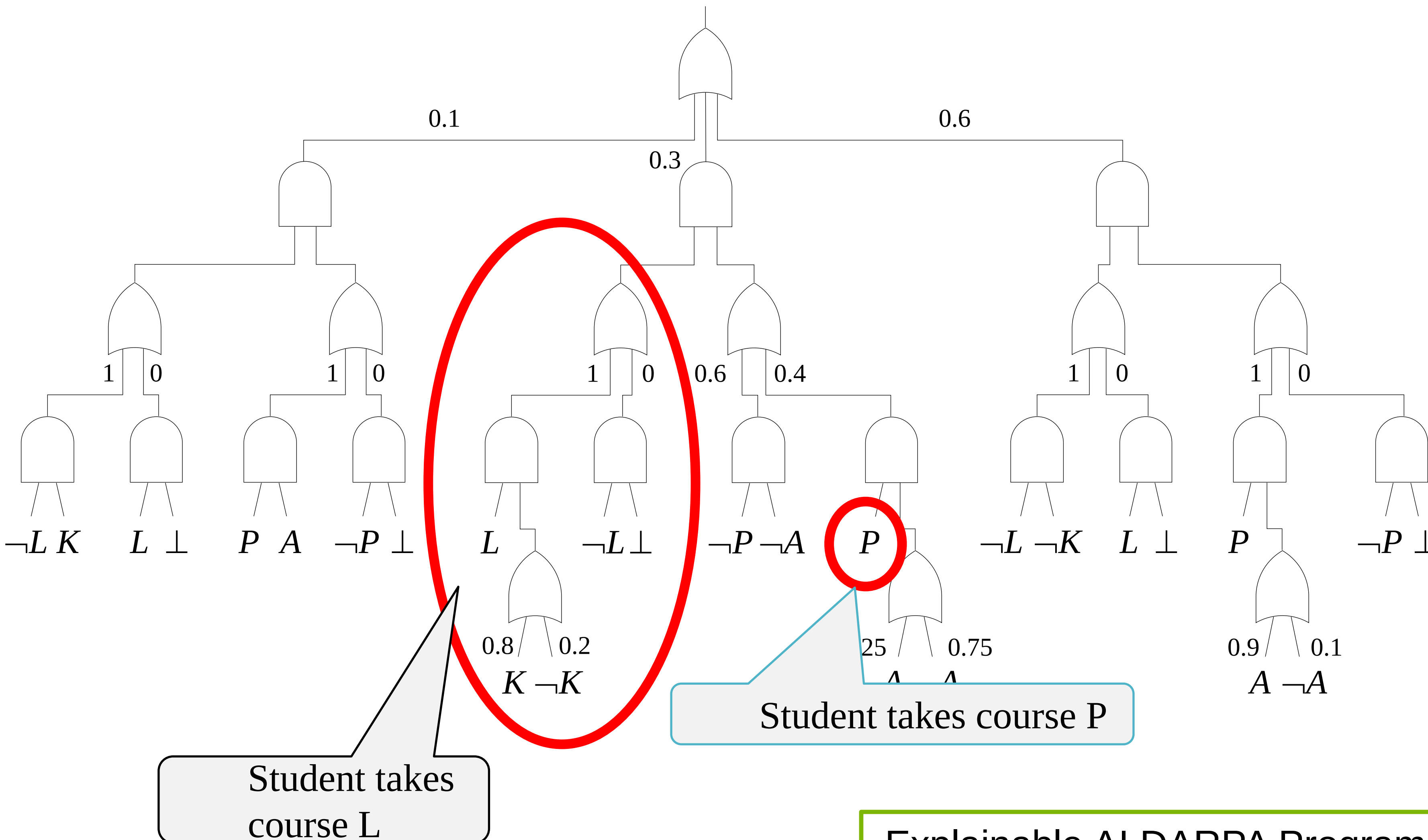
Explainable AI DARPA Program

Parameters are Interpretable



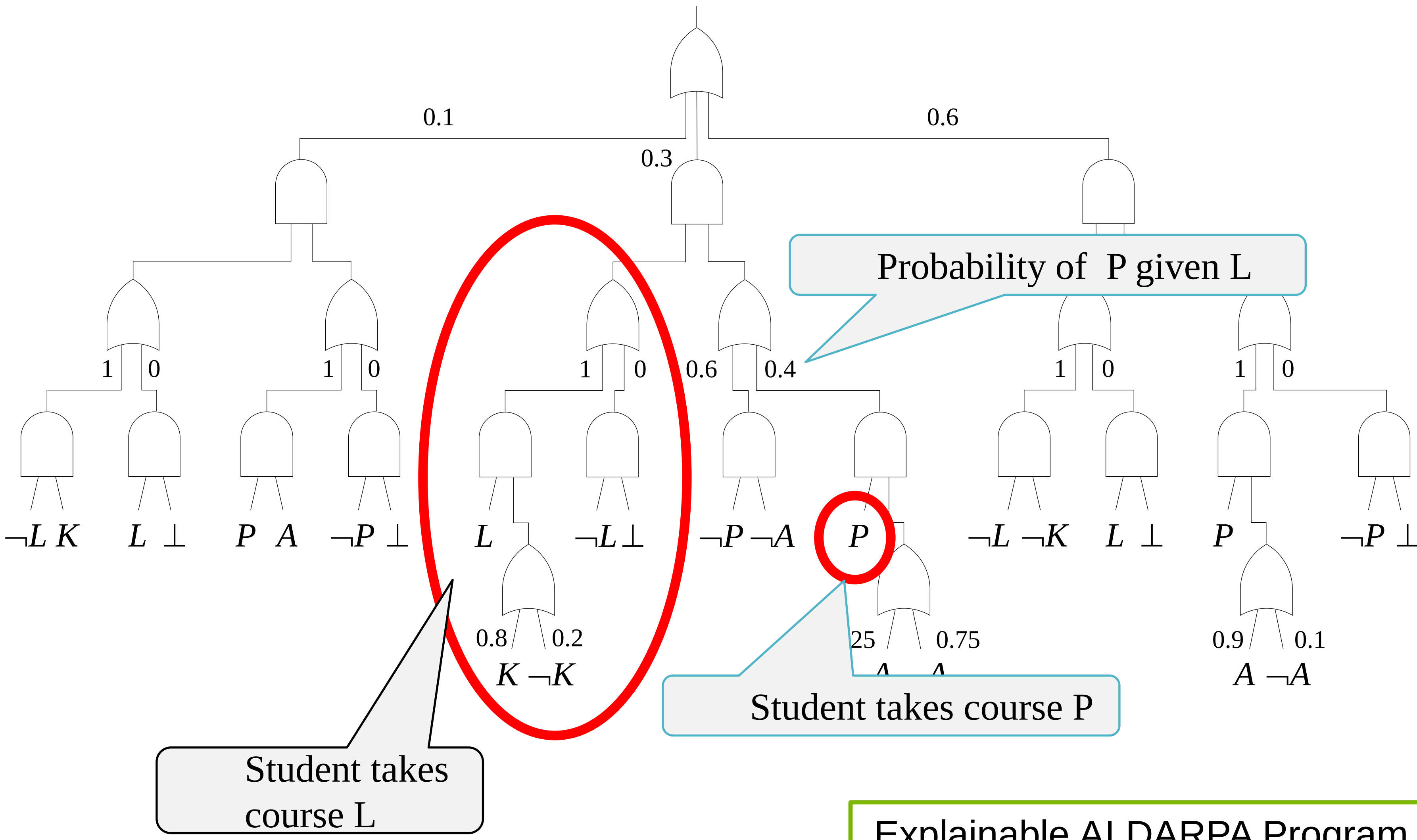
Explainable AI DARPA Program

Parameters are Interpretable



Explainable AI DARPA Program

Parameters are Interpretable



Explainable AI DARPA Program

Learning Algorithms

- Parameter learning:
  - Closed form max likelihood from complete data
  - One pass over data to estimate Pr(x|y)

Learning Algorithms

- Parameter learning:

  Closed form max likelihood from complete data

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

  Not a lot to say: very easy!

Learning Algorithms

- Parameter learning:

  Closed form max likelihood from complete data

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

  $\boxed{\text{Not a lot to say: very easy!}}$

- Structure learning:

Learning Algorithms

- Parameter learning:

  Closed form max likelihood from complete data

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

  Not a lot to say: very easy!

- Structure learning:
  - Compile logical constraint for structured space

    Use SAT solver technology

Learning Algorithms

- Parameter learning:

  Closed form max likelihood from complete data

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

  Not a lot to say: very easy!

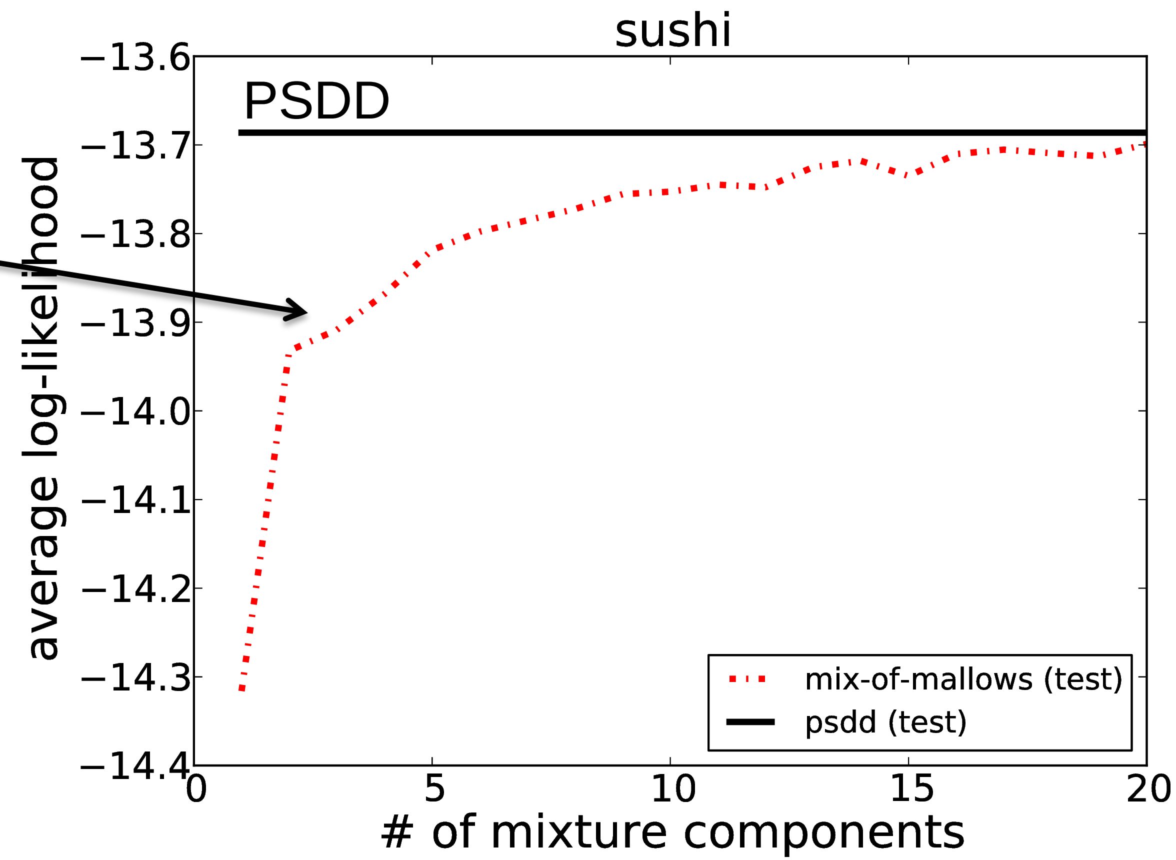
- Structure learning:
  - Compile logical constraint for structured space

    Use SAT solver technology
  - Learn structure from data by search/optimization

Learning Preference Distributions

Special-purpose distribution: Mixture-of-Mallows

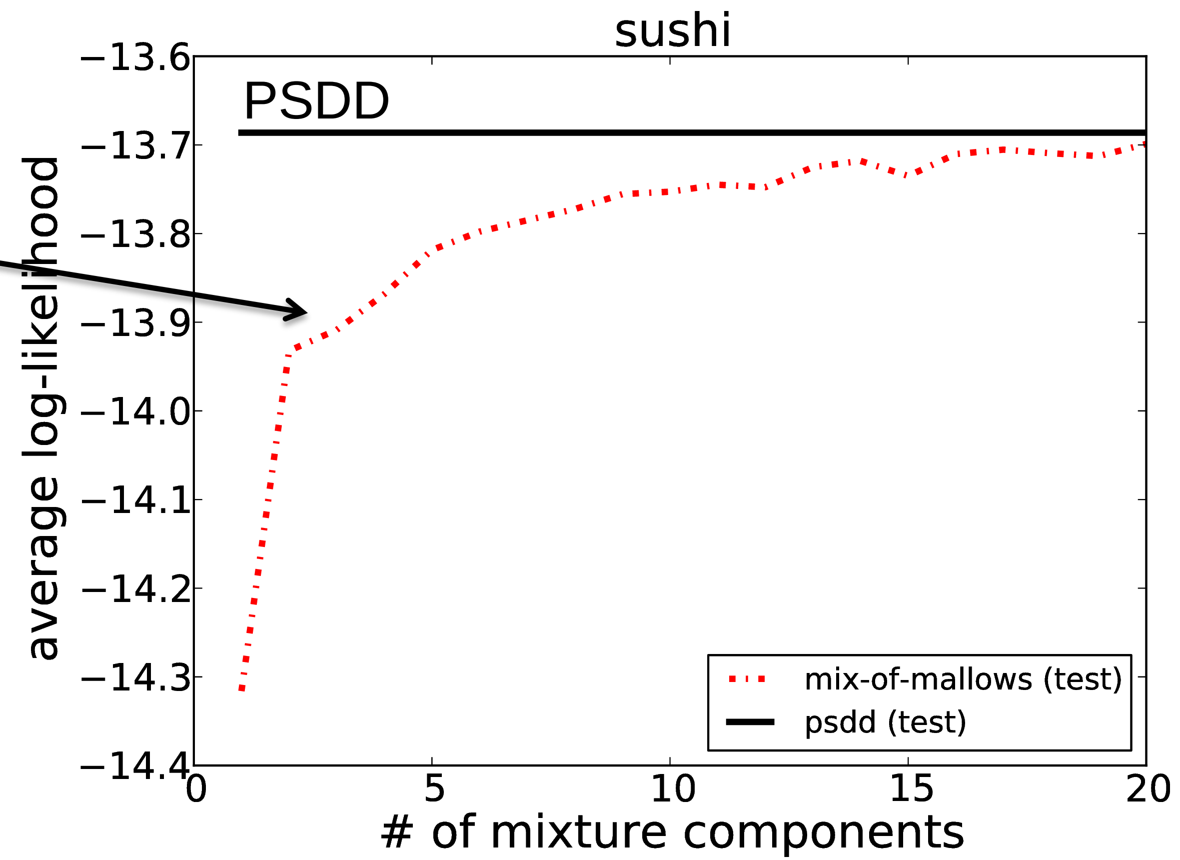
- # of components from 1 to 20
- EM with 10 random seeds
- implementation of Lu & Boutilier



Learning Preference Distributions

Special-purpose distribution: Mixture-of-Mallows

- # of components from 1 to 20
- EM with 10 random seeds
- implementation of Lu & Boutilier



This is the naive approach, circuit does not depend on data!

Learning from Incomplete Data

- Movielens Dataset:
  - 3,900 movies, 6,040 users, 1m ratings
  - take ratings from 64 most rated movies
  - ratings 1-5 converted to pairwise prefs.
- PSDD for **partial** rankings
  - 4 tiers
  - 18,711 parameters

movies by expected tier

rank	movie
1	The Godfather
2	The Usual Suspects
3	Casablanca
4	The Shawshank Redemption
5	Schindler’s List
6	One Flew Over the Cuckoo’s Nest
7	The Godfather: Part II
8	Monty Python and the Holy Grail
9	Raiders of the Lost Ark
10	Star Wars IV: A New Hope

Probabilistic-Logical Queries

rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

Probabilistic-Logical Queries

rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

- no other Star Wars movie in top-5
- at least one comedy in top-5

Probabilistic-Logical Queries

rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

- no other Star Wars movie in top-5
- at least one comedy in top-5

rank	movie
1	Star Wars V: The Empire Strikes Back
2	American Beauty
3	The Godfather
4	The Usual Suspects
5	The Shawshank Redemption

Probabilistic-Logical Queries

rank	movie
1	Star Wars V: The Empire Strikes Back
2	Star Wars IV: A New Hope
3	The Godfather
4	The Shawshank Redemption
5	The Usual Suspects

- no other Star Wars movie in top-5
- at least one comedy in top-5

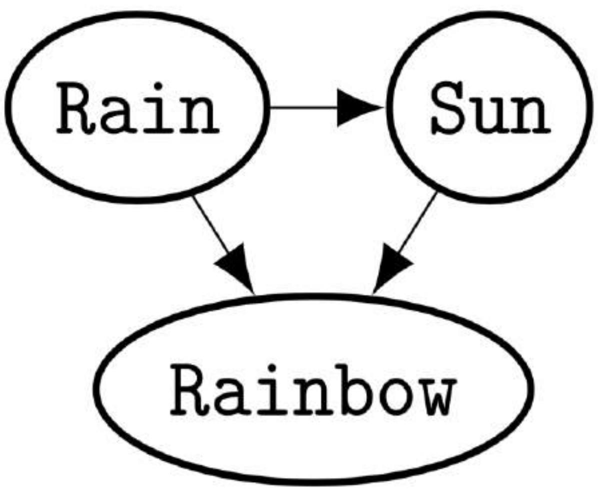
rank	movie
1	Star Wars V: The Empire Strikes Back
2	American Beauty
3	The Godfather
4	The Usual Suspects
5	The Shawshank Redemption

diversified recommendations via
logical constraints

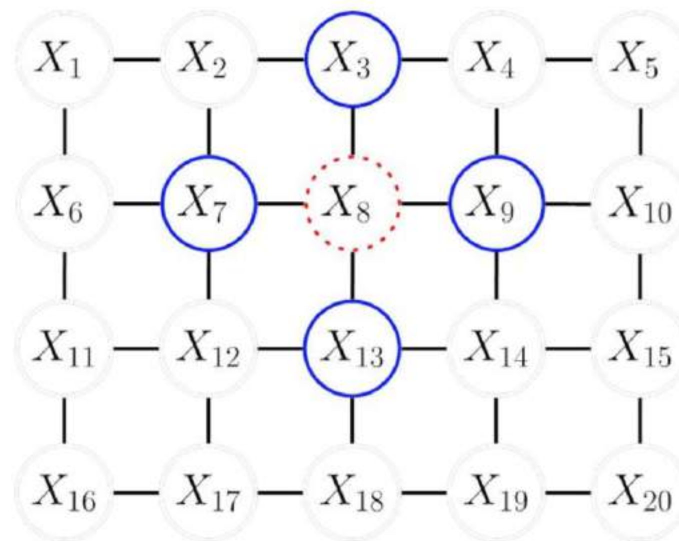
Learning Probabilistic Circuit Structure

Tractable Learning

Bayesian networks



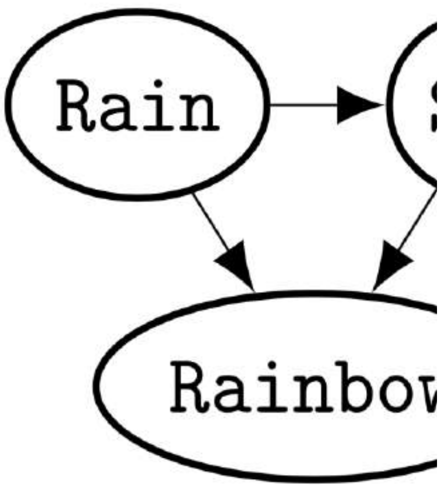
Markov networks

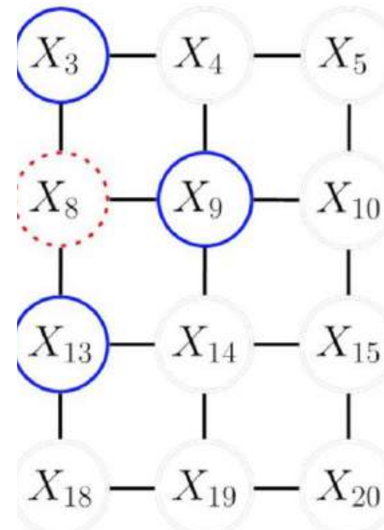


Tractable Learning

Bayesian networks

Markov networks



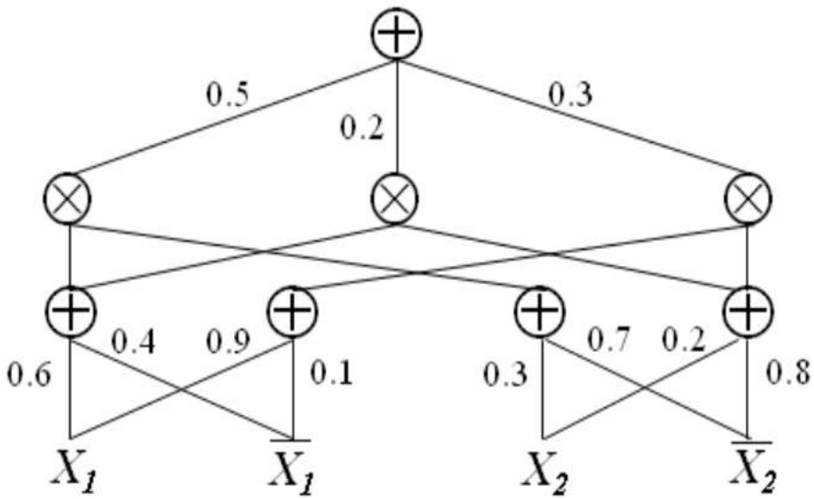


Do not support linear-time exact inference

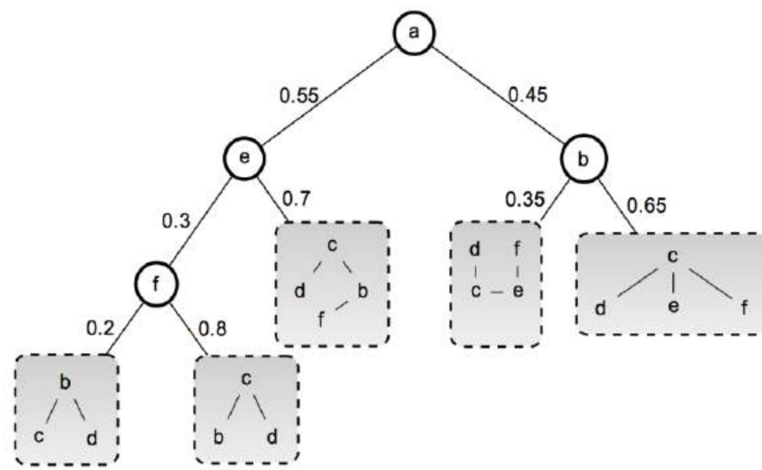
Tractable Learning

Historically: Polytrees, Chow-Liu trees, etc.

SPNs



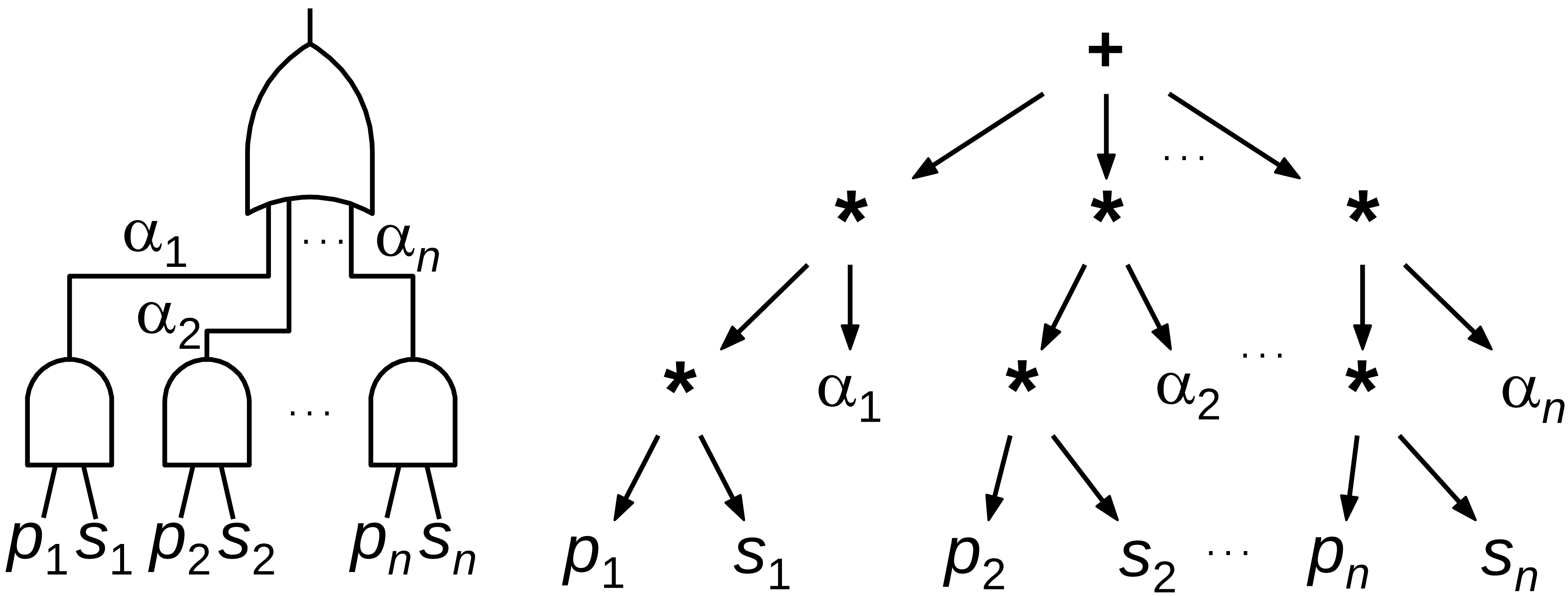
Cutset Networks



Both are Arithmetic Circuits (ACs)

[Darwiche, JACM 2003]

PSDDs are Arithmetic Circuits



PSDD **AC**

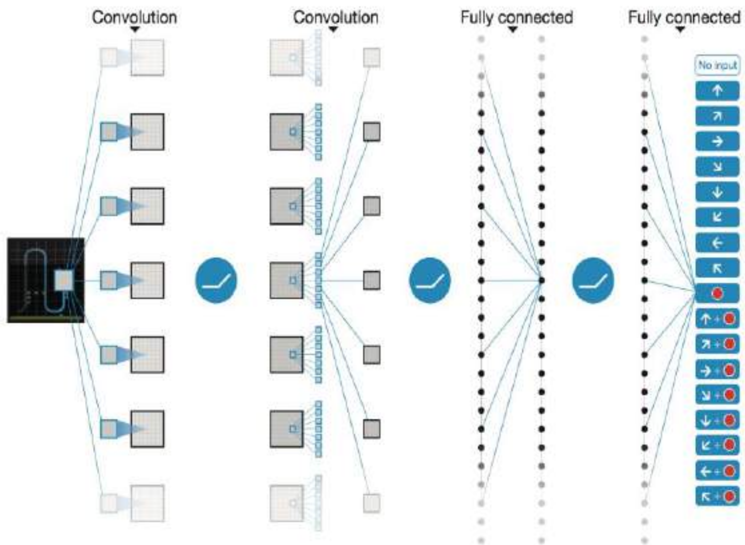
Tractable Learning

Strong Properties

Representational Freedom

Tractable Learning

DNN

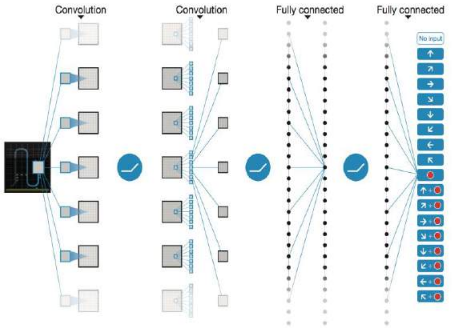


Strong Properties

Representational Freedom

Tractable Learning

DNN

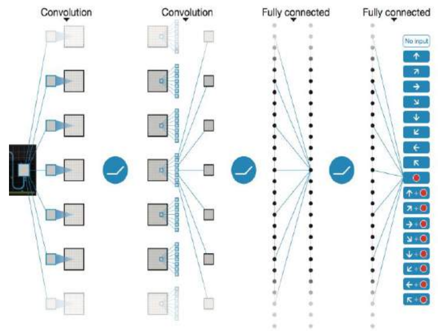


Strong Properties

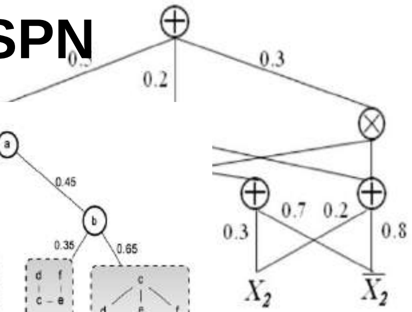
Representational Freedom

Tractable Learning

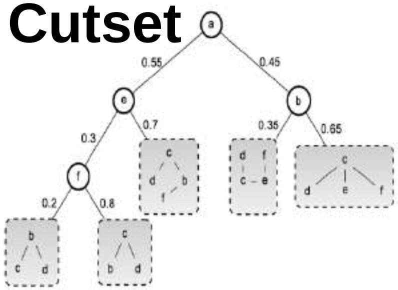
DNN



SPN



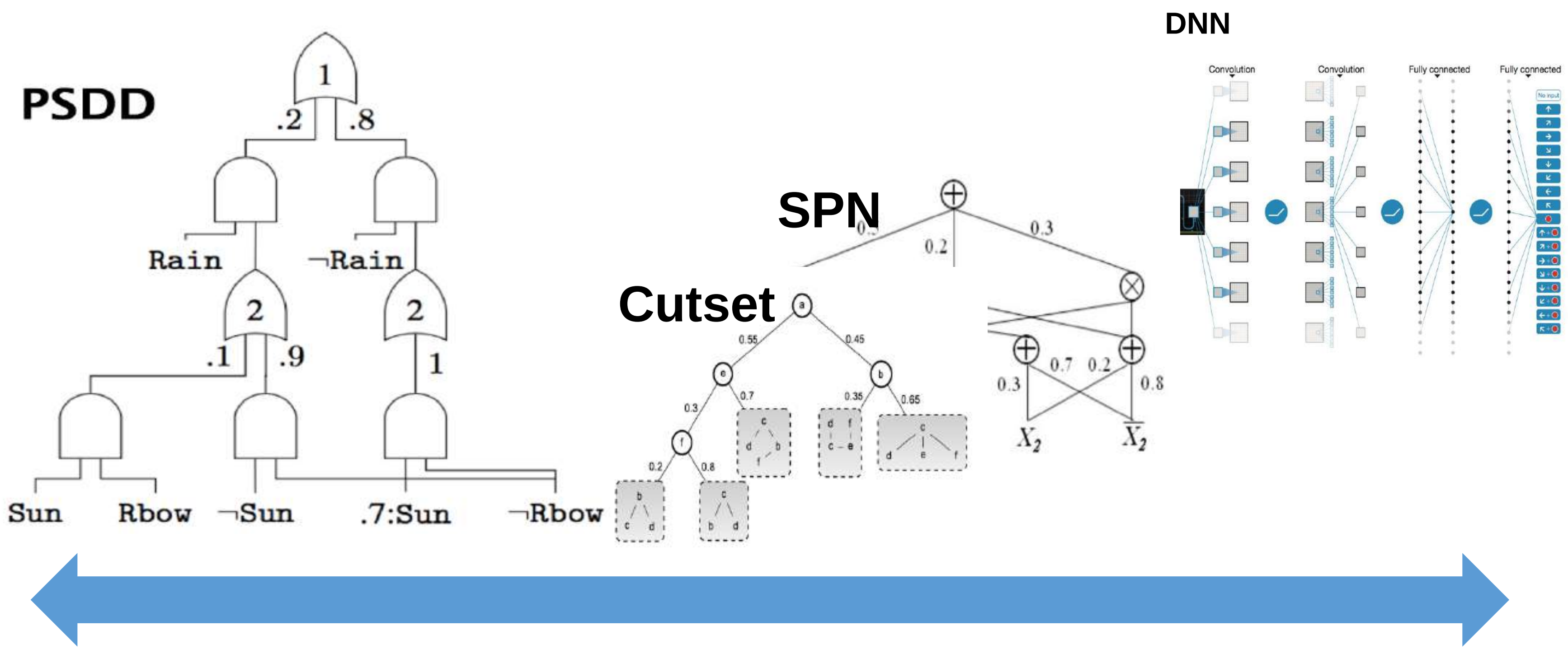
Cutset



Strong Properties

Representational Freedom

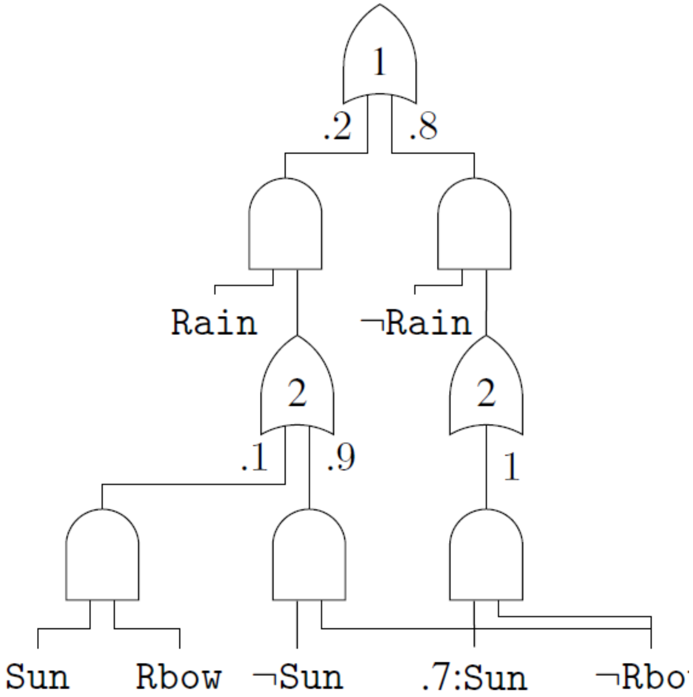
Tractable Learning



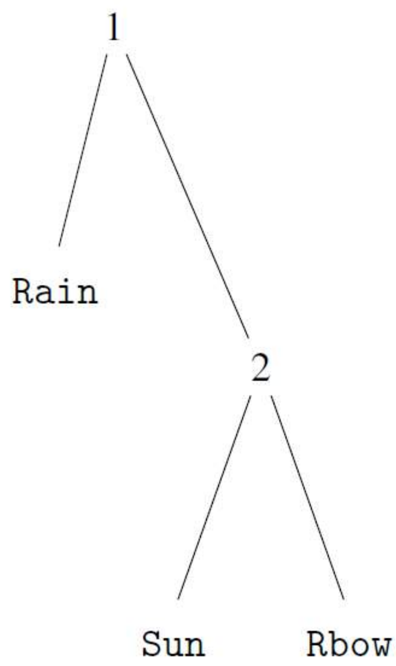
Strong Properties **Representational Freedom**

Variable Trees (vtrees)

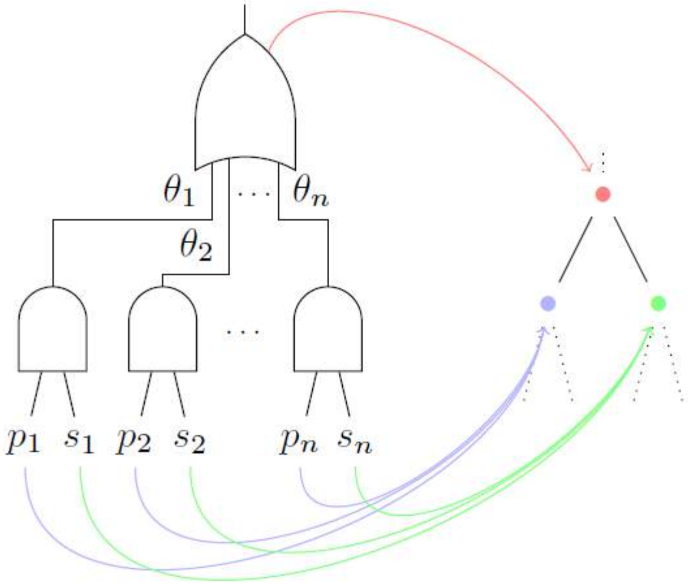
PSDD



Vtree



Correspondence



Learning Variable Trees

- How much do vars depend on each other?

$$\mathrm{MI}(\mathbf{X}, \mathbf{Y}) = \sum_{X \in \mathbf{X}} \sum_{Y \in \mathbf{Y}} \Pr(X, Y) \log \frac{\Pr(X, Y)}{\Pr(X) \Pr(Y)}$$

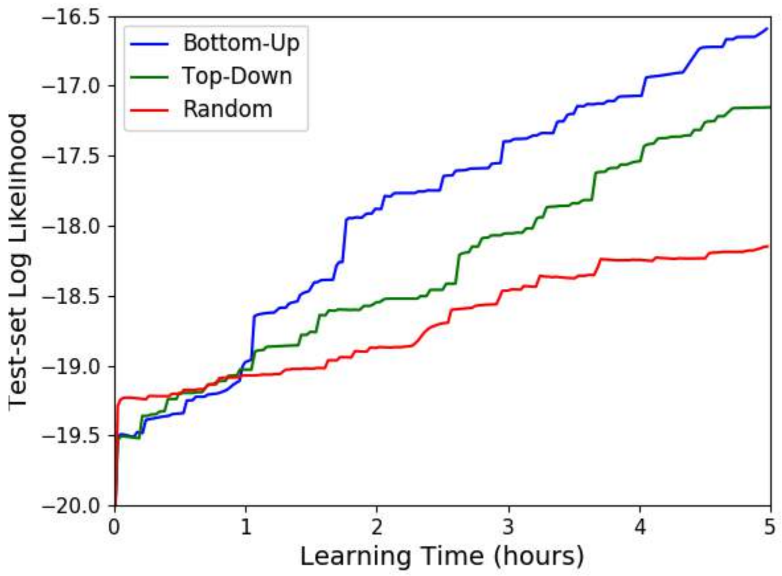
- Learn vtree by hierarchical clustering

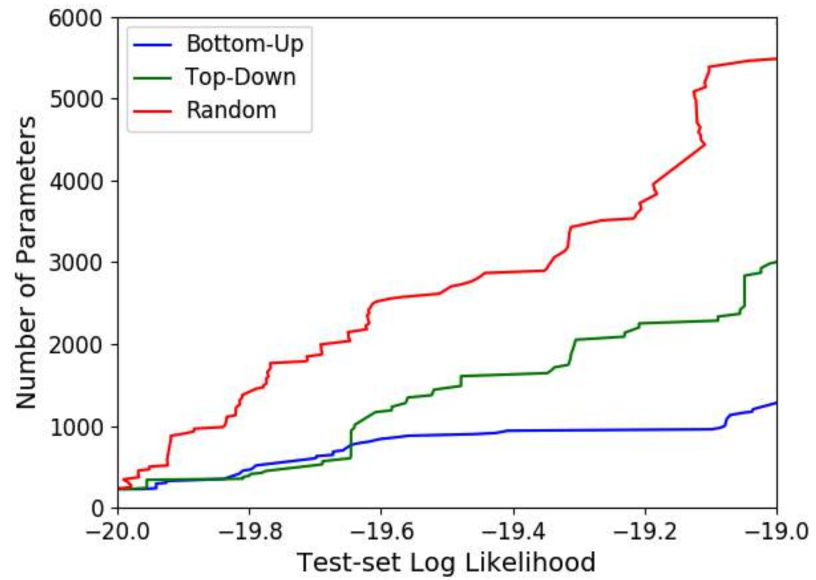
Learning Variable Trees

- How much do vars depend on each other?

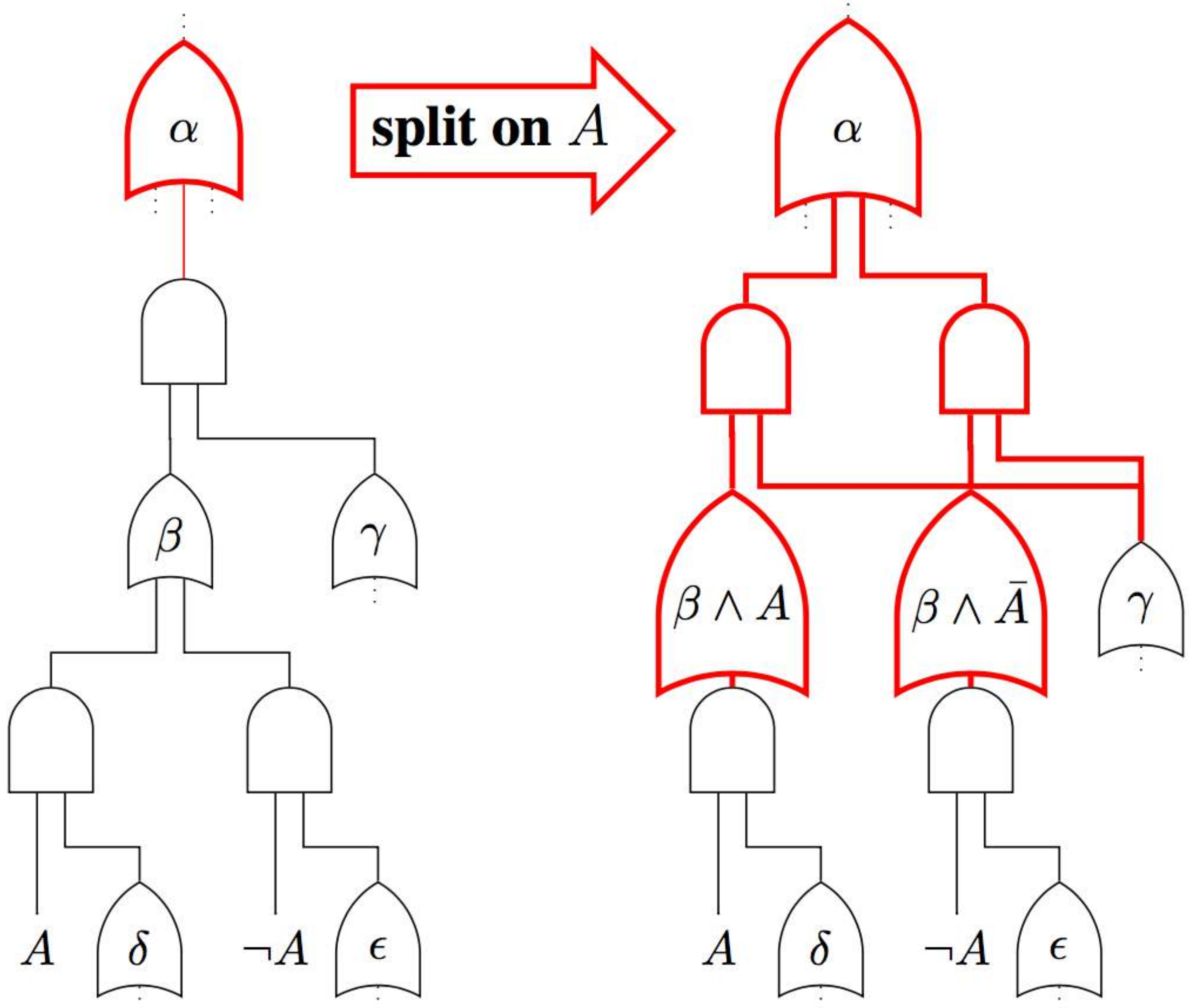
$$\mathrm{MI}(\mathbf{X},\mathbf{Y}) = \sum_{X \in \mathbf{X}} \sum_{Y \in \mathbf{Y}} \Pr(X,Y) \log \frac{\Pr(X,Y)}{\Pr(X)\Pr(Y)}$$

- Learn vtree by hierarchical clustering

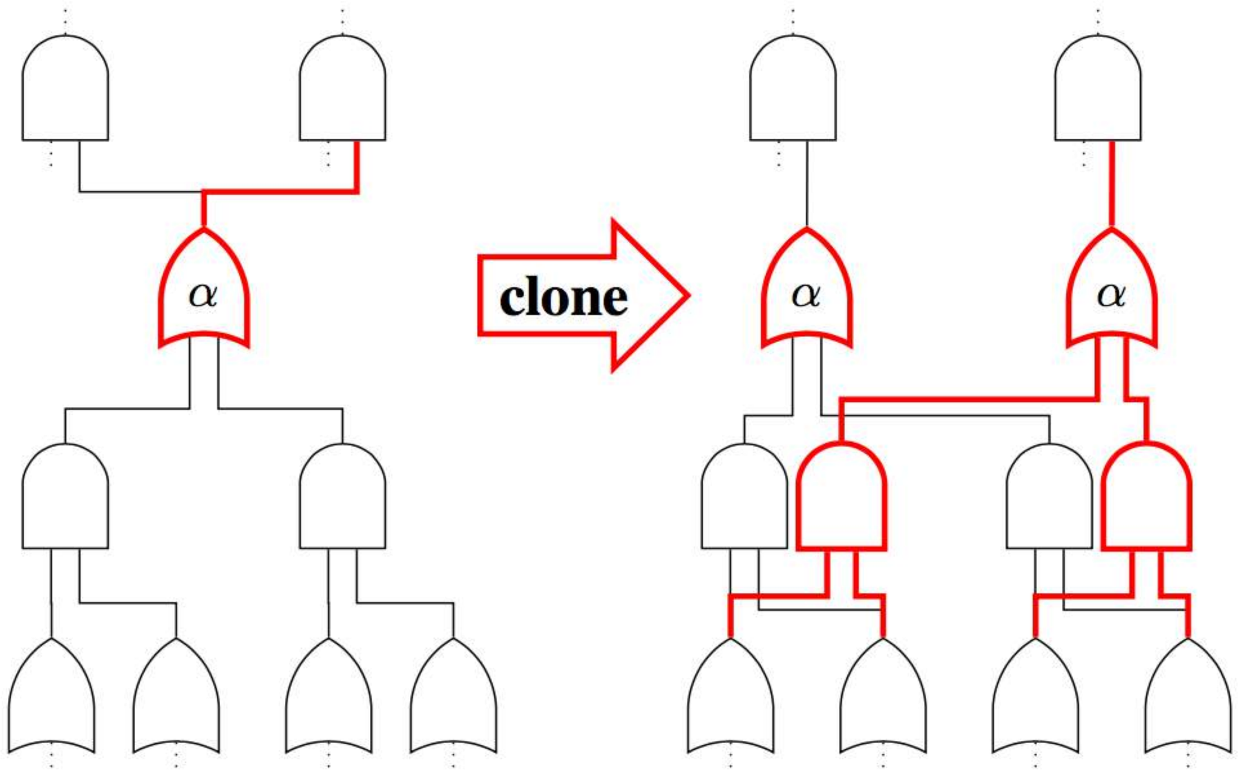




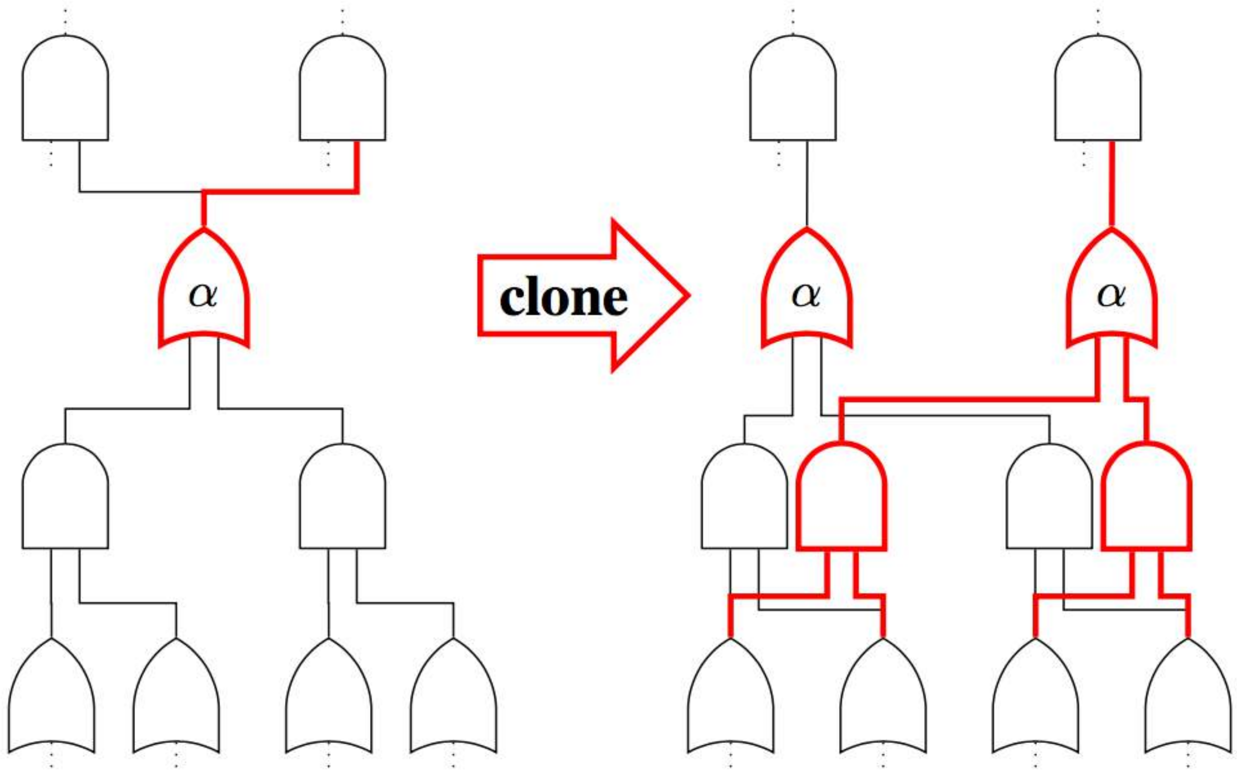
Learning Primitives



Learning Primitives

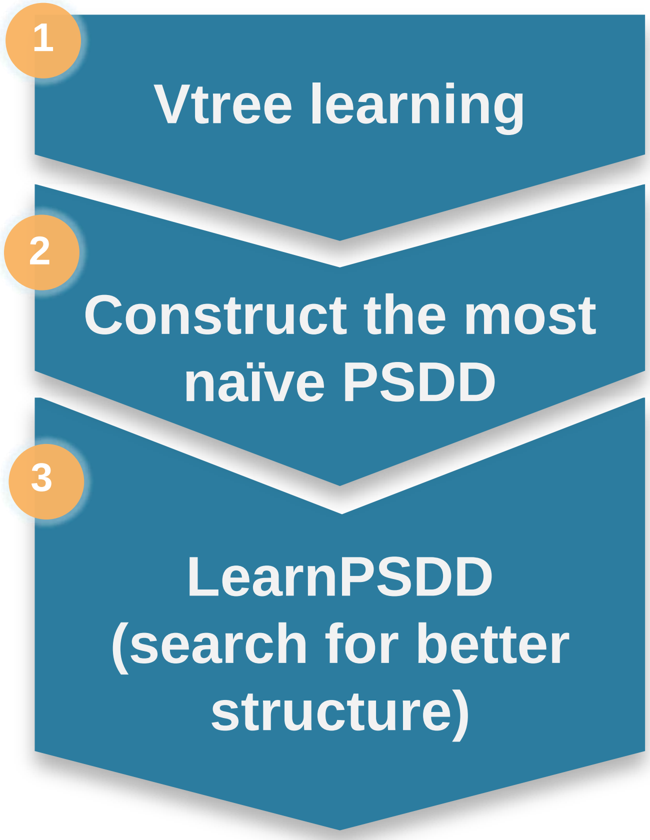


Learning Primitives

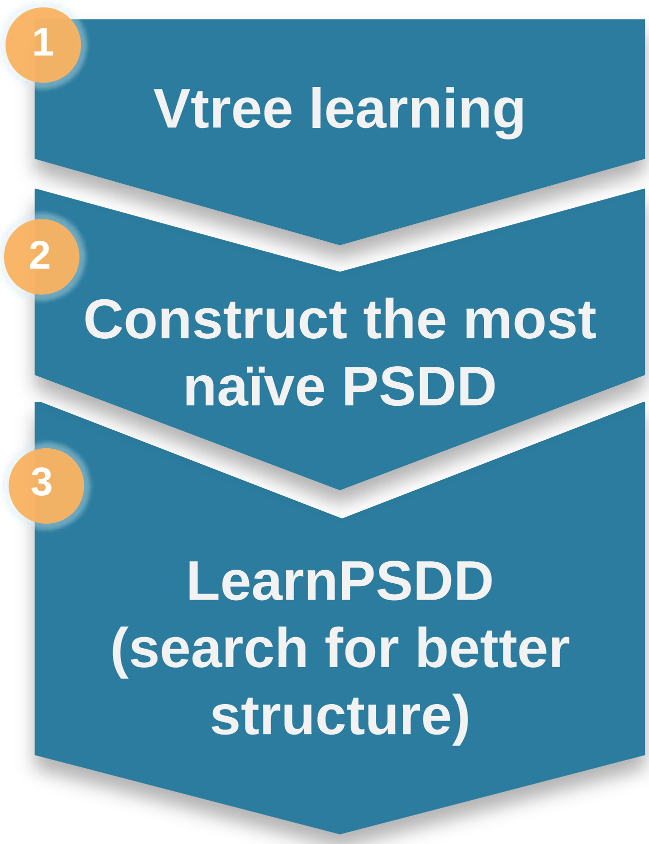


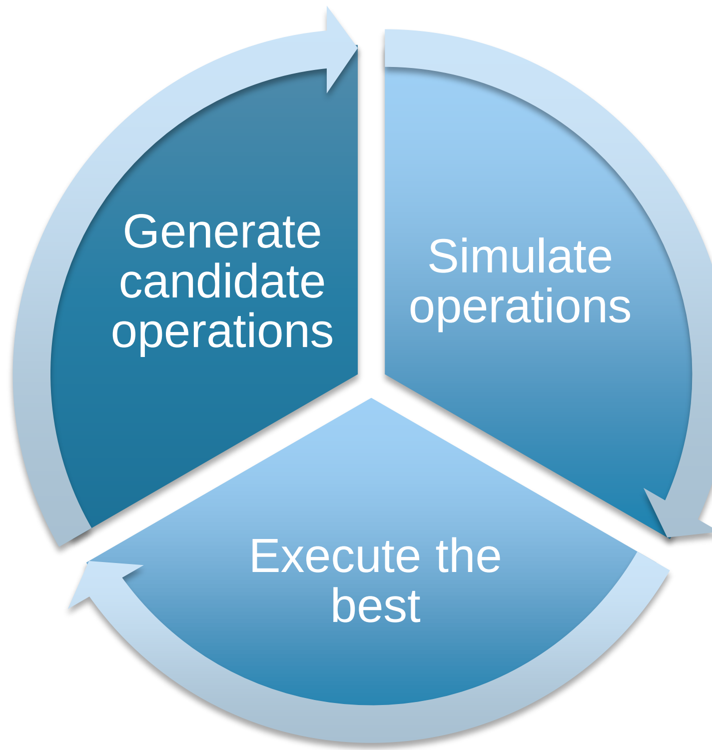
Primitives maintain PSDD properties and structured space!

LearnPSDD


1
Vtree learning
2
Construct the most naïve PSDD
3
LearnPSDD (search for better structure)

LearnPSDD





$$\text{score} = \frac{\ln \mathcal{L}(r' \mid \mathcal{D}) - \ln \mathcal{L}(r \mid \mathcal{D})}{\text{size}(r') - \text{size}(r)}$$

Experiments on 20 datasets

Datasets	|Var|	|Train|	|Valid|	|Test|	LearnPSDD		EM-LearnPSDD		SearchSPN	Merged L-SPN		Merged O-SPN	
					LL	Size	LL	Size	LL	LL	Size	LL	Size
NLTCS	16	16181	2157	3236	$-6.03^{\dagger *}$	3170	$\mathbf{-6.03}^{*}$	2147	-6.07	-6.04	3988	-6.05	1152
MSNBC	17	291326	38843	58265	$-6.05^{\dagger}$	8977	$\mathbf{-6.04}^{*}$	3891	-6.06	-6.46	2440	-6.08	9478
KDD	64	1800992	19907	34955	$-2.16^{\dagger}$	14974	$\mathbf{-2.12}^{*}$	9182	$\mathbf{-2.16}$	-2.14	6670	-2.19	16608
Plants	69	17412	2321	3482	-14.93	13129	-13.79^{*}	13951	$-13.12^{\dagger}$	$\mathbf{-12.69}$	47802	-13.49	36960
Audio	100	15000	2000	3000	-42.53	13765	-41.98^{*}	9721	$-40.13^{\dagger}$	$\mathbf{-40.02}$	10804	-42.06	6142
Jester	100	9000	1000	4116	-57.67	11322	-53.47^{*}	7014	$-53.08^{\dagger}$	$\mathbf{-52.97}$	10002	-55.36	4996
Netflix	100	15000	2000	3000	-58.92	10997	-58.41^{*}	6250	$-56.91^{\dagger}$	$\mathbf{-56.64}$	11604	-58.64	6142
Accidents	111	12758	1700	2551	-34.13	10489	-33.64^{*}	6752	$-30.02^{\dagger}$	$\mathbf{-30.01}$	13322	-30.83	6846
Retail	135	22041	2938	4408	-11.13	4091	$\mathbf{-10.81}^{*}$	7251	$-10.97^{\dagger}$	-10.87	2162	-10.95	3158
Pumsb-Star	163	12262	1635	2452	-34.11	10489	-33.67^{*}	7965	$-28.69^{\dagger}$	$\mathbf{-24.11}$	17604	-24.34	18338
DNA	180	1600	400	1186	-89.11^{*}	6068	-92.67	14864	$-81.76^{\dagger}$	$\mathbf{-85.51}$	4320	-87.49	1430
Kosarek	190	33375	4450	6675	$-10.99^{\dagger}$	11034	-10.81^{*}	10179	-11.00	$\mathbf{-10.62}$	5318	-10.98	6712
MSWeb	294	29441	32750	5000	$-10.18^{\dagger}$	11389	-9.97^{*}	14512	-10.25	$\mathbf{-9.90}$	16484	-10.06	12770
Book	500	8700	1159	1739	-35.90	15197	-34.97^{*}	11292	$-34.91^{\dagger}$	$\mathbf{-34.76}$	11998	-37.44	11916
EachMovie	500	4524	1002	591	-56.43^{*}	12483	-58.01	16074	$-53.28^{\dagger}$	$\mathbf{-52.07}$	15998	-58.05	19846
WebKB	839	2803	558	838	-163.42	10033	-161.09^{*}	18431	$-157.88^{\dagger}$	$\mathbf{-153.55}$	20134	-161.17	10046
Reuters-52	889	6532	1028	1530	-94.94	10585	-89.61^{*}	9546	$-86.38^{\dagger}$	$\mathbf{-83.90}$	46232	-87.49	28334
20NewsGrp.	910	11293	3764	3764	-161.41	12222	-161.09^{*}	18431	$-153.63^{\dagger}$	$\mathbf{-154.67}$	43684	-161.46	29016
BBC	1058	1670	225	330	-260.83	10585	$\mathbf{-253.19}^{*}$	20327	$-252.13^{\dagger}$	-253.45	21160	-260.59	8454
AD	1556	2461	327	491	-30.49^{*}	9666	-31.78	9521	$-16.97^{\dagger}$	-16.77	49790	$\mathbf{-15.39}$	31070

Experiments on 20 datasets

Compare with O-SPN: smaller size in 14, better LL in 11, win on both in 6

Compare with L-SPN: smaller size in 14, better LL in 6, win on both in 2

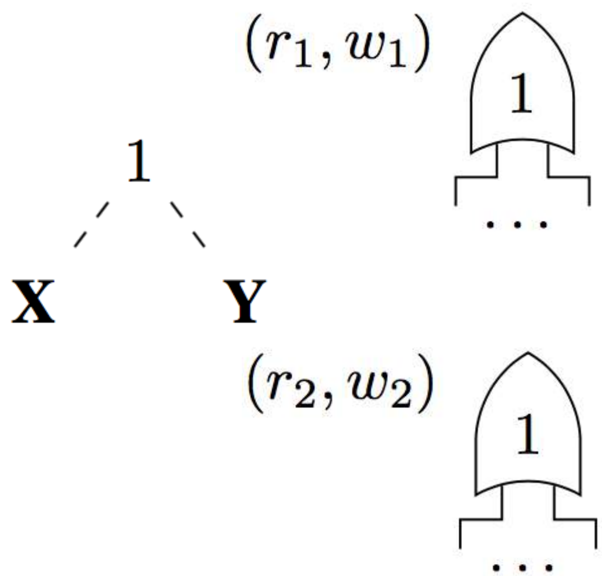
Experiments on 20 datasets

Compare with O-SPN: smaller size in 14, better LL in 11, win on both in 6

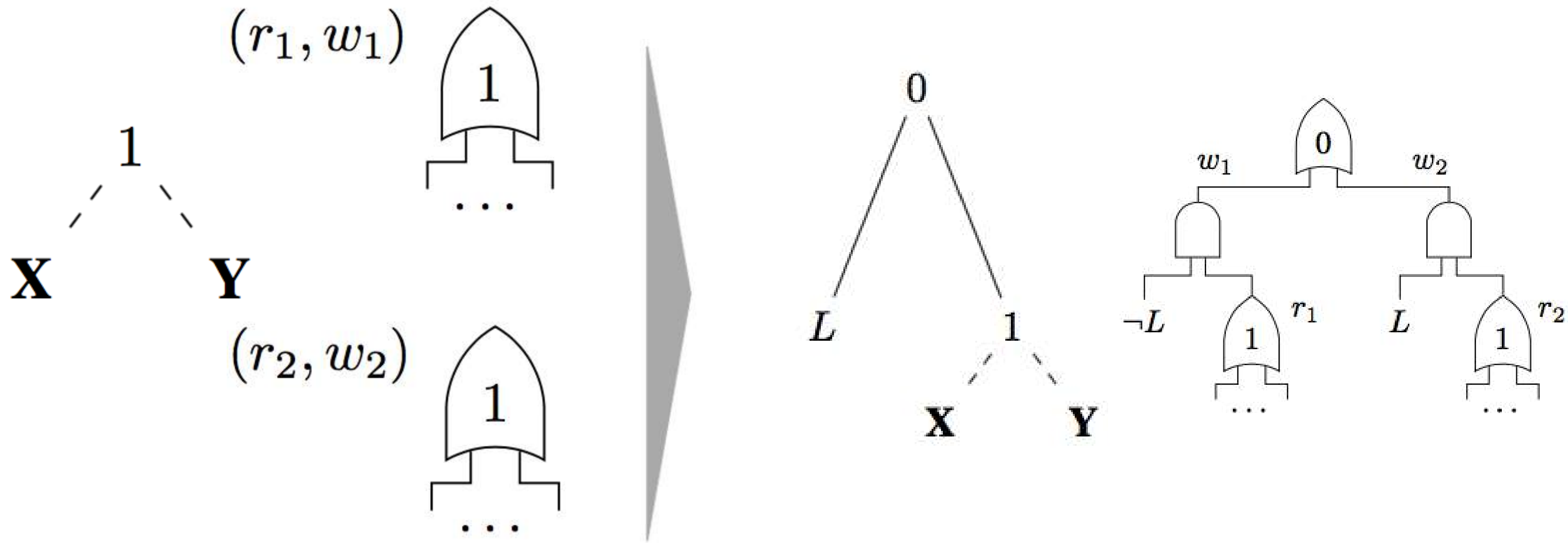
Compare with L-SPN: smaller size in 14, better LL in 6, win on both in 2

Comparable in performance & Smaller in size

Ensembles of PSDDs

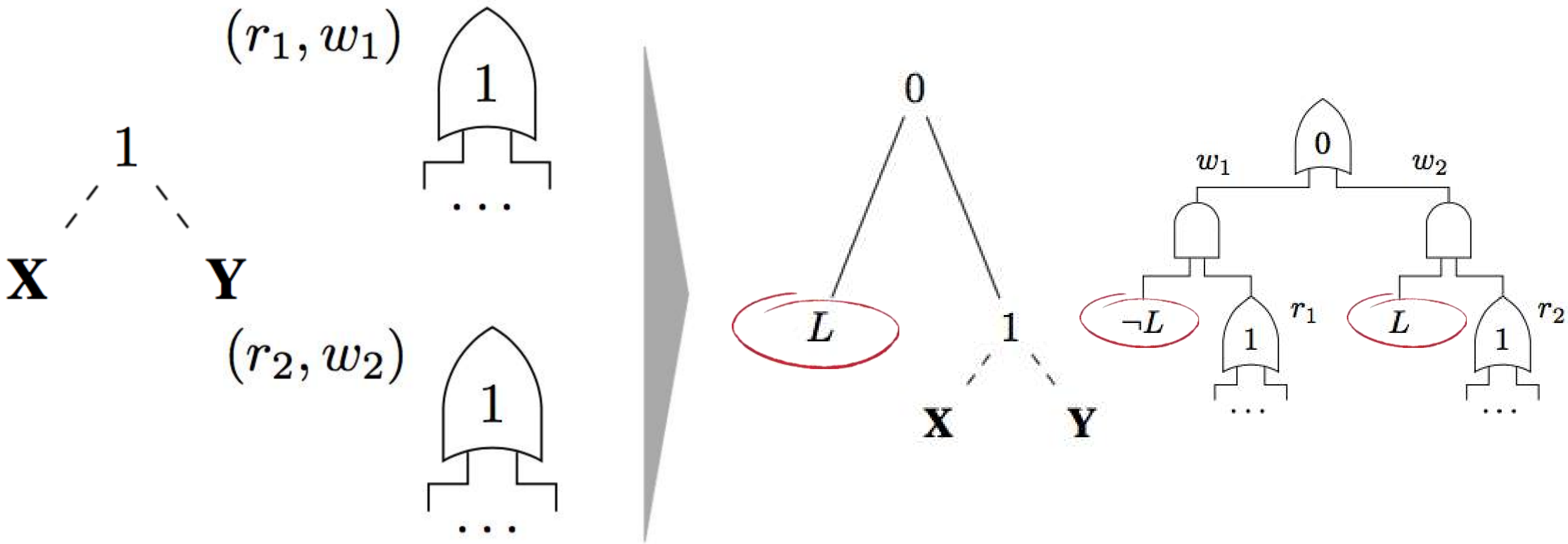


Ensembles of PSDDs



EM/Bagging

Ensembles of PSDDs



EM/Bagging

State-of-the-Art Performance

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 Performance

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 in 6 datasets

What happens if you have a structured space?

Multi-valued data = exactly-one constraint

What happens if you have a structured space?

Multi-valued data = exactly-one constraint

$$\begin{cases} \boldsymbol{x_1} \lor \boldsymbol{x_2} \lor \boldsymbol{x_3} \\ \neg \boldsymbol{x_1} \lor \neg \boldsymbol{x_2} \\ \neg \boldsymbol{x_2} \lor \neg \boldsymbol{x_3} \\ \neg \boldsymbol{x_1} \lor \neg \boldsymbol{x_3} \end{cases}$$

What happens if you have a structured space?

Multi-valued data = exactly-one constraint

$$\begin{cases} \boldsymbol{x_1} \vee \boldsymbol{x_2} \vee \boldsymbol{x_3} \\ \neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_2} \\ \neg \boldsymbol{x_2} \vee \neg \boldsymbol{x_3} \\ \neg \boldsymbol{x_1} \vee \neg \boldsymbol{x_3} \end{cases}$$

Datasets	**No Constraint**	**PSDD**	**LearnPSDD**
Adult	−18.41	−14.14	−12.86
CovType	−14.39	−8.81	−7.32

What happens if you have a structured space?

Multi-valued data = exactly-one constraint

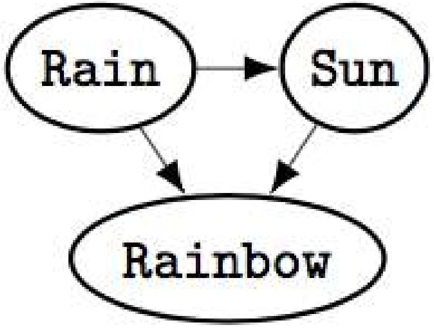
$$\begin{cases} x_1 \lor x_2 \lor x_3 \\ \neg x_1 \lor \neg x_2 \\ \neg x_2 \lor \neg x_3 \\ \neg x_1 \lor \neg x_3 \end{cases}$$

Datasets	**No Constraint**	**PSDD**	**LearnPSDD**
Adult	−18.41	−14.14	−12.86
CovType	−14.39	−8.81	−7.32

Never omit domain constraints!

Circuit-Based Probabilistic Reasoning

Compilation for Inference

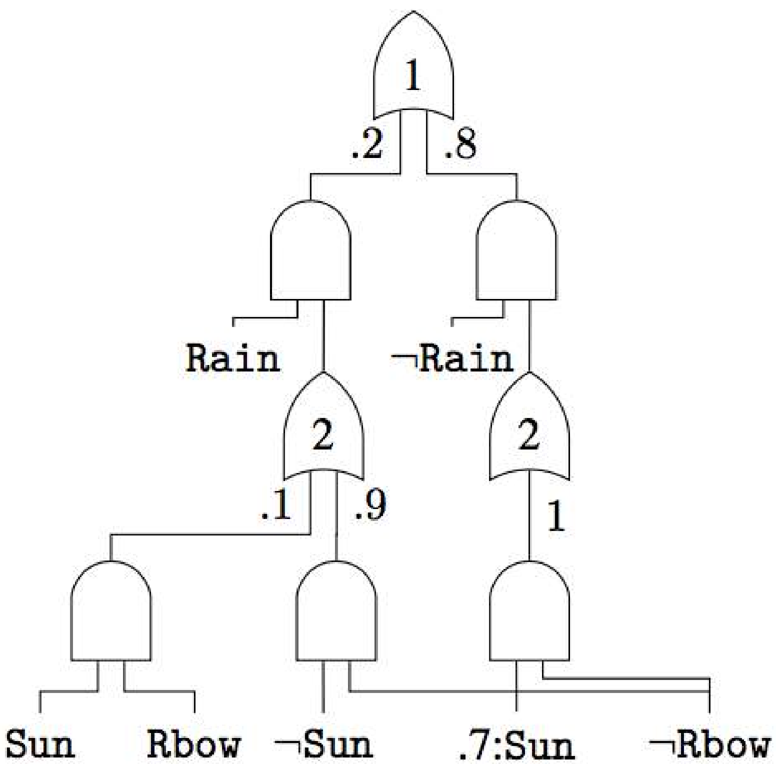


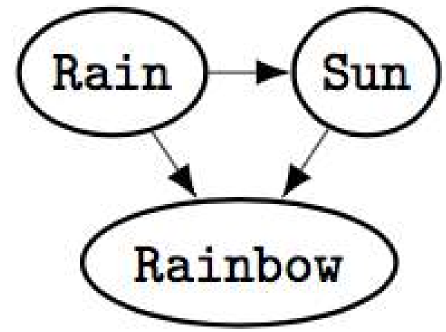
$$\Pr(\texttt{Rain}) = 0.2,$$

$$\Pr(\texttt{Sun} \mid \texttt{Rain}) = \begin{cases} 0.1 \text{ if } & \texttt{Rain} \\ 0.7 \text{ if } & \neg\texttt{Rain} \end{cases}$$

$$\Pr(\texttt{Rbow} \mid \texttt{R}, \texttt{S}) = \begin{cases} 1 \text{ if } \texttt{Rain} \wedge \texttt{Sun} \\ 0 \text{ otherwise} \end{cases}$$

Compilation for Inference





$$\Pr(\texttt{Rain}) = 0.2,$$

$$\Pr(\texttt{Sun} \mid \texttt{Rain}) = \begin{cases} 0.1 \text{ if } \texttt{Rain} \\ 0.7 \text{ if } \neg\texttt{Rain} \end{cases}$$

$$\Pr(\texttt{Rbow} \mid \texttt{R}, \texttt{S}) = \begin{cases} 1 \text{ if } \texttt{Rain} \wedge \texttt{Sun} \\ 0 \text{ otherwise} \end{cases}$$

Ongoing Work

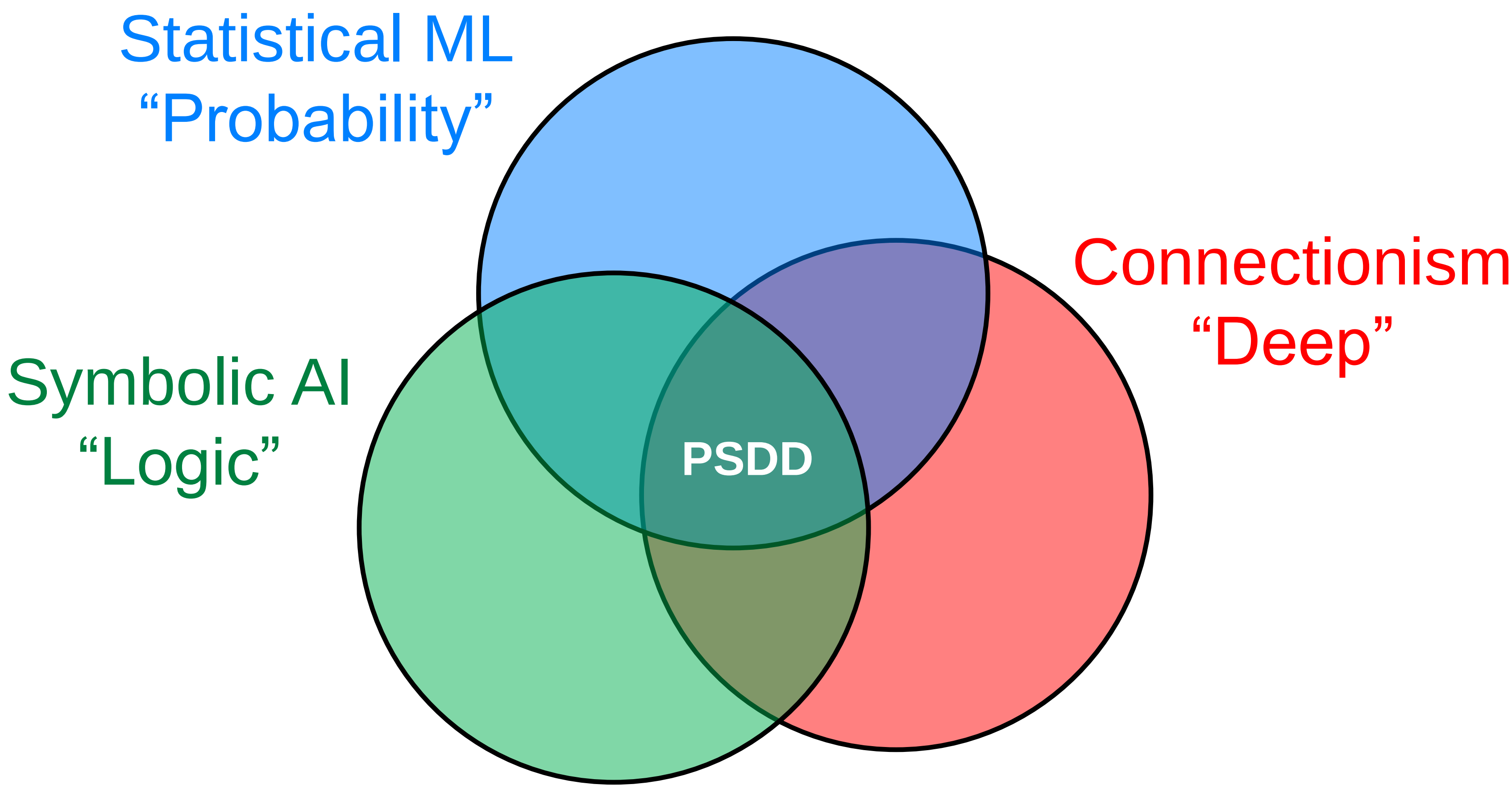
- Probabilistic program inference
  by compilation
- Approximate inference
  by collapsed compilation
- Robust feature selection
  by compilation [IJCAI18]

- Powerful reasoning toolbox!

Conclusions

- Logic is everywhere in machine learning ☺
- Probabilistic circuits build on logical circuits
  1. Tractability
  2. Semantics
  3. Natural encoding of structured spaces
- Learning is effective
  1. Enforcing neural network output constraints
     State of the art semi-supervised learning and complex output
  2. Density estimation from constraints encoding structured space
     State of the art learning preference distributions
  3. Density estimation from standard unstructured datasets
     State of the art on standard tractable learning datasets

Conclusions



References

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 International Conference on Machine Learning (ICML)*, 2018.

YooJung Choi and Guy Van den Broeck. **On Robust Trimming of Bayesian Network Classifiers**, *In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang and Guy Van den Broeck. **A Semantic Loss Function for Deep Learning Under Weak Supervision**, *In NIPS 2017 Workshop on Learning with Limited Labeled Data: Weak Supervision and Beyond*, 2017.

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.

YooJung Choi, Adnan Darwiche and Guy Van den Broeck. **Optimal Feature Selection for Decision Robustness in Bayesian Networks**, *In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

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.

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.

Arthur Choi, Guy Van den Broeck and Adnan Darwiche. **Probability Distributions over Structured Spaces**, *In Proceedings of the AAAI Spring Symposium on KRR*, 2015.

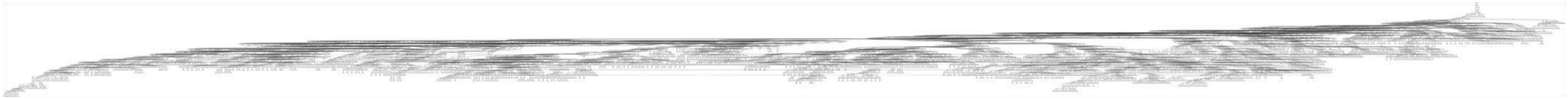
Arthur Choi, Guy Van den Broeck and Adnan Darwiche. **Tractable Learning for Structured Probability Spaces: A Case Study in Learning Preference Distributions**, *In Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

Doga Kisa, Guy Van den Broeck, Arthur Choi and Adnan Darwiche. **Probabilistic sentential decision diagrams: Learning with massive logical constraints**, *In ICML Workshop on Learning Tractable Probabilistic Models (LTPM)*, 2014.

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.

(… and ongoing work by Tal Friedman, YooJung Choi, and Yitao Liang)

Questions?



PSDD with 15,000 nodes

LearnPSDD code: https://github.com/UCLA-StarAI/LearnPSDD
Other code online soon