

# Scalable Learning of Probabilistic Circuits

USP



# Motivation

Given a selection of sushi...



...and people's preferences...

**Alice:**     

**Bob:**     

**Carol:**     

...how can we model this as a probability distribution...

$$p(1^{\text{st}} = \text{salmon nigiri}, 3^{\text{rd}} = \text{salmon nigiri})$$

$$p(2^{\text{nd}} = \text{salmon nigiri} \mid 1^{\text{st}} = \text{maki roll})$$

$$\arg \max p(1^{\text{st}} = ?, 2^{\text{nd}} = ?, 3^{\text{rd}} = ?, 4^{\text{th}} = \text{maki roll}, 5^{\text{th}} = \text{tuna nigiri})$$

$$p((3^{\text{rd}} = \text{salmon nigiri} \rightarrow 1^{\text{st}} = \text{maki roll}) \vee 2^{\text{nd}} = \text{salmon nigiri})$$

...and extract meaningful queries from it?

# Motivation

Given a selection of sushi...



...and people's preferences...

**Alice:**     

**Bob:**     

**Carol:**     

...how can we model this as a probability distribution...

$$p(1^{\text{st}} = \text{salmon nigiri}, 3^{\text{rd}} = \text{salmon maki})$$

$$p(2^{\text{nd}} = \text{salmon maki} \mid 1^{\text{st}} = \text{white rice ball})$$

$$\arg \max p(1^{\text{st}} = ?, 2^{\text{nd}} = ?, 3^{\text{rd}} = ?, 4^{\text{th}} = \text{white rice ball}, 5^{\text{th}} = \text{tuna maki})$$

$$p((3^{\text{rd}} = \text{salmon nigiri} \rightarrow 1^{\text{st}} = \text{white rice ball}) \vee 2^{\text{nd}} = \text{salmon maki})$$

**Marginals**

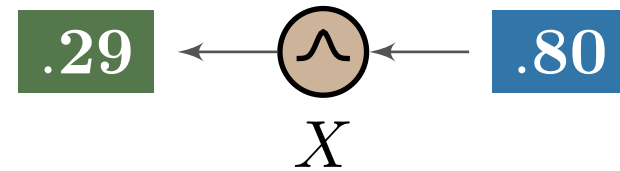
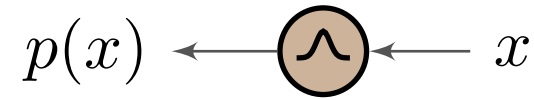
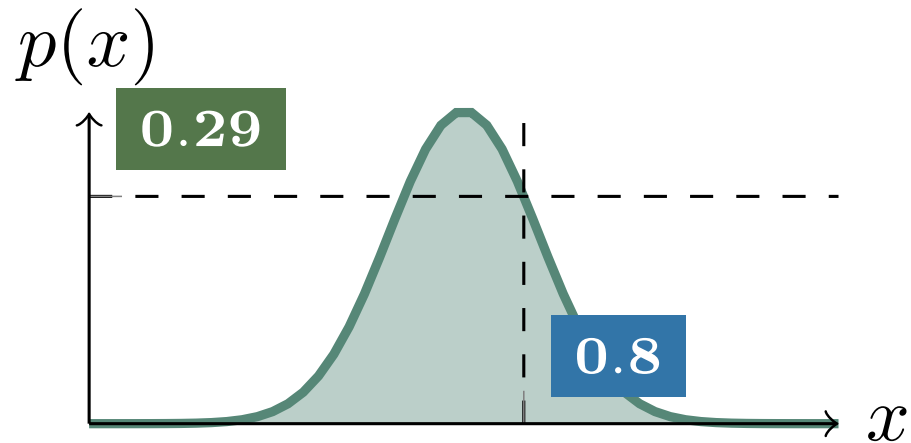
**Conditionals**

**MPE**

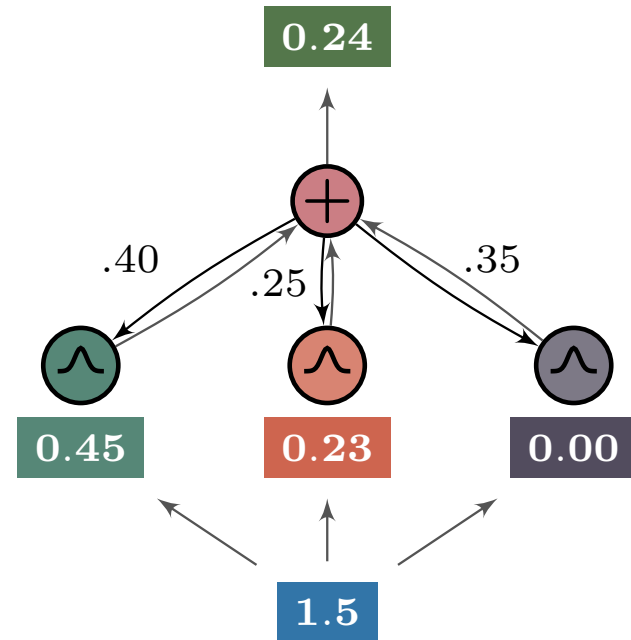
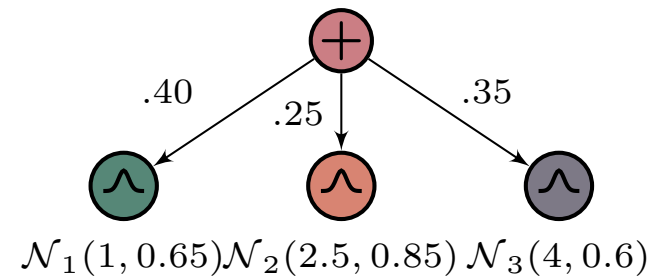
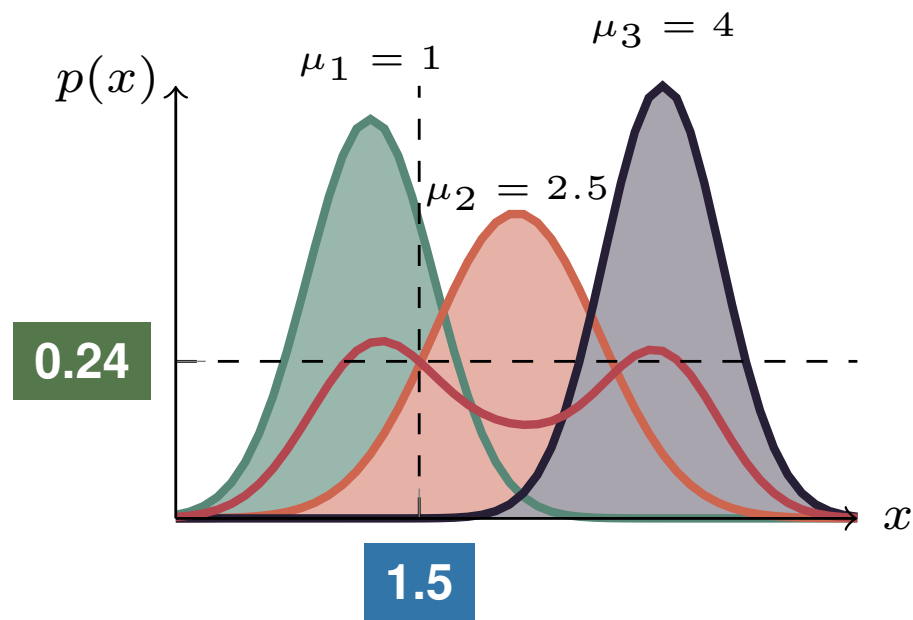
**Logical events**

...and extract meaningful queries from it?

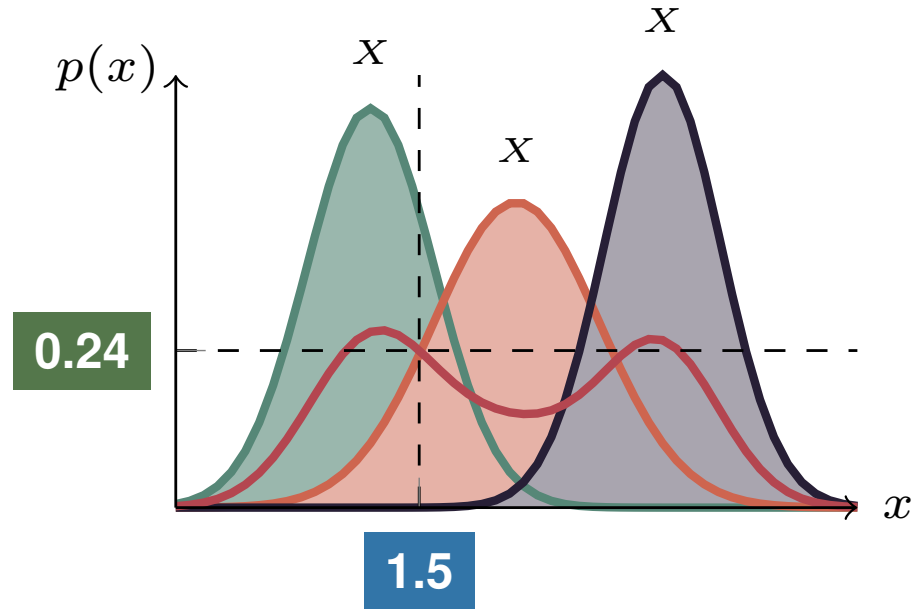
# Probabilistic Circuits – Inputs



# Probabilistic Circuits – Sums

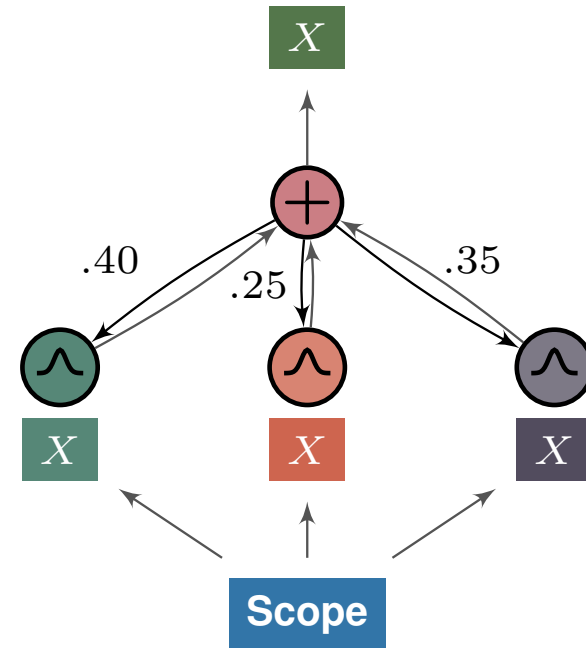
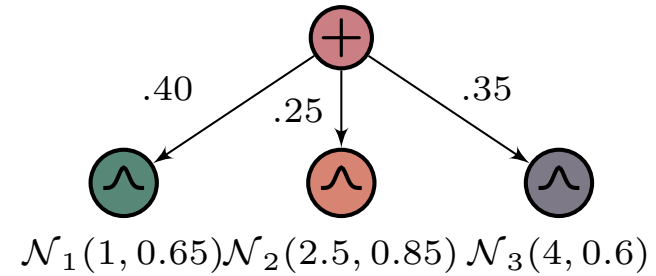


# Probabilistic Circuits – Smoothness

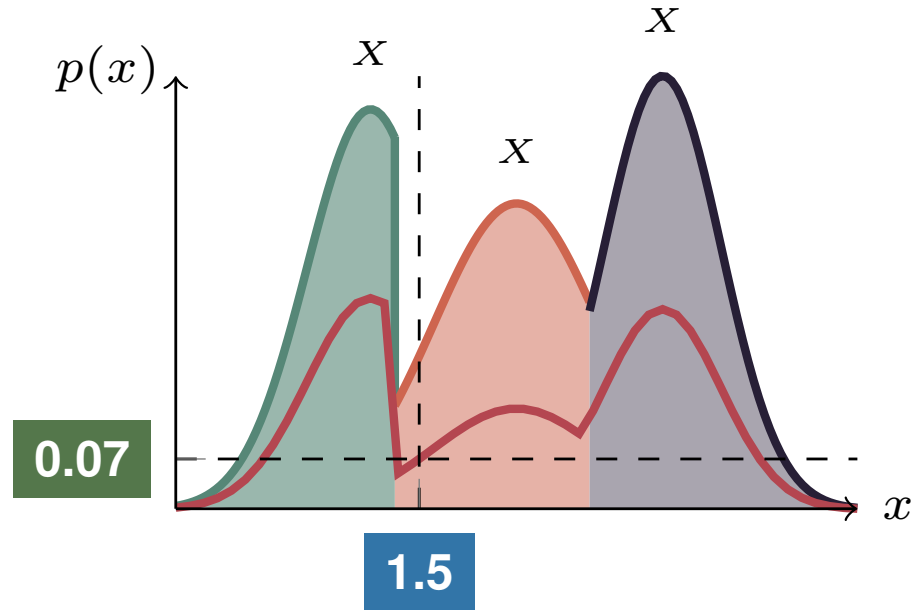


**Definition 1** (Smoothness).

Every sum node child mentions the same variables.

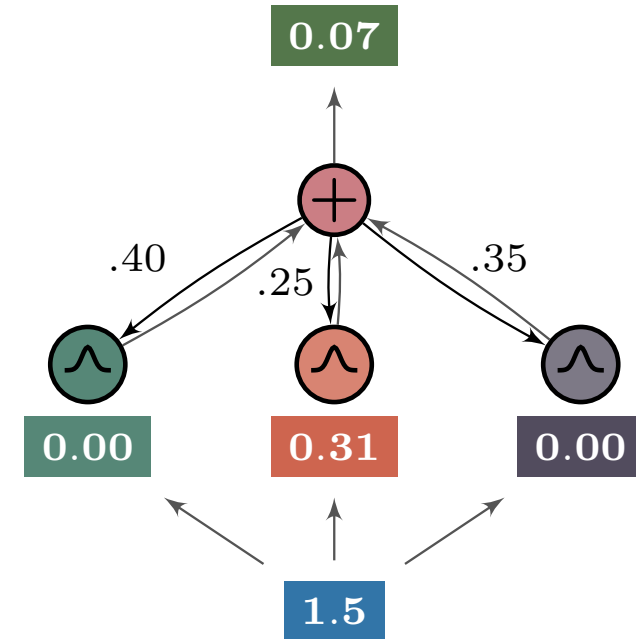
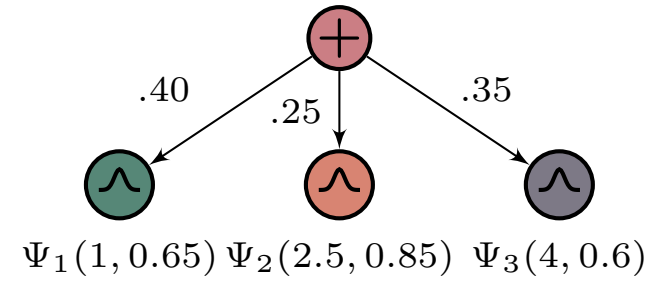


# Probabilistic Circuits – Determinism

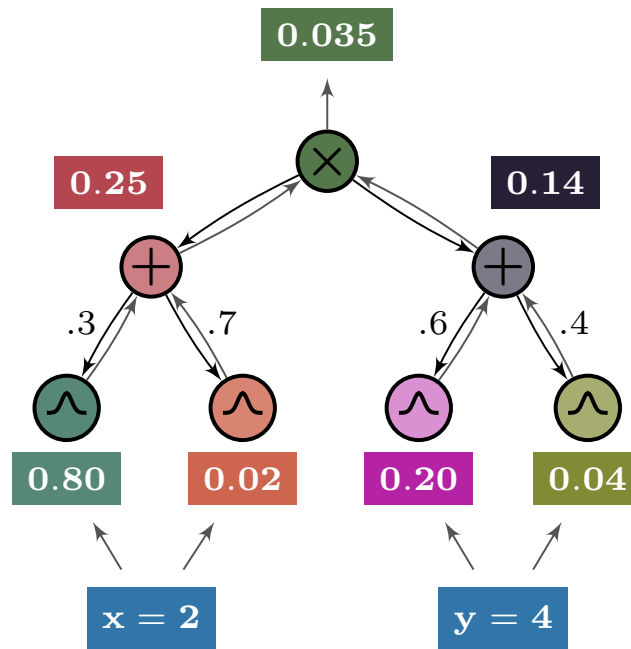
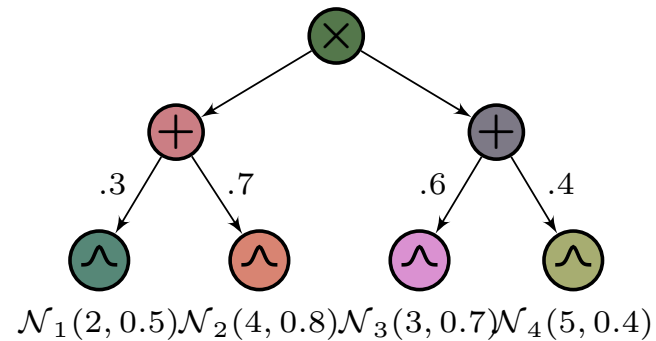
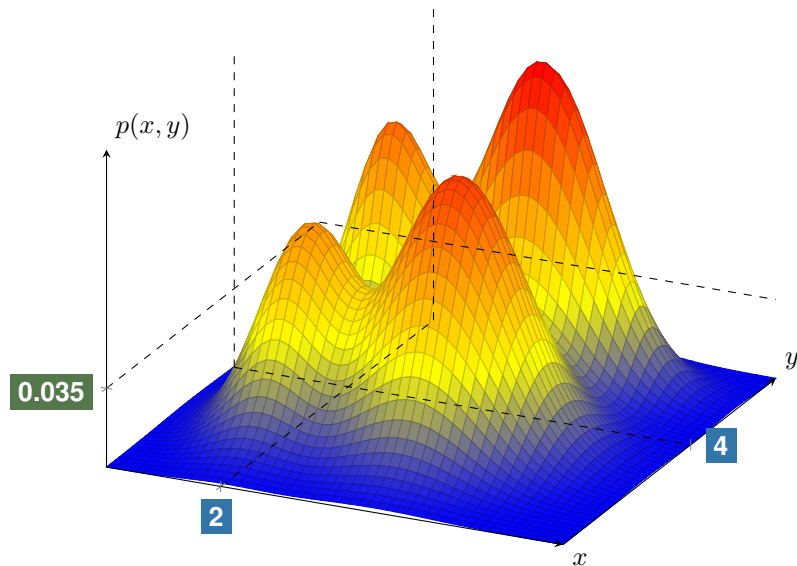
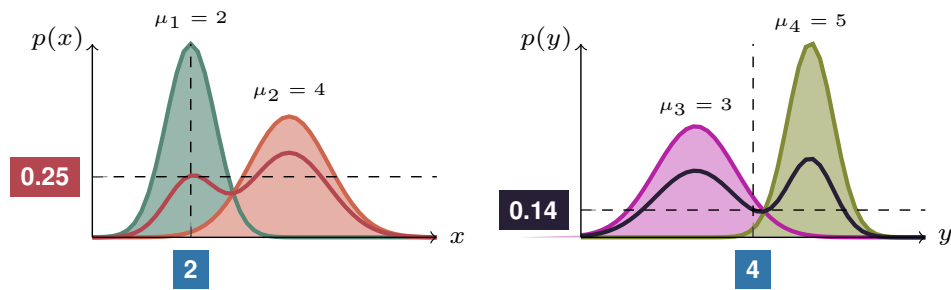


**Definition 2** (Determinism).

*At most one sum node child has a positive value.*

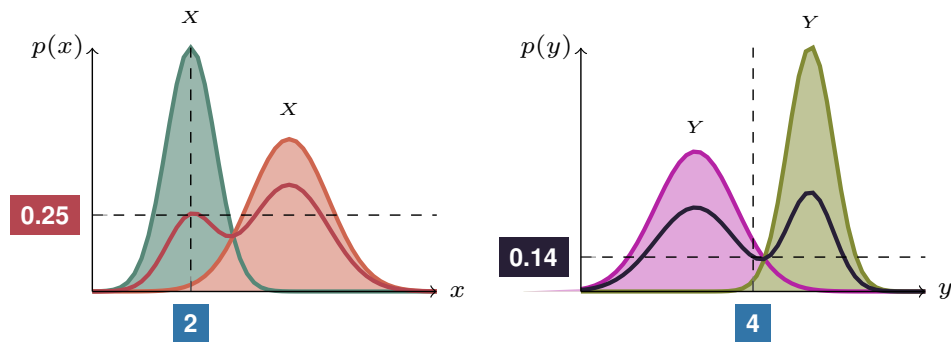


# Probabilistic Circuits – Products

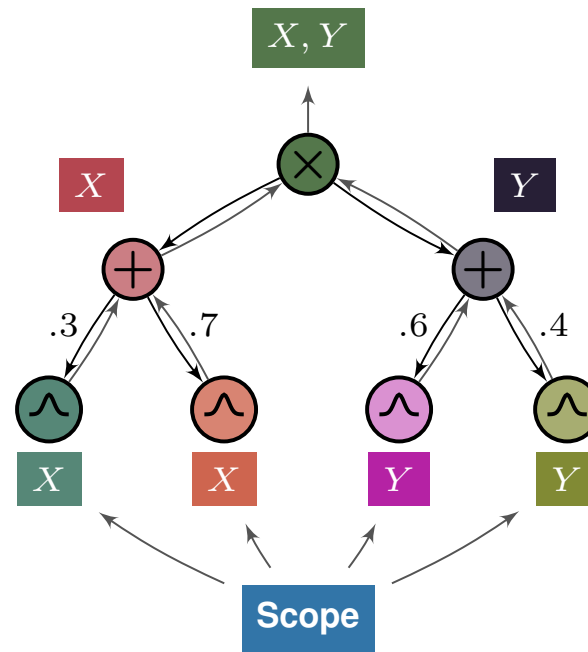
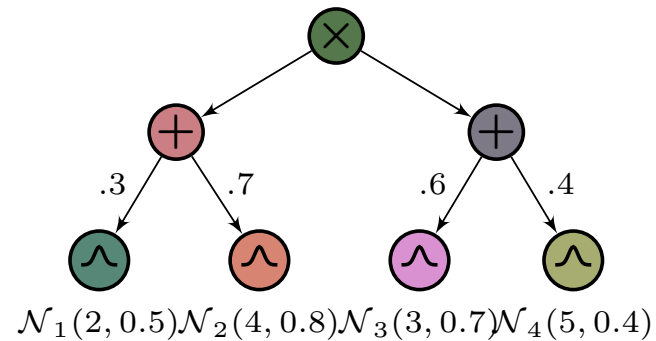




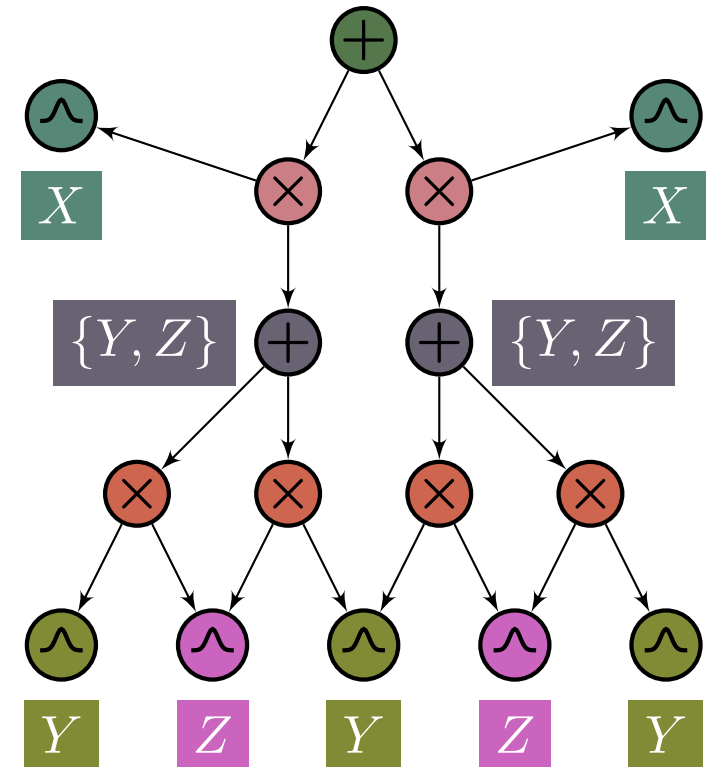
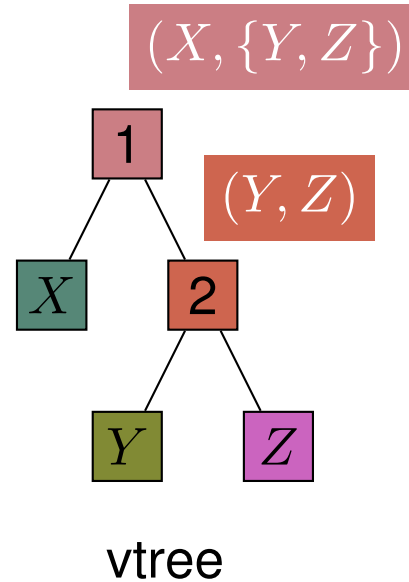
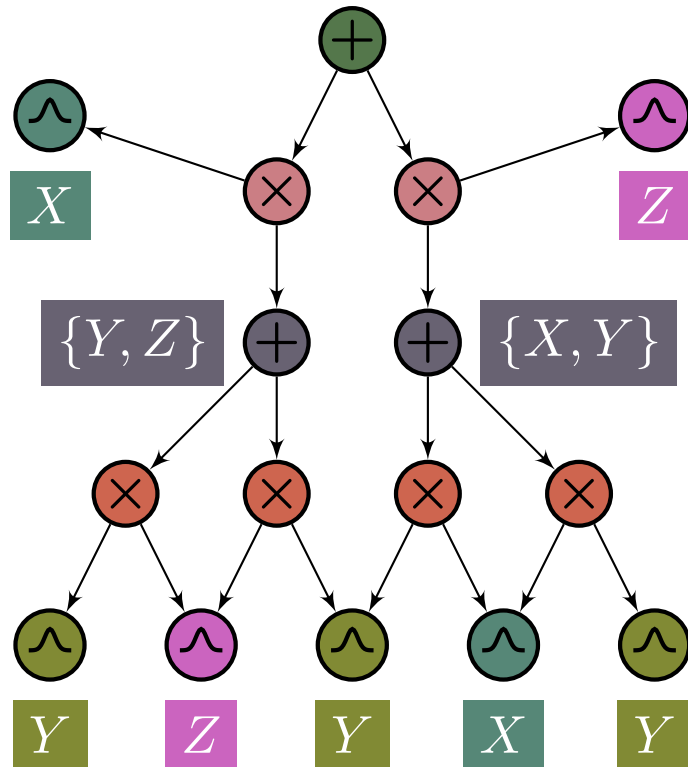
# Probabilistic Circuits – Decomposability



**Definition 3** (Decomposability).  
*Every product node child mentions different variables.*



# Probabilistic Circuits – Structured Decomposability



**Definition 4** (Structured decomposability). *Every product node follows a vtree decomposition.*

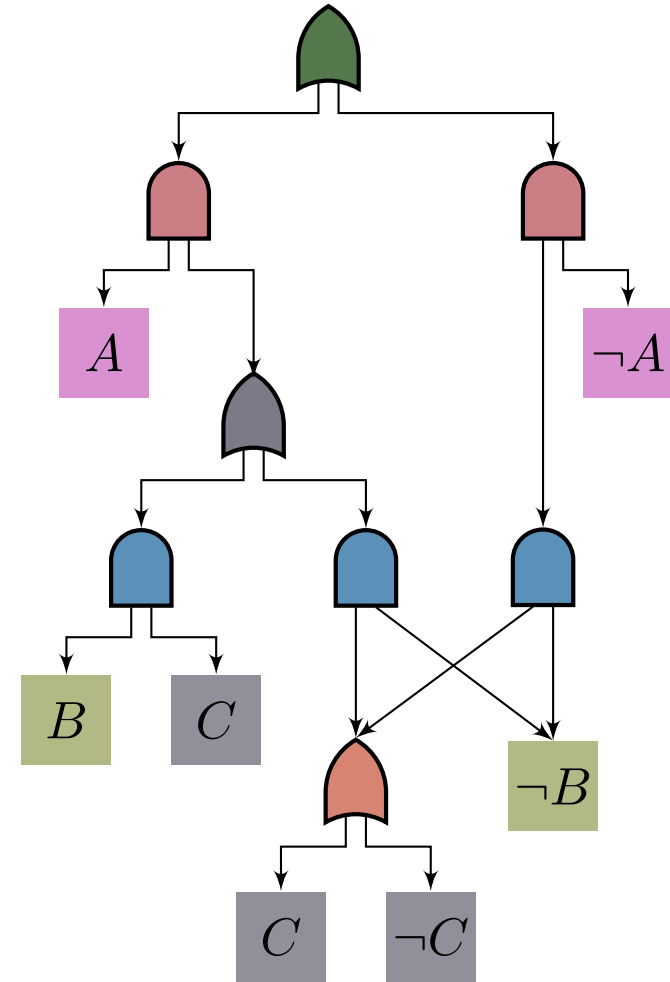
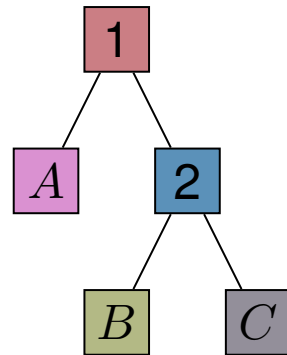
# Probabilistic Circuits – Tractability

Query	+Sm?	+Dec?	+Det?	+Str Dec?
Evidence	✓	✓	✓	✓
Marginals	✗	✓	✓	✓
Conditionals	✗	✓	✓	✓
MPE	✗	✗	✓	✓
Shannon Entropy*	✗	✗	✓	✓
Rényi Entropy*	✗	✗	✓	✓
Cross Entropy*	✗	✗	✗	✓
Kullback-Leibler Div*	✗	✗	✗	✓
Rényi's Alpha Div*	✗	✗	✗	✓
Cauchy-Schwarz Div*	✗	✗	✗	✓
Logical Events	✗	✗	✗	✓
Mutual Information*	✗	✗	✗	✓

# Probabilistic Circuits – Logic Circuits

$A$	$B$	$C$	$\phi(\mathbf{x})$
0	0	0	1
1	0	0	1
0	1	0	0
1	1	0	0
0	0	1	1
1	0	1	1
0	1	1	0
1	1	1	1

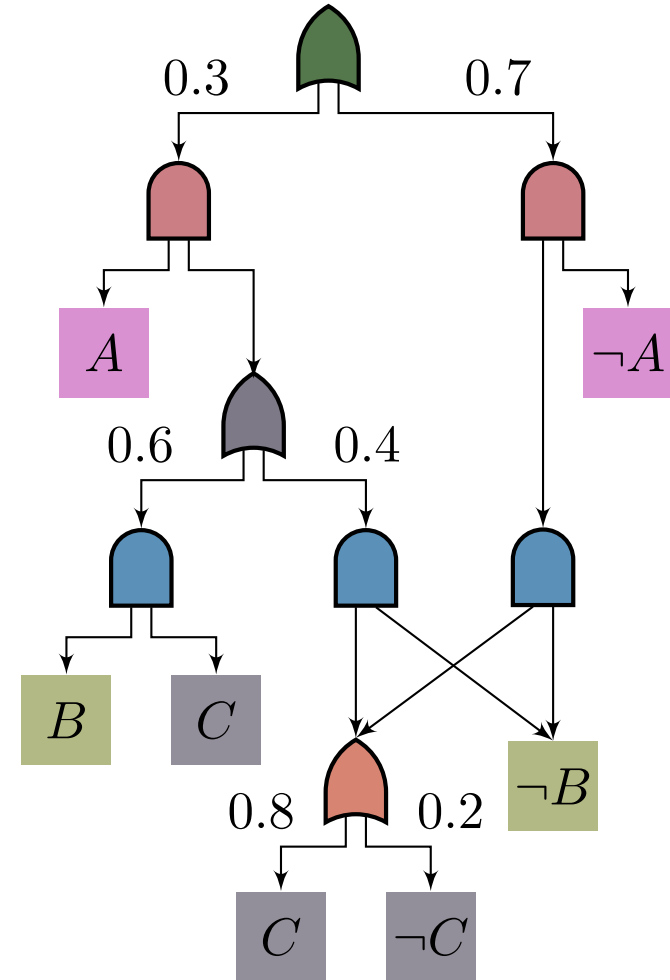
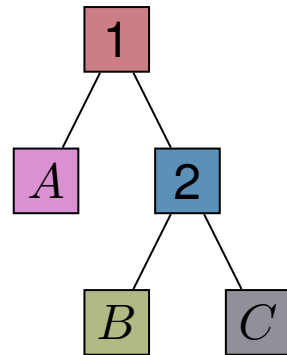
$$\phi(A, B, C) = (A \vee B) \wedge (\neg B \vee C)$$



# Probabilistic Circuits – Support

$A$	$B$	$C$	$\phi(\mathbf{x})$	$p(\mathbf{x})$
0	0	0	1	0.140
1	0	0	1	0.024
0	1	0	0	0.000
1	1	0	0	0.000
0	0	1	1	0.560
1	0	1	1	0.096
0	1	1	0	0.000
1	1	1	1	0.180

$$\phi(A, B, C) = (A \vee B) \wedge (\neg B \vee C)$$



# Learning Probabilistic Circuits

## Divide-and-Conquer Approaches (DIV)

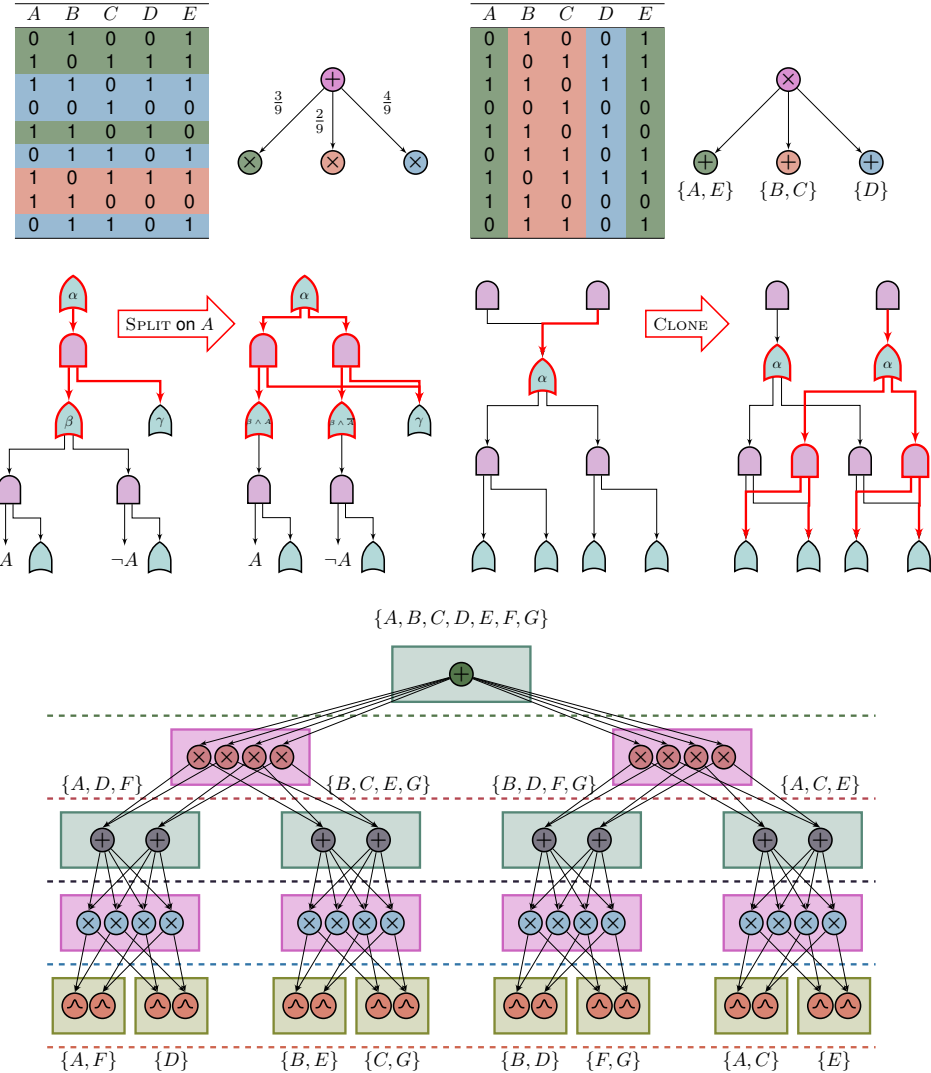
- Usually recursive;
- Splits data by similarity and stat dep;
- Stat dep usually costly;
- Usually tree-shaped.

## Incremental Approaches (INCR)

- Requires an initial circuit;
- Grows from local transformations;
- Local transformations preserve properties;
- Searching for candidates to transform is costly.

## Random Approaches (RAND)

- Fast;
- Randomly generates circuits;
- Data blind and data guided approaches exist;
- Usually relies on many hyperparams;
- Worse performance.



# Learning Probabilistic Circuits – Where are we right now?

Name	Class	Time Complexity	# hyperparams	Accepts logic?	Sm?	Dec?	Det?	Str Dec?	$\{0, 1\}?$	$\mathbb{N}?$	$\mathbb{R}?$	Reference
LEARNSPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & , \text{ if sum} \\ \mathcal{O}(nm^3) & , \text{ if product} \end{cases}$	$\geq 2$	$\times$	✓	✓	$\times$	$\times$	✓	✓	✓	Gens and Domingos [2013]
ID-SPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & , \text{ if sum} \\ \mathcal{O}(nm^3) & , \text{ if product} \end{cases}$	$\geq 2 + 3$	$\times$	✓	✓	$\times$	$\times$	✓	✓	$\times$	Rooshenas and Lowd [2014]
PROMETHEUS	DIV	$\begin{cases} \mathcal{O}(ic(rn + m)) & , \text{ if input} \\ \mathcal{O}(nkmc) & , \text{ if sum} \\ \mathcal{O}(m(\log m)^2) & , \text{ if product} \end{cases}$	$\geq 1$	$\times$	✓	✓	$\times$	$\times$	✓	✓	✓	Jaini et al. [2018a]
LEARNPSSD	INCR	$\begin{cases} \mathcal{O}(m^2) & , \text{ top-down vtree} \\ \mathcal{O}(m^4) & , \text{ bottom-up vtree} \\ \mathcal{O}(i C ^2) & , \text{ circuit structure} \end{cases}$	1	✓	✓	✓	✓	✓	✓	$\times$	$\times$	Liang et al. [2017]
STRUDEL	INCR	$\begin{cases} \mathcal{O}(m^2n) & , \text{ CLT + vtree} \\ \mathcal{O}(i( C n + m^2)) & , \text{ circuit structure} \end{cases}$	1	✓	✓	✓	✓	✓	✓	$\times$	$\times$	Dang et al. [2020]
RAT-SPN	RAND	$\mathcal{O}(rd(s + l))$	4	$\times$	✓	✓	$\times$	$\times$	✓	✓	✓	Peharz et al. [2020]
XPC	RAND	$\mathcal{O}(i(t + kn) + ikm^2n)$	3	$\times$	✓	✓	✓	✓	✓	$\times$	$\times$	Mauro et al. [2021]
$\Rightarrow$ SAMPLEPSDD	RAND	$\begin{cases} \mathcal{O}(m) & , \text{ random vtree} \\ \mathcal{O}(kc \log c + \log_2^2 k) & , \text{ per call} \end{cases}$	1	✓	✓	✓	✓	✓	✓	$\times$	$\times$	Geh and Mauá [2021]
LEARNRP	RAND	$\begin{cases} \mathcal{O}(m^2) & , \text{ top-down vtree} \\ \mathcal{O}(m^4) & , \text{ bottom-up vtree} \\ \mathcal{O}(knm) & , \text{ per call} \end{cases}$	0	$\times$	✓	✓	$\times$	✓	✓	✓	✓	To appear

# A Logical Perspective



# Motivation



## Example:

$$n = 3, k = 3$$

$X_{11}$	$X_{12}$	$X_{13}$	$X_{21}$	$\dots$	$X_{33}$	$p(\mathbf{x}) > 0$
0	0	0	0	0	0	0
1	0	0	0	0	0	0
0	1	0	0	0	0	0
1	1	0	0	0	0	0
0	0	1	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	1	1	1	1	1	0
1	1	1	1	1	1	0

Assignments:  $2^{3 \cdot 3} = 512$

Positive assignments:  $3! = 6$

If we assume

$n$  sushi types,

$k$  sized rankings with  $k \leq n$ ,

$X_{ij}$  binary variables;  $i$  is sushi type,  $j$  is position in ranking;

then the total number of possible assignments of the  $n \cdot k$  variables is  $2^{nk}$ ...

...but many of these are zero probability assignments!

If we can embed total ranking constraints...

...we go down to  $k!$  total assignments!

**Takeaway:** models which exploit domain knowledge are much more efficient!

# Motivation

## Existing approaches:

LEARNPSDD (Liang et al. [2017]):

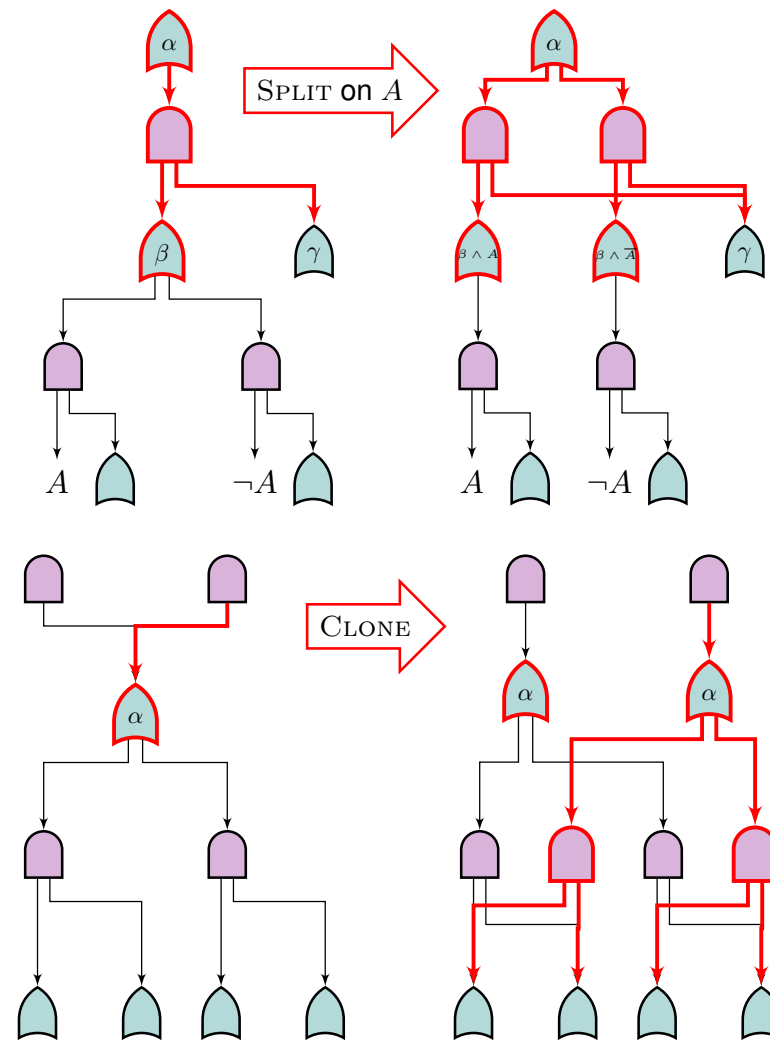
- ✗ Requires initial logic circuit encoding the support...
- ✗ Scales poorly to complex formulae and/or high dimension...
- ✗ Costly whole circuit evaluation at every iteration...
- ✓ Very good performance!

STRUDEL (Dang et al. [2020]):

- ✓ Constructs an initial structure (from a CLT)!
- ✗ But does not encode constraints...
- ✓ Scales to high dimension!
- ✗ As long as the circuit doesn't get too big...

SAMPLEPSDD (Geh and Mauá [2021]):

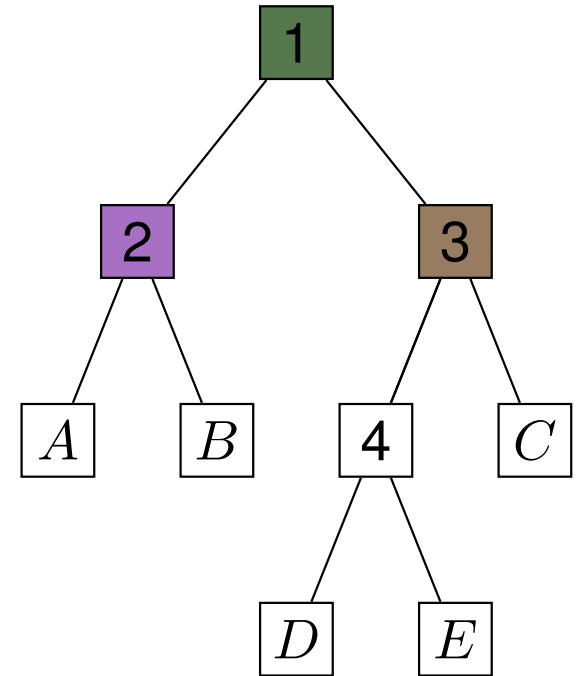
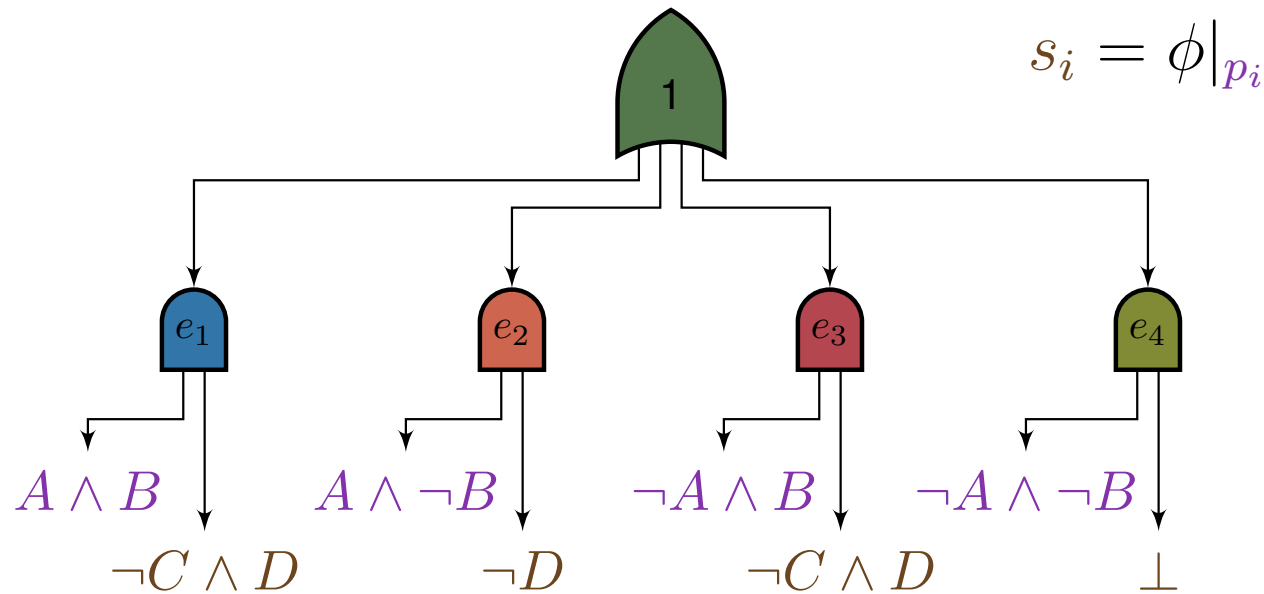
- ✓ Scales to high dimension and complex formulae!
- ✓ Constructs a structure consistent with constraints!
- ✗ But does so by relaxing the formula...
- ✗ Performance varies on set bounds and vtree structure...



# SAMPLEPSDD

Common assumption:  $p_i$  are conjunctions of literals.

$$\phi(A, B, C, D) = (A \wedge \neg B \wedge \neg D) \vee (B \wedge \neg C \wedge D)$$

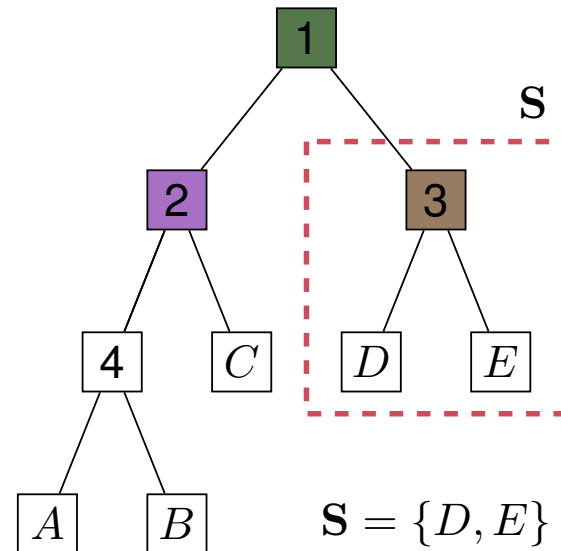
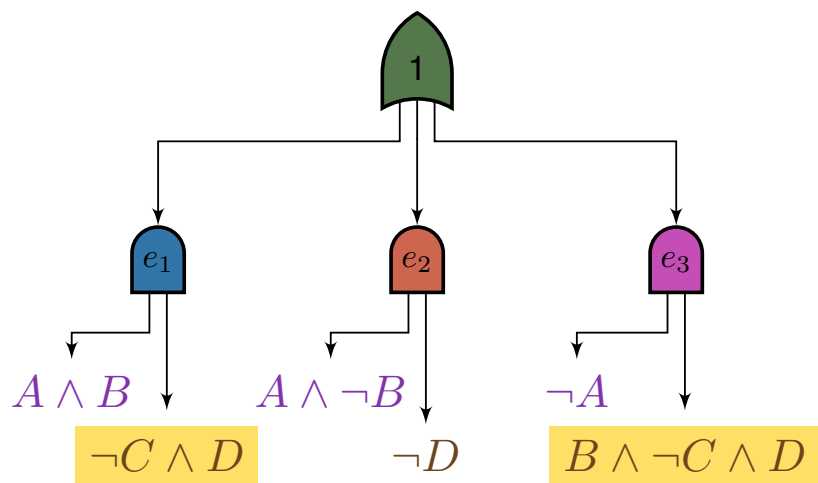


**Problem:** size of circuit is **exponential** in the size of  $p_i$ 's scope.

# SAMPLEPSDD

**Solution:** randomly sample a bounded number ( $k$ ) of  $p_i$

$$\phi(A, B, C, D) = (A \wedge \neg B \wedge \neg D) \vee (B \wedge \neg C \wedge D)$$



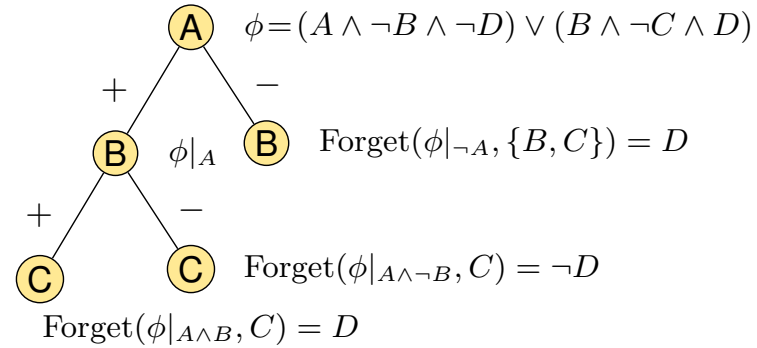
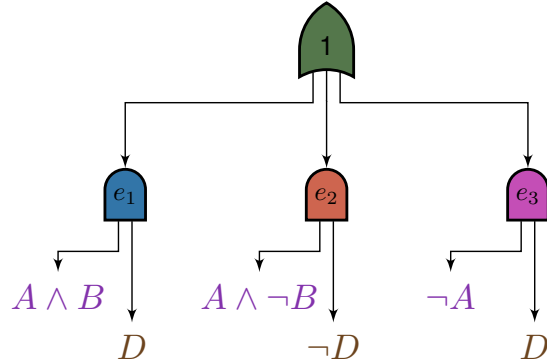
**But:** this violates structured decomposability:

$\neg C \wedge D$  contains  $C$ , and  $C \notin S$

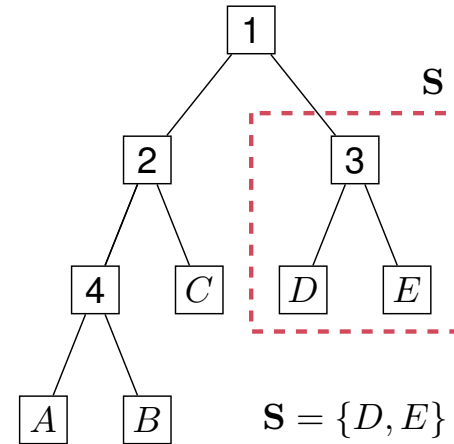
$\neg B \wedge \neg C \wedge D$  contains  $B$  and  $C$ , and  $B, C \notin S$

# SAMPLEPSDD

**New solution:** relax logical constraints  $\phi$

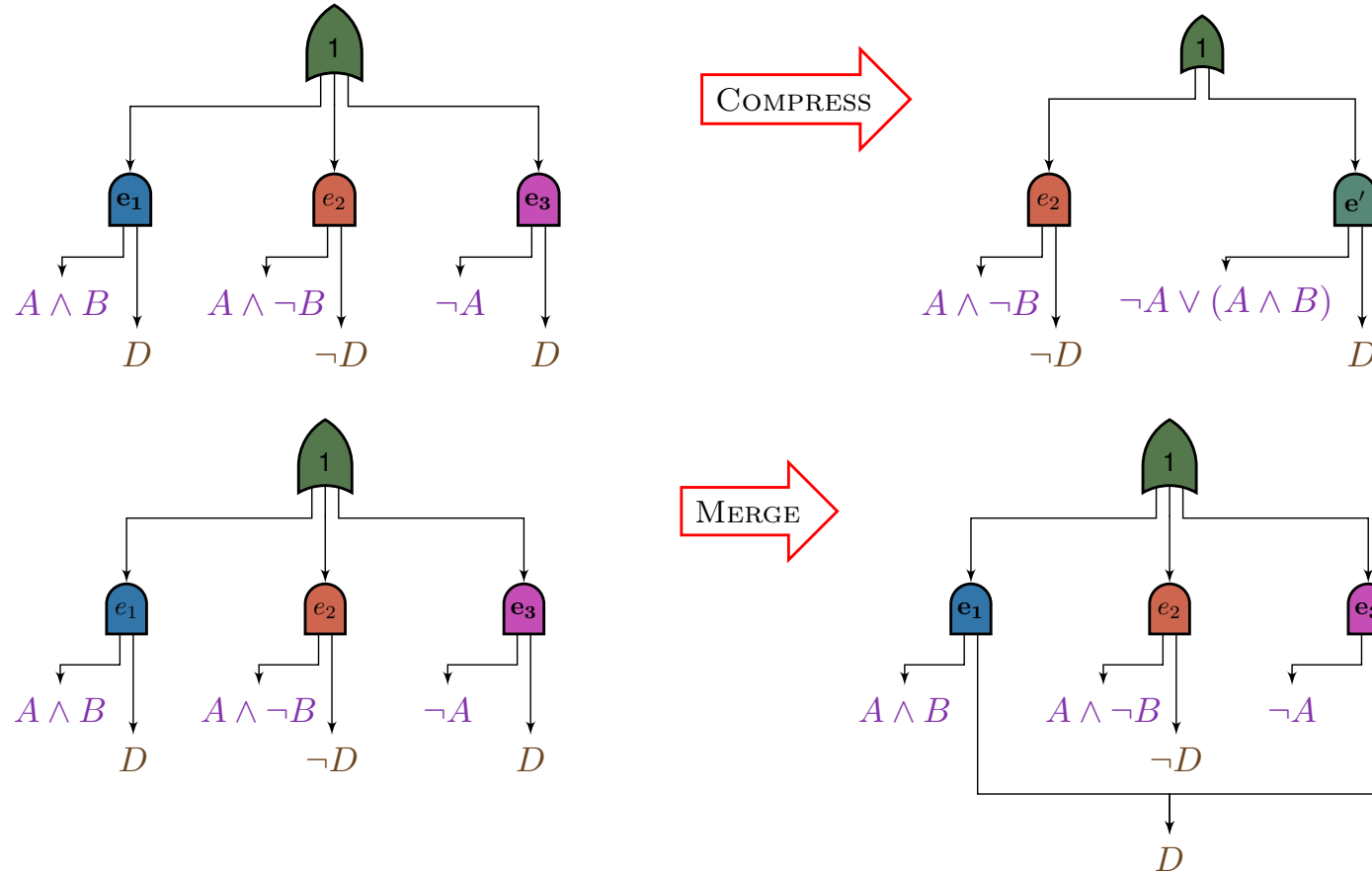


Now all  $s_i$  respect  $S$



# SAMPLEPSDD

Apply **local transformations** for variety and size reduction



# Experiments

**Evaluation:** we sample 30 PSDDs and use 5 ensemble strategies:

- Likelihood weighting (LLW),
- Uniform weights,
- ◆ Expectation-Maximization (EM),
- ▲ Stacking,
- ▼ Bayesian Model Combination (BMC);

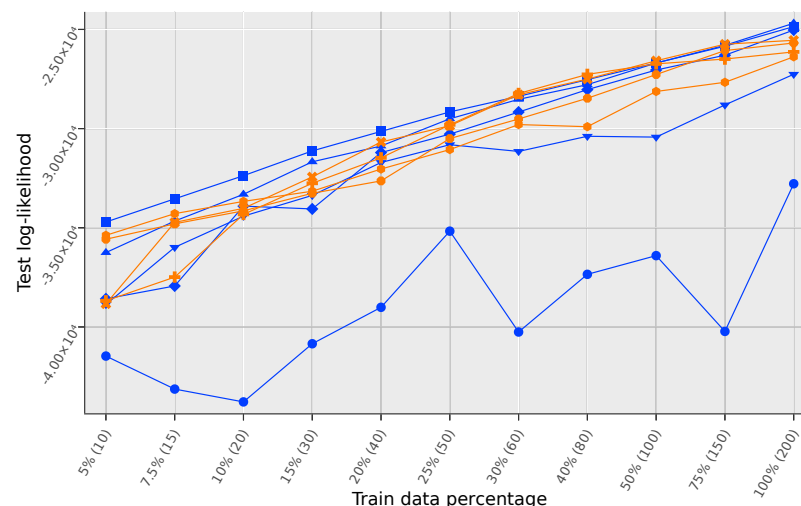
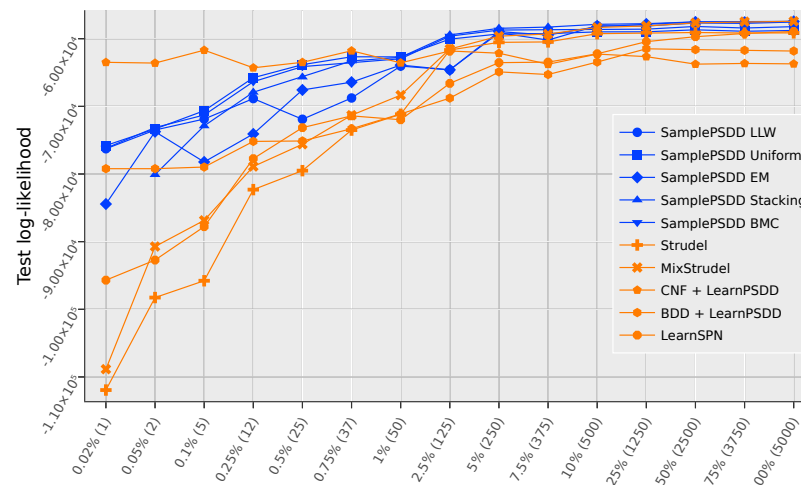
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

**Datasets:** we evaluate with 5 data + knowledge as logic constraints:

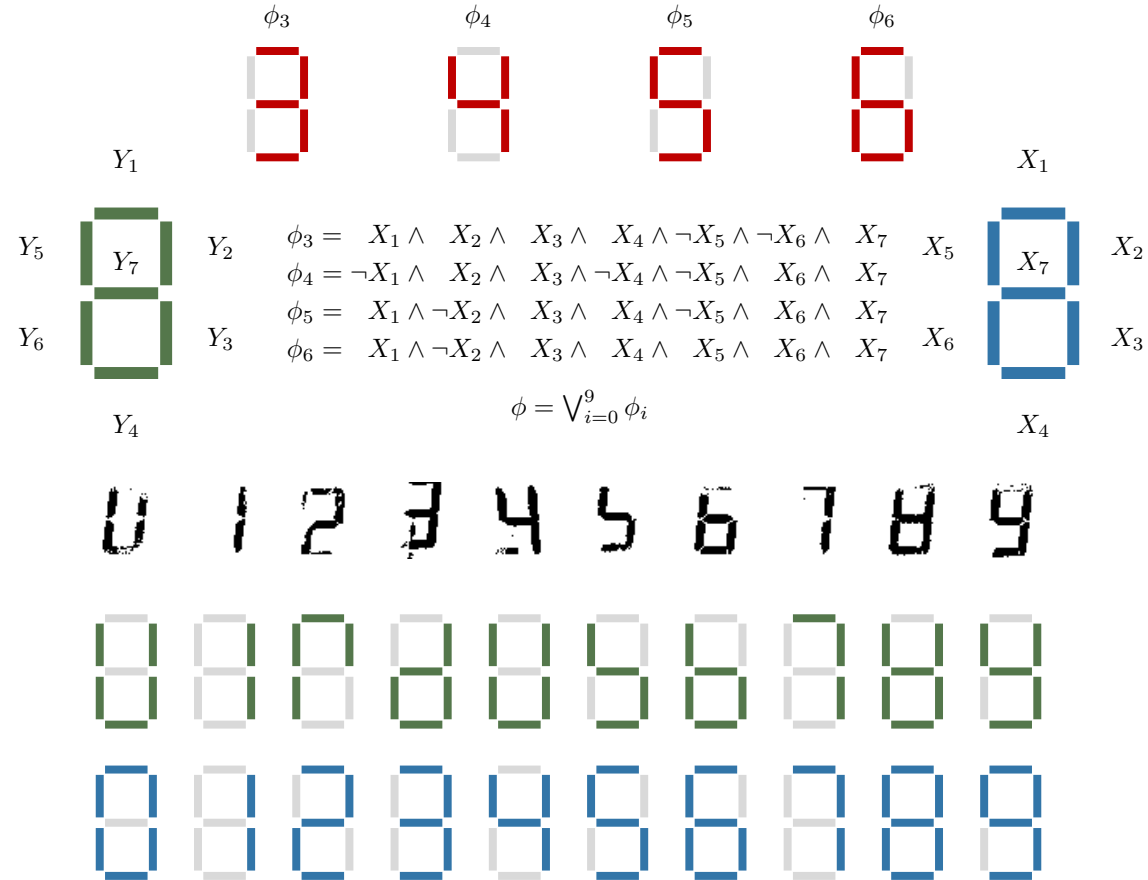
	Dataset	#vars	#train	$\phi$ 's size
⇒	LED	14	5000	23
⇒	LED + IMAGES	157	700	39899
	SUSHI RANKING	100	3500	17413
	SUSHI TOP 5	10	3500	37
	DOTA 2 GAMES	227	92650	1308

Our approach fares **better** with **fewer** data, yet  
remains **competitive** under **lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],  
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]



# Experiments – LED





# Experiments

**Evaluation:** we sample 30 PSDDs and use 5 ensemble strategies:

- Likelihood weighting (LLW),
- Uniform weights,
- ◆ Expectation-Maximization (EM),
- ▲ Stacking,
- ▼ Bayesian Model Combination (BMC);

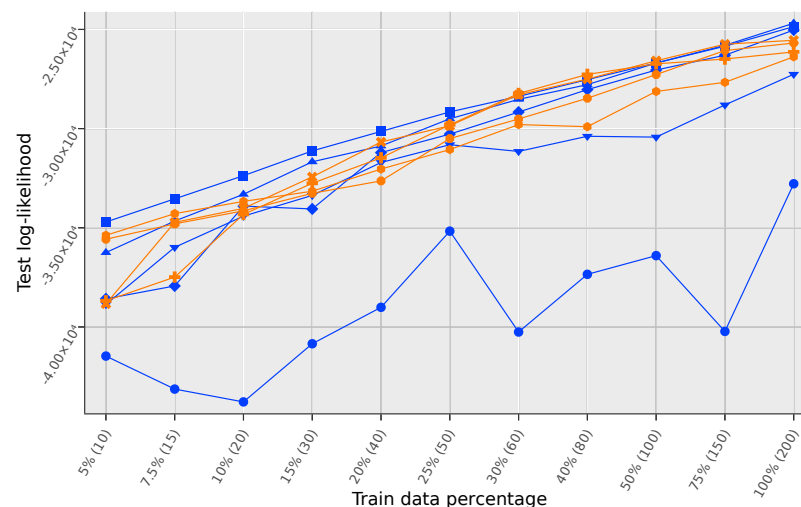
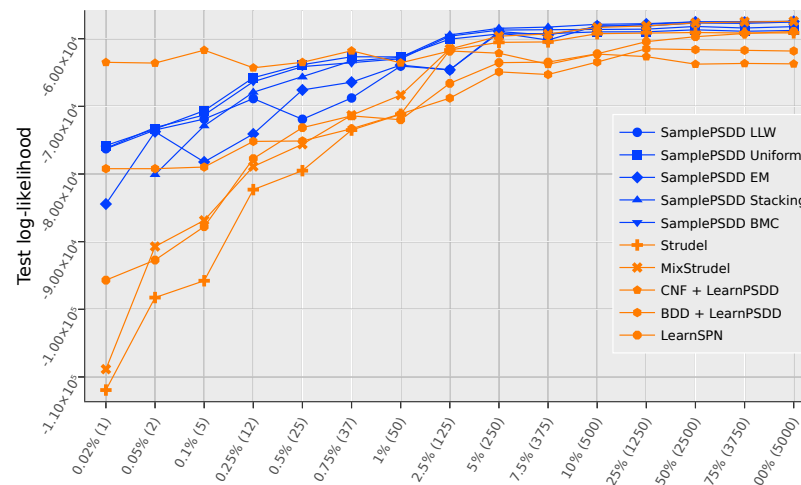
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

**Datasets:** we evaluate with 5 data + knowledge as logic constraints:

	Dataset	#vars	#train	$\phi$ 's size
⇒	LED	14	5000	23
⇒	LED + IMAGES	157	700	39899
	SUSHI RANKING	100	3500	17413
	SUSHI TOP 5	10	3500	37
	DOTA 2 GAMES	227	92650	1308

Our approach fares **better** with **fewer** data, yet  
remains **competitive** under **lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],  
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]



# Experiments

**Evaluation:** we sample 30 PSDDs and use 5 ensemble strategies:

- Likelihood weighting (LLW),
- Uniform weights,
- ◆ Expectation-Maximization (EM),
- ▲ Stacking,
- ▼ Bayesian Model Combination (BMC):

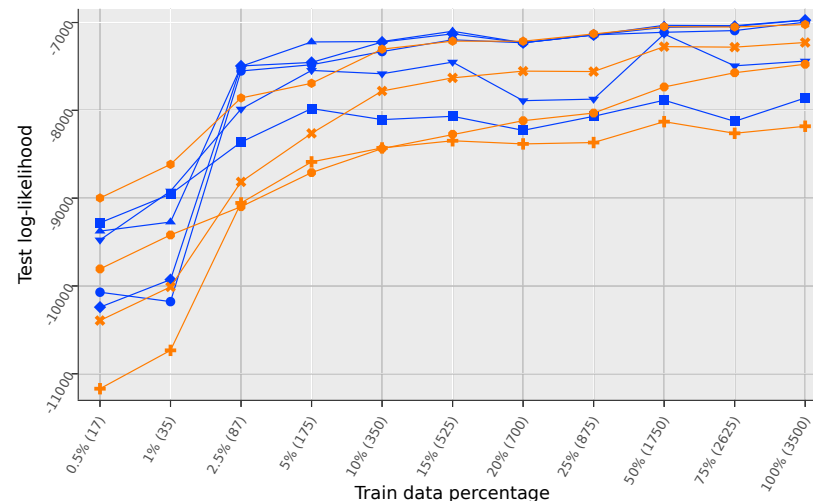
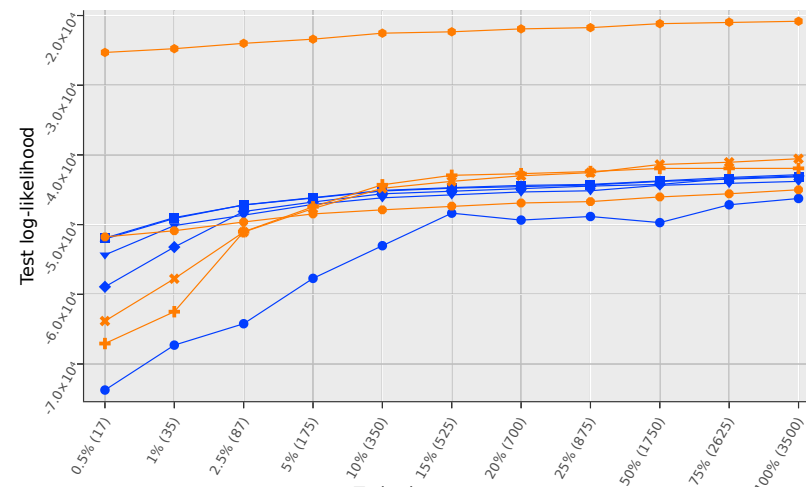
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

**Datasets:** we evaluate with 5 data + knowledge as logic constraints:

Dataset	#vars	#train	$\phi$ 's size
LED	14	5000	23
LED + IMAGES	157	700	39899
⇒ SUSHI RANKING	100	3500	17413
⇒ SUSHI TOP 5	10	3500	37
DOTA 2 GAMES	227	92650	1308

Our approach fares **better** with **fewer** data, yet remains **competitive** under **lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017], Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]



# Experiments – SUSHI RANKING



$n$  sushi types and  $k$  rank positions

$$\alpha = \underbrace{\begin{pmatrix} X_{i1} \wedge \neg X_{i2} \wedge \cdots \wedge \neg X_{ik} \\ \vee (\neg X_{i1} \wedge X_{i2} \wedge \cdots \wedge \neg X_{ik}) \\ \vdots \\ \vee (\neg X_{i1} \wedge \neg X_{i2} \wedge \cdots \wedge X_{ik}) \end{pmatrix}}_{\text{Rank position}}$$

$$\beta = \underbrace{\begin{pmatrix} X_{1j} \wedge \neg X_{2j} \wedge \cdots \wedge \neg X_{nj} \\ \vee (\neg X_{1j} \wedge X_{2j} \wedge \cdots \wedge \neg X_{nj}) \\ \vdots \\ \vee (\neg X_{1j} \wedge \neg X_{2j} \wedge \cdots \wedge X_{nj}) \end{pmatrix}}_{\text{Type uniqueness}}$$

$$\phi = \alpha \wedge \beta$$

# Experiments – SUSHI TOP 5



$n$  sushi types and  $k$  rank positions

Top  $k$  out of  $n$  sushi  $\equiv$   $n$ -choose- $k$  model  
 $n$ -choose- $k$  model  $\equiv$  cardinality Exactly( $k, n$ )

$$\phi = \text{Exactly}(k, n) = \left( \sum_X X = k \right)$$

# Experiments

**Evaluation:** we sample 30 PSDDs and use 5 ensemble strategies:

- Likelihood weighting (LLW),
- Uniform weights,
- ◆ Expectation-Maximization (EM),
- ▲ Stacking,
- ▼ Bayesian Model Combination (BMC);

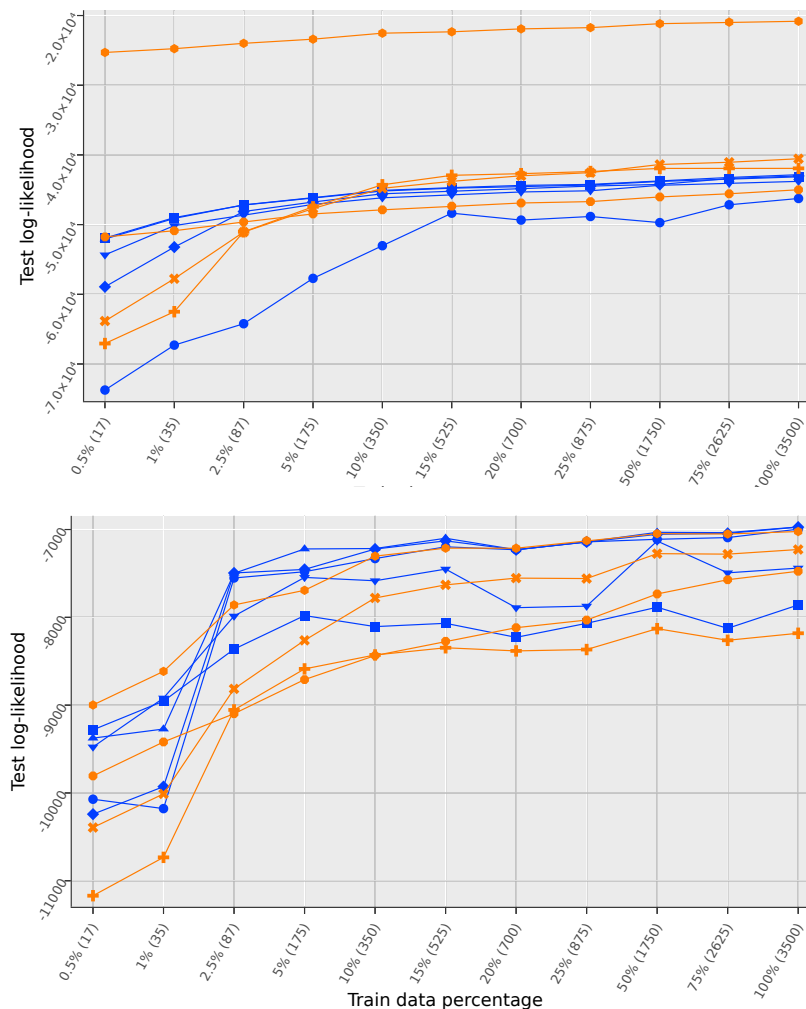
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

**Datasets:** we evaluate with 5 data + knowledge as logic constraints:

Dataset	#vars	#train	$\phi$ 's size
LED	14	5000	23
LED + IMAGES	157	700	39899
⇒ SUSHI RANKING	100	3500	17413
⇒ SUSHI TOP 5	10	3500	37
DOTA 2 GAMES	227	92650	1308

Our approach fares **better** with **fewer data**, yet  
remains **competitive** under **lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],  
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]



# Experiments

**Evaluation:** we sample 30 PSDDs and use 5 ensemble strategies:

- Likelihood weighting (LLW),
- Uniform weights,
- ◆ Expectation-Maximization (EM),
- ▲ Stacking,
- ▼ Bayesian Model Combination (BMC);

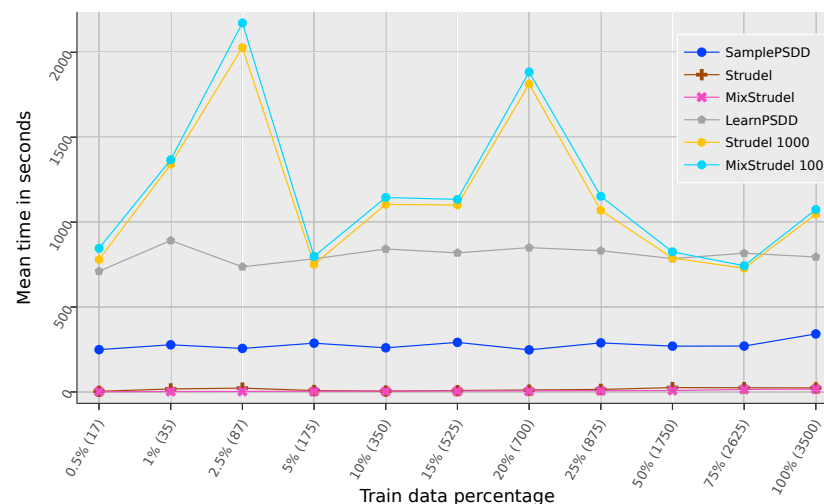
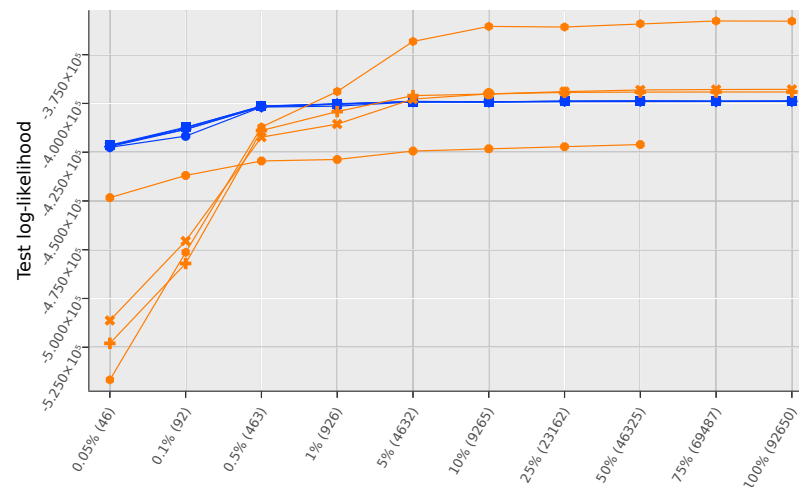
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

**Datasets:** we evaluate with 5 data + knowledge as logic constraints:

Dataset	#vars	#train	$\phi$ 's size
LED	14	5000	23
LED + IMAGES	157	700	39899
SUSHI RANKING	100	3500	17413
SUSHI TOP 5	10	3500	37
⇒ DOTA 2 GAMES	227	92650	1308

Our approach fares **better** with **fewer** data, yet  
remains **competitive** under **lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],  
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]



# Experiments – DOTA 2 GAMES

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

$X_1$	$X_2$	$X_3$	$X_4$	$X_5$
4	11	17	23	28

$\alpha = \text{Exactly}(k, n)$   
Intractable as CNF

$n$  characters,  $k$  for each team

$Y_5$	$Y_4$	$Y_3$	$Y_2$	$Y_1$
25	20	13	8	5

$\beta = \text{Exactly}(k, n)$   
Intractable as CNF

$\gamma = X_i \neq Y_j, \forall X_i, Y_j$   
Intractable as BDD

$$\phi = \alpha \wedge \beta \wedge \gamma$$

# Experiments

**Evaluation:** we sample 30 PSDDs and use 5 ensemble strategies:

- Likelihood weighting (LLW),
- Uniform weights,
- ◆ Expectation-Maximization (EM),
- ▲ Stacking,
- ▼ Bayesian Model Combination (BMC);

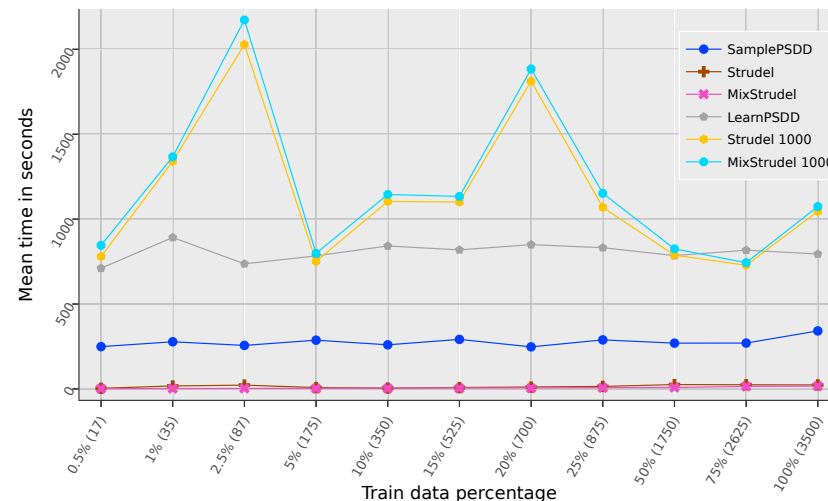
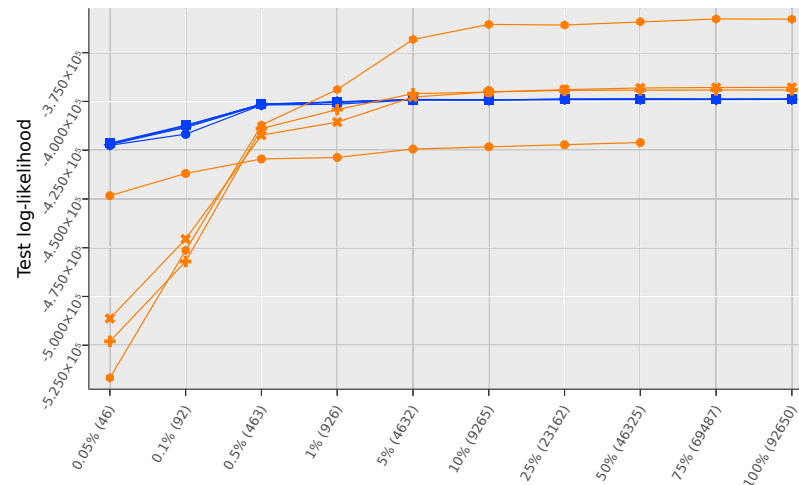
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

**Datasets:** we evaluate with 5 data + knowledge as logic constraints:

Dataset	#vars	#train	$\phi$ 's size
LED	14	5000	23
LED + IMAGES	157	700	39899
SUSHI RANKING	100	3500	17413
SUSHI TOP 5	10	3500	37
⇒ DOTA 2 GAMES	227	92650	1308

Our approach fares **better** with **fewer** data, yet  
remains **competitive** under **lots** of data.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],  
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]

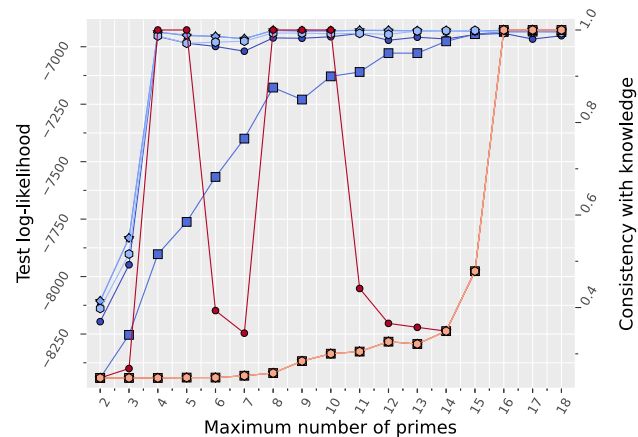
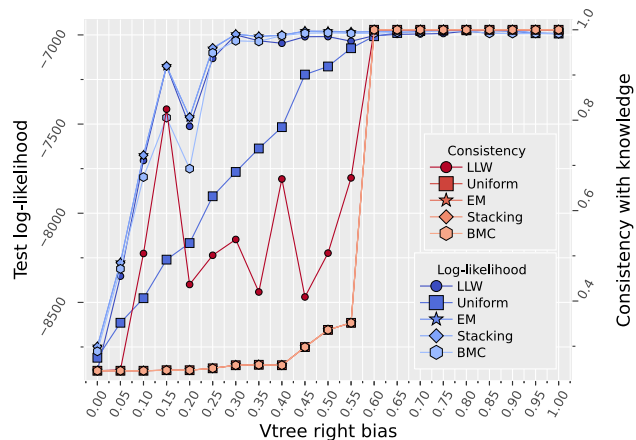




# SAMPLEPSDD – Experiments

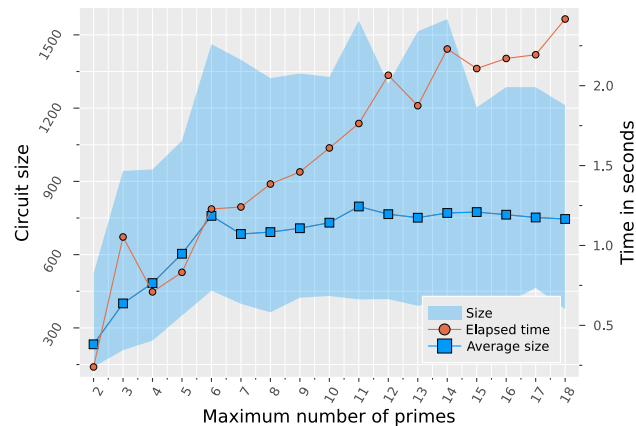
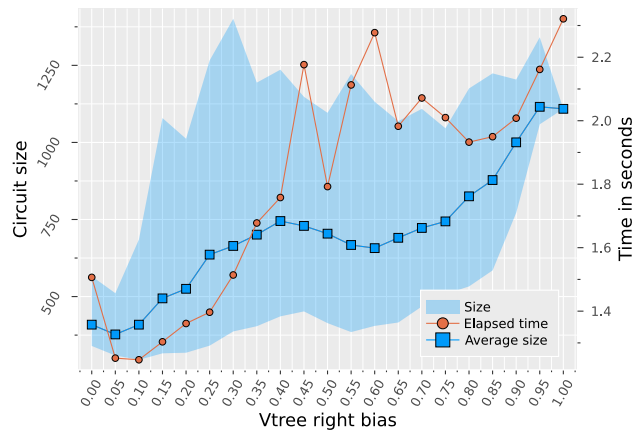
What is the impact of higher  $k$ 's and right-leaning vtrees

in log-likelihood and consistency?



Samples perform better with higher  $k$ 's and right-leaning vtrees ...

...but at a cost to complexity.



# Learning Probabilistic Circuits – Where are we right now?

Name	Class	Time Complexity	# hyperparams	Accepts logic?	Sm?	Dec?	Det?	Str Dec?	$\{0, 1\}$ ?	$\mathbb{N}$ ?	$\mathbb{R}$ ?	Reference
LEARNSPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & , \text{ if sum} \\ \mathcal{O}(nm^3) & , \text{ if product} \end{cases}$	$\geq 2$	✗	✓	✓	✗	✗	✓	✓	✓	Gens and Domingos [2013]
ID-SPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & , \text{ if sum} \\ \mathcal{O}(nm^3) & , \text{ if product} \\ \mathcal{O}(ic(rn + m)) & , \text{ if input} \end{cases}$	$\geq 2 + 3$	✗	✓	✓	✗	✗	✓	✓	✗	Rooshenas and Lowd [2014]
PROMETHEUS	DIV	$\begin{cases} \mathcal{O}(nkmc) & , \text{ if sum} \\ \mathcal{O}(m(\log m)^2) & , \text{ if product} \end{cases}$	$\geq 1$	✗	✓	✓	✗	✗	✓	✓	✓	Jaini et al. [2018a]
LEARNPSSD	INCR	$\begin{cases} \mathcal{O}(m^2) & , \text{ top-down vtree} \\ \mathcal{O}(m^4) & , \text{ bottom-up vtree} \\ \mathcal{O}(i \mathcal{C} ^2) & , \text{ circuit structure} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Liang et al. [2017]
STRUDEL	INCR	$\begin{cases} \mathcal{O}(m^2n) & , \text{ CLT + vtree} \\ \mathcal{O}(i( \mathcal{C} n + m^2)) & , \text{ circuit structure} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Dang et al. [2020]
RAT-SPN	RAND	$\mathcal{O}(rd(s + l))$	4	✗	✓	✓	✗	✗	✓	✓	✓	Peharz et al. [2020]
XPC	RAND	$\mathcal{O}(i(t + kn) + ikm^2n)$	3	✗	✓	✓	✓	✓	✓	✗	✗	Mauro et al. [2021]
SAMPLEPSSD	RAND	$\begin{cases} \mathcal{O}(m) & , \text{ random vtree} \\ \mathcal{O}(kc \log c + \log_2^2 k) & , \text{ per call} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Geh and Mauá [2021]
$\Rightarrow$ LEARNRP	RAND	$\begin{cases} \mathcal{O}(m^2) & , \text{ top-down vtree} \\ \mathcal{O}(m^4) & , \text{ bottom-up vtree} \\ \mathcal{O}(knm) & , \text{ per call} \end{cases}$	0	✗	✓	✓	✗	✓	✓	✓	✓	To appear

# A Data Perspective

# Motivation

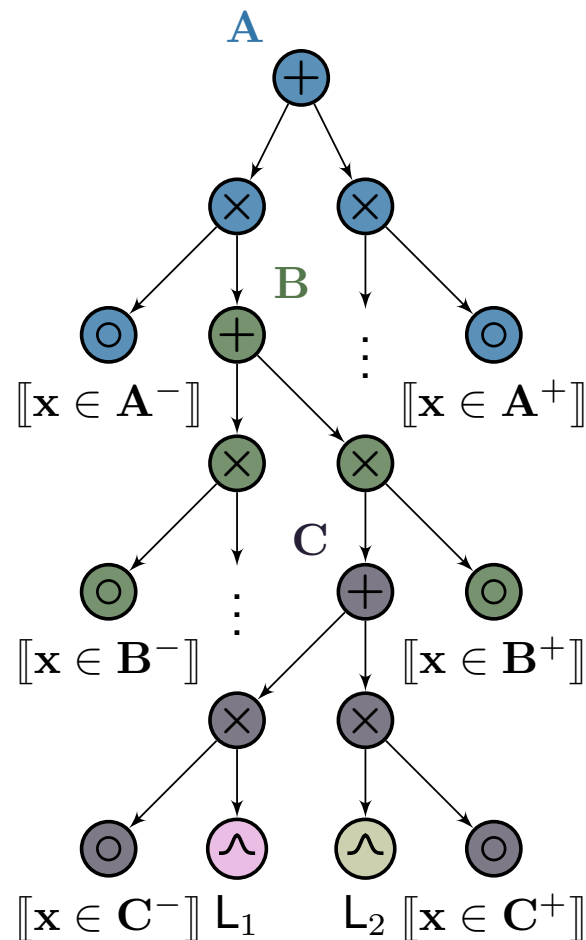
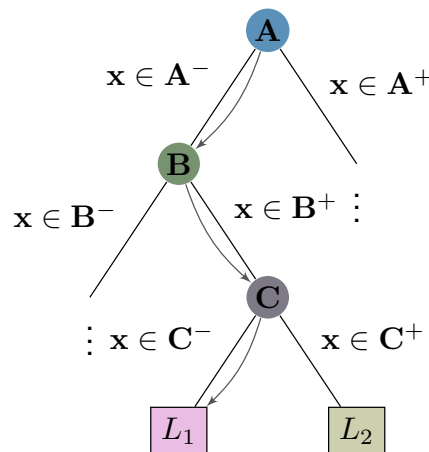
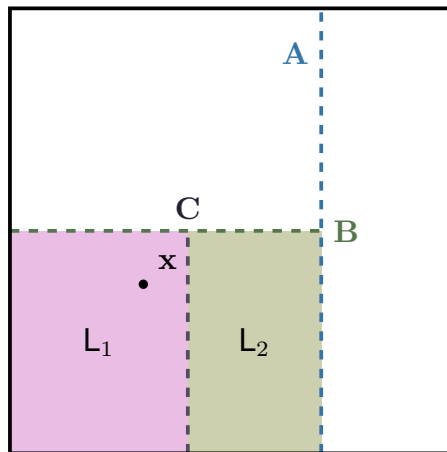
## Density Estimation Trees...

- ✓ ...are fast;
- ✓ ...are interpretable;
- ✓ ...are (somewhat) explainable;
- ✓ ...have extensive literature coverage;
- ✗ ...are not so expressive;
- ✗ ...only accept marginalization queries;
- ✗ ...are not so accurate;

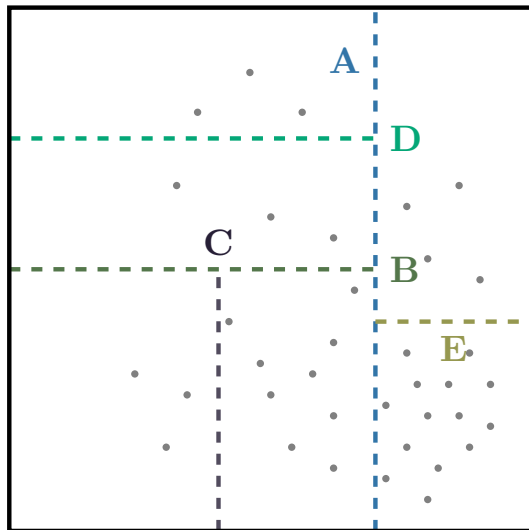
...but are subsumed by circuits!

Learn DETs  $\subseteq$  Learn PCs?

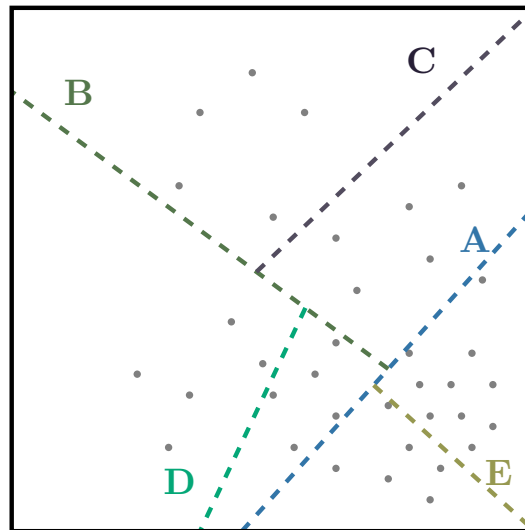
Can we take advantage of known learning procedures in DETs and transplant them to more general circuits?



# Random Projections



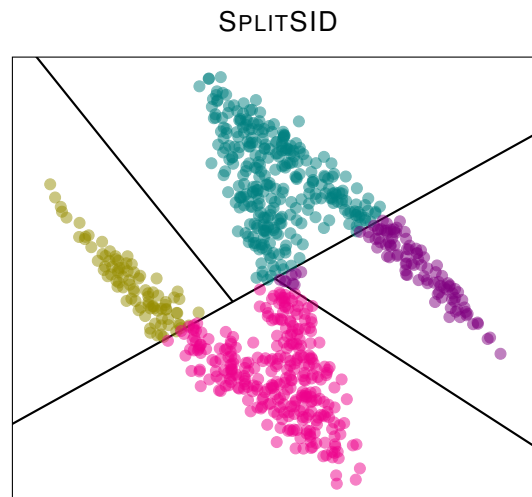
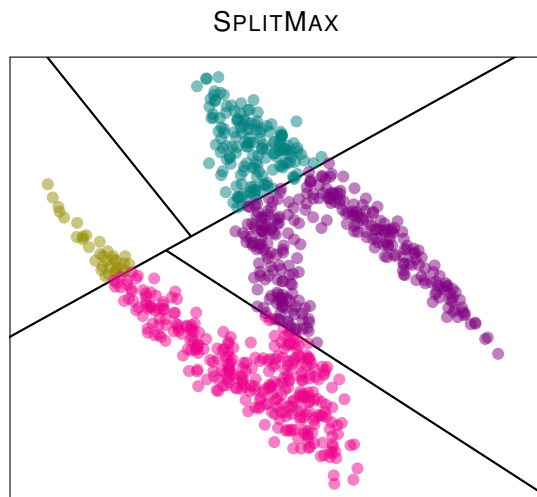
*Axis-aligned projections*



*Random projections*

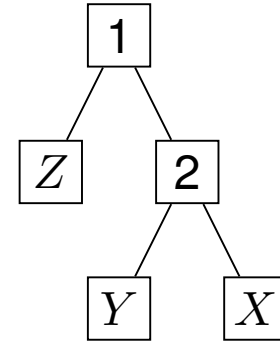
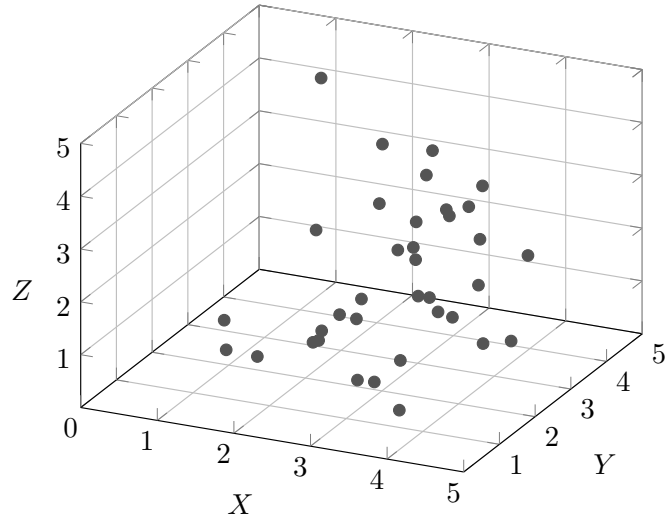
If the data has *intrinsic dimension*  $d$ , then with constant probability the part of the data at level  $d$  or higher of the tree has average diameter less than half of the data.

# Random Projections

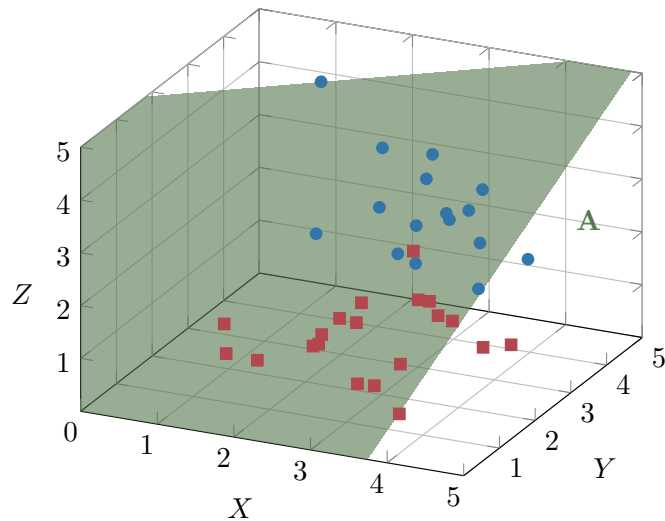


If the data has *intrinsic dimension*  $d$ , then with constant probability the part of the data at level  $d$  or higher of the tree has average diameter less than half of the data.

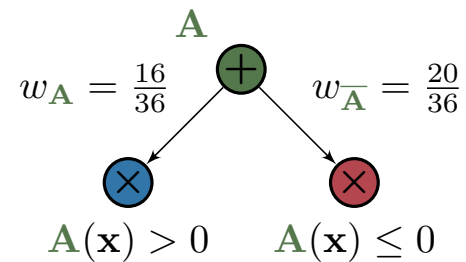
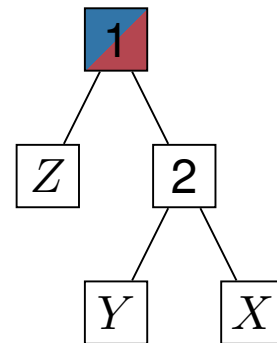
# LearnRP



# LearnRP



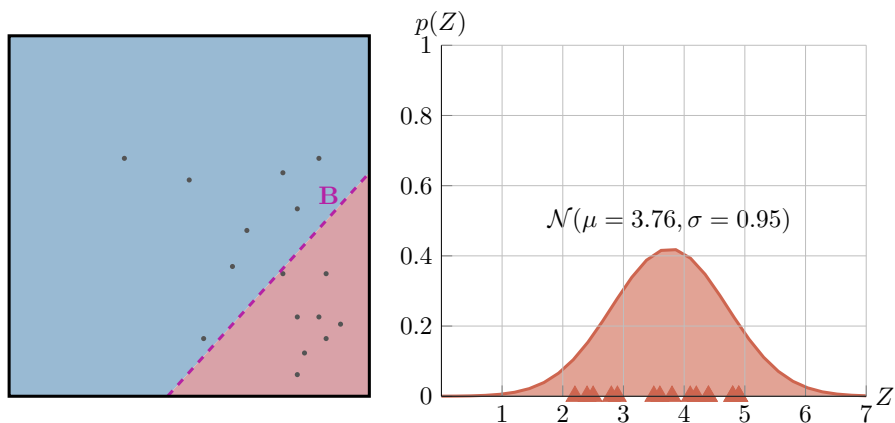
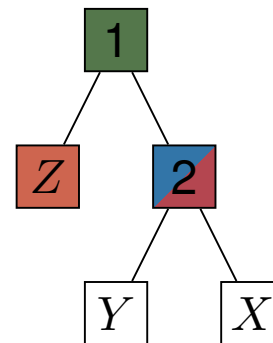
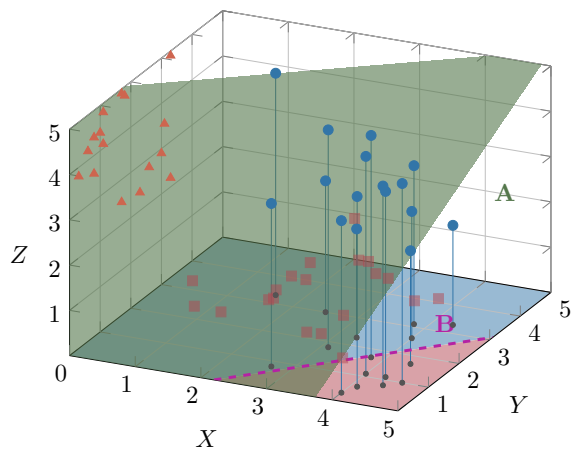
$$\mathbf{A}(x, y, z) = [x \ y \ z] \cdot \underbrace{\begin{bmatrix} -0.31 \\ -0.40 \\ 0.85 \end{bmatrix}}_a + \underbrace{1}_{\theta}$$



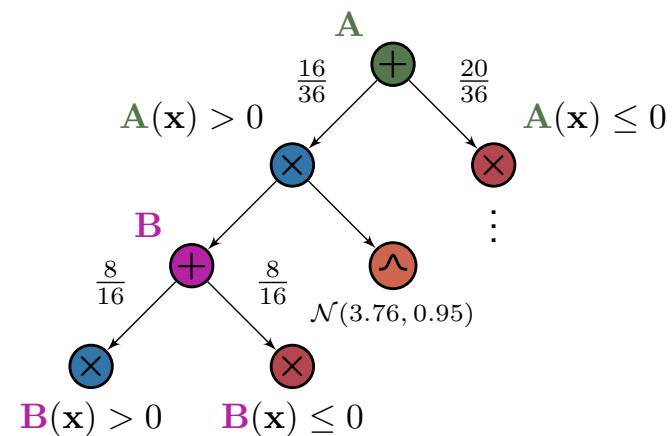
$w_{\mathbf{A}}$ : probability of  $\mathbf{A}(\mathbf{x}) > 0$



# LearnRP



$$\mathbf{B}(x, y) = [x \ y] \cdot \underbrace{\begin{bmatrix} 1.10 \\ -1.00 \end{bmatrix}}_b - \underbrace{2.43}_\gamma$$



# Parameter Optimization

## Expectation-Maximization (EM)

- Full EM (dataset  $\mathbf{D}$ )

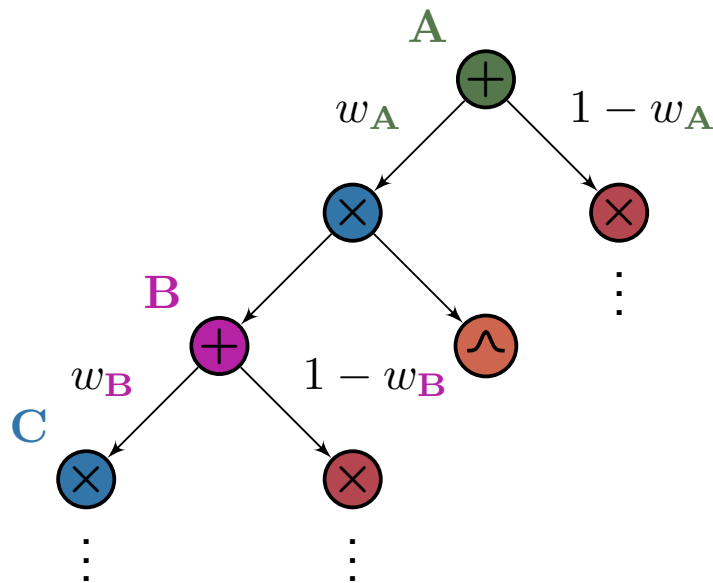
$$w_{\mathbf{B}} \propto w_{\mathbf{B}} \cdot \sum_{\mathbf{x} \in \mathbf{D}} \frac{1}{p_{\mathbf{A}}(\mathbf{x})} \cdot \frac{\partial p_{\mathbf{A}}(\mathbf{x})}{\partial p_{\mathbf{B}}(\mathbf{x})} \cdot p_{\mathbf{C}}(\mathbf{x})$$

- Minibatch EM (batch  $\mathbf{M} \subset \mathbf{D}$ )

$$w_{\mathbf{B}} \propto w_{\mathbf{B}} \cdot \sum_{\mathbf{x} \in \mathbf{M}} \frac{1}{p_{\mathbf{A}}(\mathbf{x})} \cdot \frac{\partial p_{\mathbf{A}}(\mathbf{x})}{\partial p_{\mathbf{B}}(\mathbf{x})} \cdot p_{\mathbf{C}}(\mathbf{x})$$

**LEARNRP-100:** LEARNRP + 100 itrs of minibatch

**LEARNRP-F:** LEARNRP-100 + 30 itrs of full



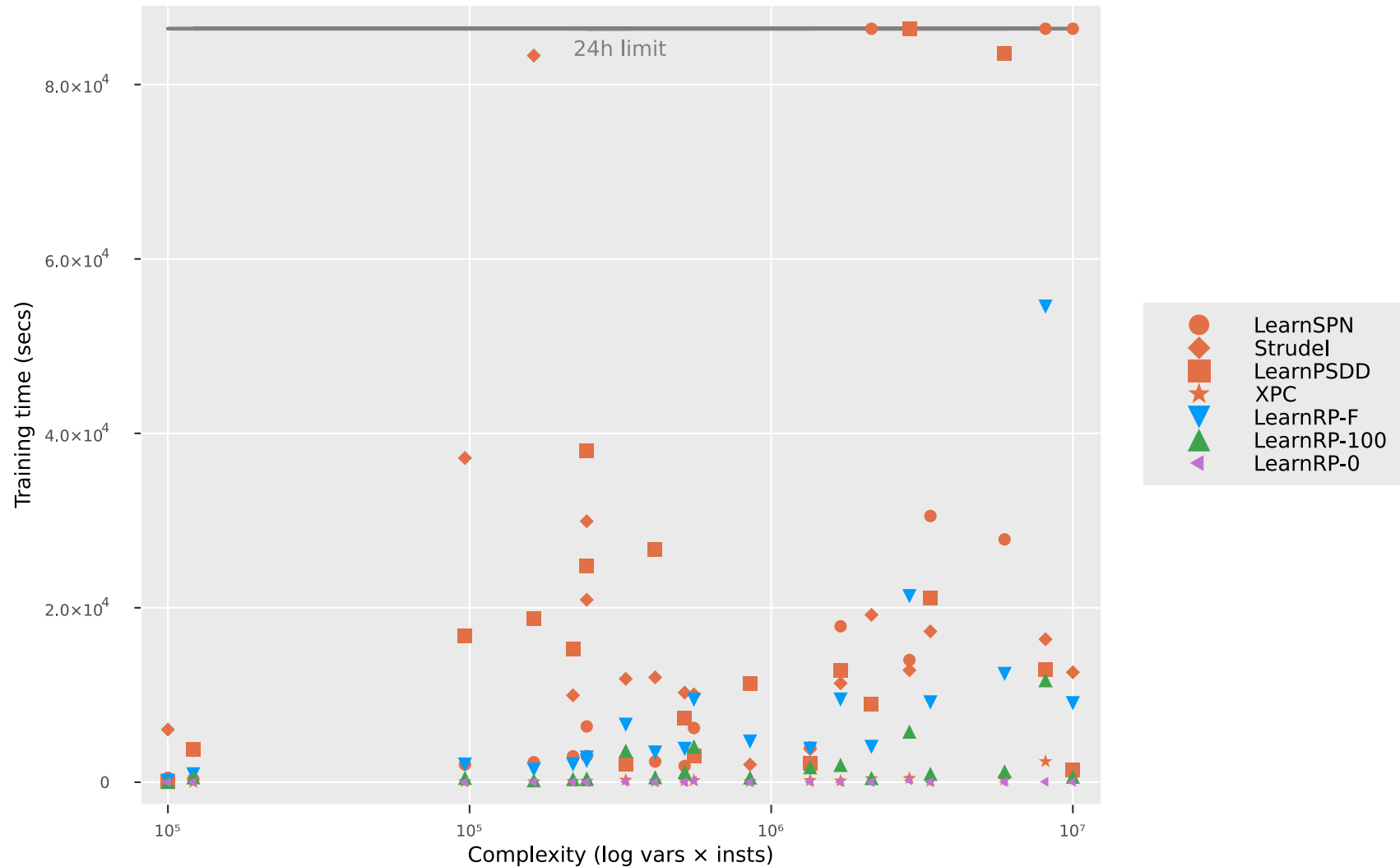
# LEARNRP – Datasets

Dataset	Vars	Train	Test	Domain	Dataset	Vars	Train	Test	Domain
ACCIDENTS	111	12758	2551	$\{0, 1\}$	NLTCS	16	16181	3236	$\{0, 1\}$
AD	1556	2461	491	$\{0, 1\}$	PLANTS	69	17412	3482	$\{0, 1\}$
AUDIO	100	15000	3000	$\{0, 1\}$	PUMSB-STAR	163	12262	2452	$\{0, 1\}$
BBC	1058	1670	330	$\{0, 1\}$	EACHMOVIE	500	4524	591	$\{0, 1\}$
NETFLIX	100	15000	3000	$\{0, 1\}$	RETAIL	135	22041	4408	$\{0, 1\}$
BOOK	500	8700	1739	$\{0, 1\}$	ABALONE	8	3760	417	$\mathbb{R}$
20-NEWSGRP	910	11293	3764	$\{0, 1\}$	CA	22	7373	819	$\mathbb{R}$
REUTERS-52	889	6532	1540	$\{0, 1\}$	QUAKE	4	1961	217	$\mathbb{R}$
WEBKB	839	2803	838	$\{0, 1\}$	SENSORLESS	48	52659	5850	$\mathbb{R}$
DNA	180	1600	1186	$\{0, 1\}$	BANKNOTE	4	1235	137	$\mathbb{R}$
JESTER	100	9000	4116	$\{0, 1\}$	FLOWSIZE	3	1358674	150963	$\mathbb{R}$
KDD	65	180092	34955	$\{0, 1\}$	KINEMATICS	8	7373	819	$\mathbb{R}$
KOSAREK	190	33375	6675	$\{0, 1\}$	IRIS	4	90	10	$\mathbb{R}$
MSNBC	17	291326	58265	$\{0, 1\}$	OLDFAITH	2	245	27	$\mathbb{R}$
MSWEB	294	29441	5000	$\{0, 1\}$	CHEMDIABET	3	131	14	$\mathbb{R}$

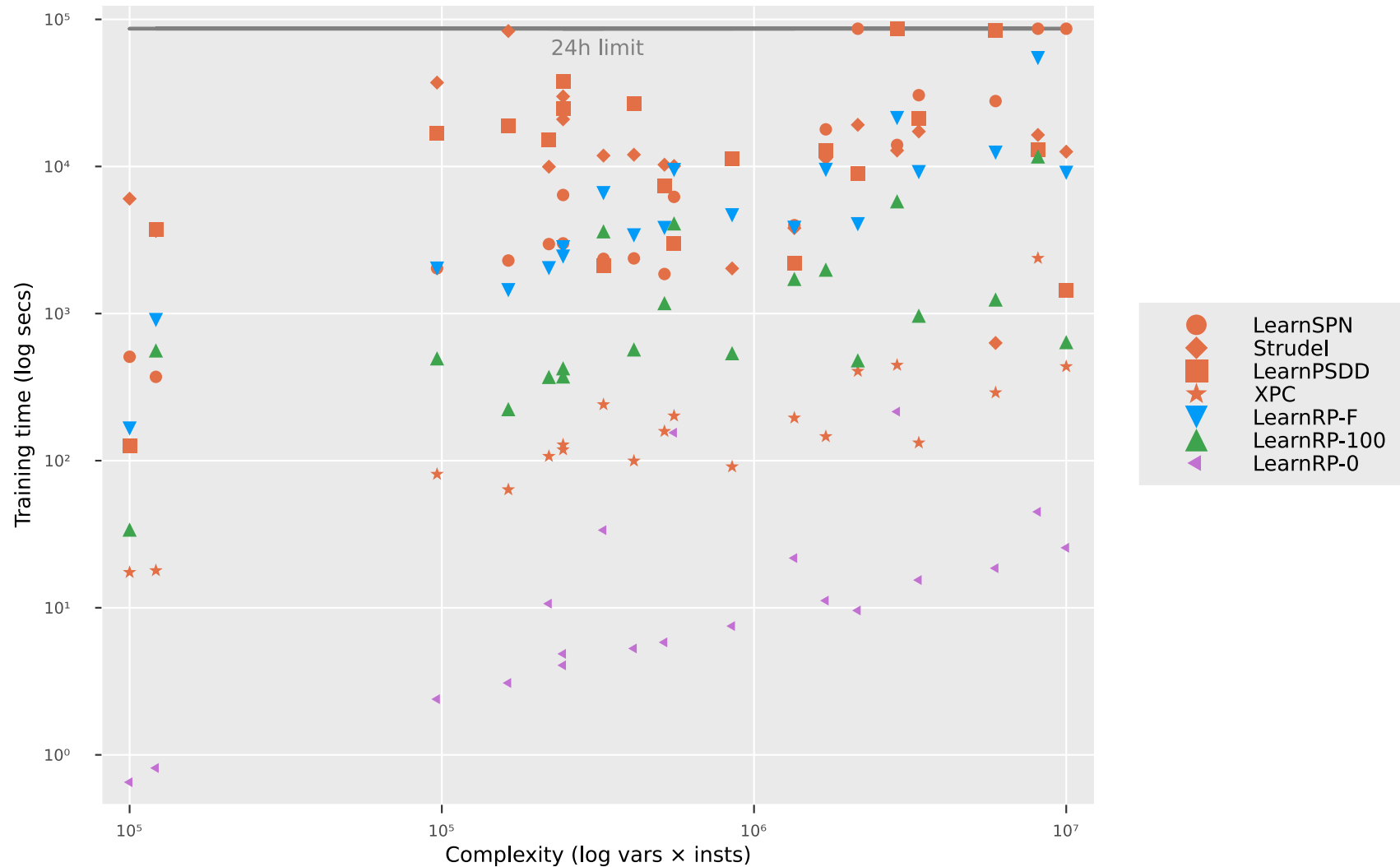
# Experiments

Dataset	LEARNSPN	STRUDEL	LEARNPSTD	XPC	PROMETHEUS	LEARNRP-F	LEARNRP-100
ACCIDENTS	-30.03	<u>-28.73</u>	-30.16	-31.02	<b>-27.91</b>	<u>-28.65</u>	-28.87
AD	-19.73	<u>-16.38</u>	-31.78	<b>-15.50</b>	-23.96	<u>-19.20</u>	-20.32
AUDIO	-40.50	-41.50	<u>-39.94</u>	-40.91	<b>-39.80</b>	<u>-40.18</u>	-40.23
BBC	<u>-250.68</u>	-254.41	-253.19	<b>-248.34</b>	<u>-248.50</u>	-254.97	-255.55
NETFLIX	<u>-57.02</u>	-58.69	<b>-55.71</b>	-57.58	<u>-56.47</u>	-57.07	-57.05
BOOK	-35.88	-34.99	-34.97	-34.75	<u>-34.40</u>	<u>-33.57</u>	<b>-33.52</b>
20-NEWSGRP	-155.92	-154.47	-155.97	<u>-153.75</u>	-154.17	<u>-152.78</u>	<b>-152.76</b>
REUTERS-52	<u>-85.06</u>	-86.22	-89.61	<u>-84.70</u>	<b>-84.59</b>	-85.73	-85.47
WEBKB	-158.20	-155.33	-161.09	<u>-153.67</u>	-155.21	<u>-154.43</u>	<b>-152.60</b>
DNA	<b>-82.52</b>	-86.22	-88.01	-86.61	-84.45	<u>-83.03</u>	<u>-83.85</u>
JESTER	-75.98	-55.03	<b>-51.29</b>	-53.43	<u>-52.80</u>	-52.92	<u>-52.89</u>
KDD	-2.18	<u>-2.13</u>	<b>-2.11</b>	-2.15	<u>-2.12</u>	<u>-2.13</u>	-2.14
KOSAREK	-10.98	-10.68	<b>-10.52</b>	-10.77	<u>-10.59</u>	<u>-10.65</u>	-10.67
MSNBC	<u>-6.11</u>	<b>-6.04</b>	<b>-6.04</b>	<u>-6.18</u>	<b>-6.04</b>	-6.31	-6.36
MSWEB	-10.25	<b>-9.71</b>	-9.89	-9.93	<u>-9.86</u>	<u>-9.85</u>	-9.97
NLTCS	-6.11	-6.06	<b>-5.99</b>	<u>-6.05</u>	<u>-6.01</u>	-6.35	-6.23
PLANTS	<u>-12.97</u>	<u>-12.98</u>	-13.02	-14.19	<b>-12.81</b>	-13.68	-14.00
PUMSB-STAR	<u>-24.78</u>	<u>-24.12</u>	-26.12	-26.06	<b>-22.75</b>	-25.88	-26.19
EACHMOVIE	-52.48	-53.67	-58.01	-54.82	<u>-51.49</u>	<u>-51.37</u>	<b>-51.06</b>
RETAIL	-11.04	<u>-10.81</u>	<b>-10.72</b>	-10.94	-10.87	<u>-10.85</u>	-10.86
<b>Avg. Rank</b>	4.80 ± 1.91	4.22 ± 1.81	4.05 ± 2.56	4.60 ± 1.93	<b>2.55 ± 1.43</b>	<u>3.62 ± 1.56</u>	4.15 ± 2.03
<b>Pos. (mean)</b>	7th	5th	<u>3rd</u>	6th	<b>1st</b>	<u>2nd</u>	4th

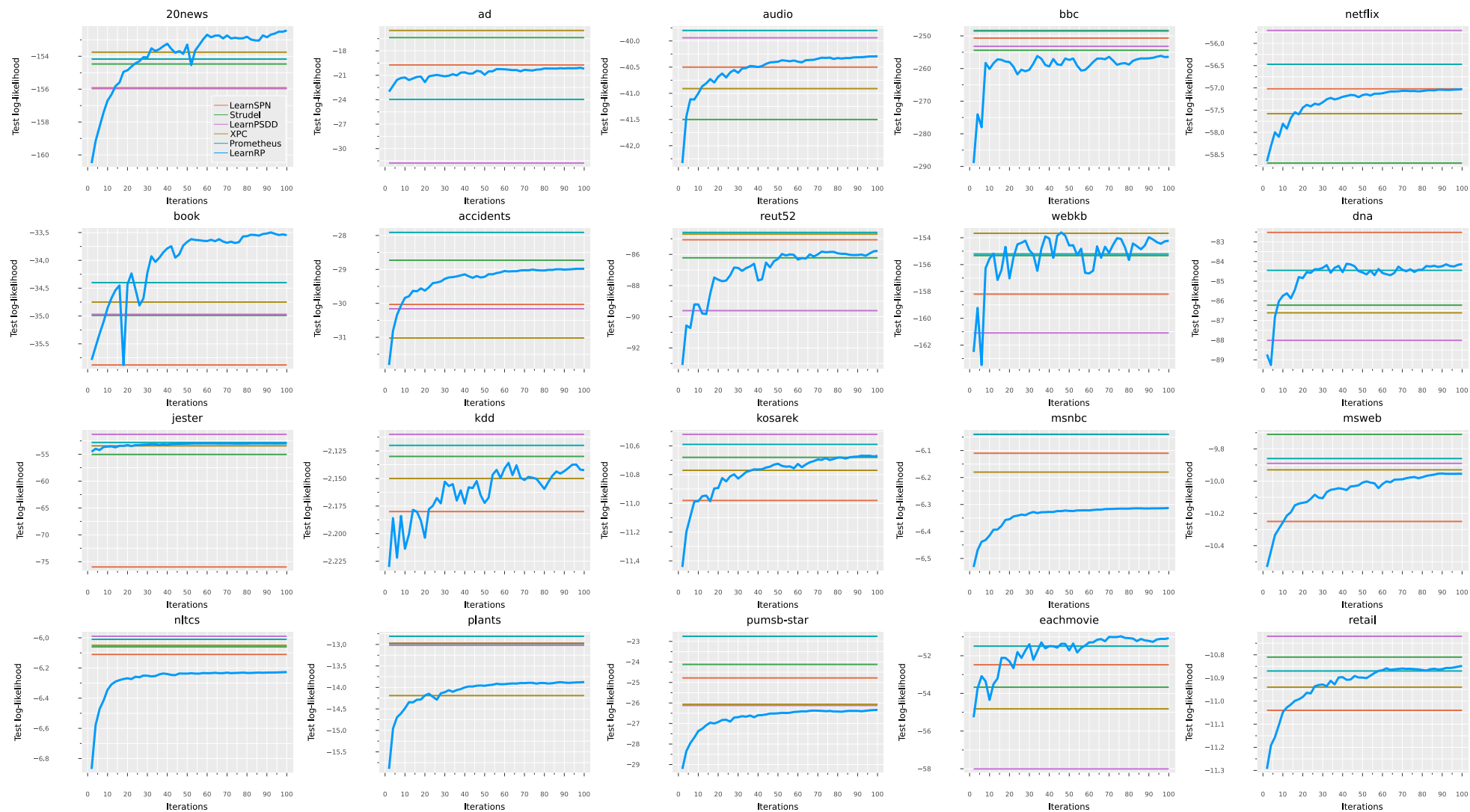
# Experiments



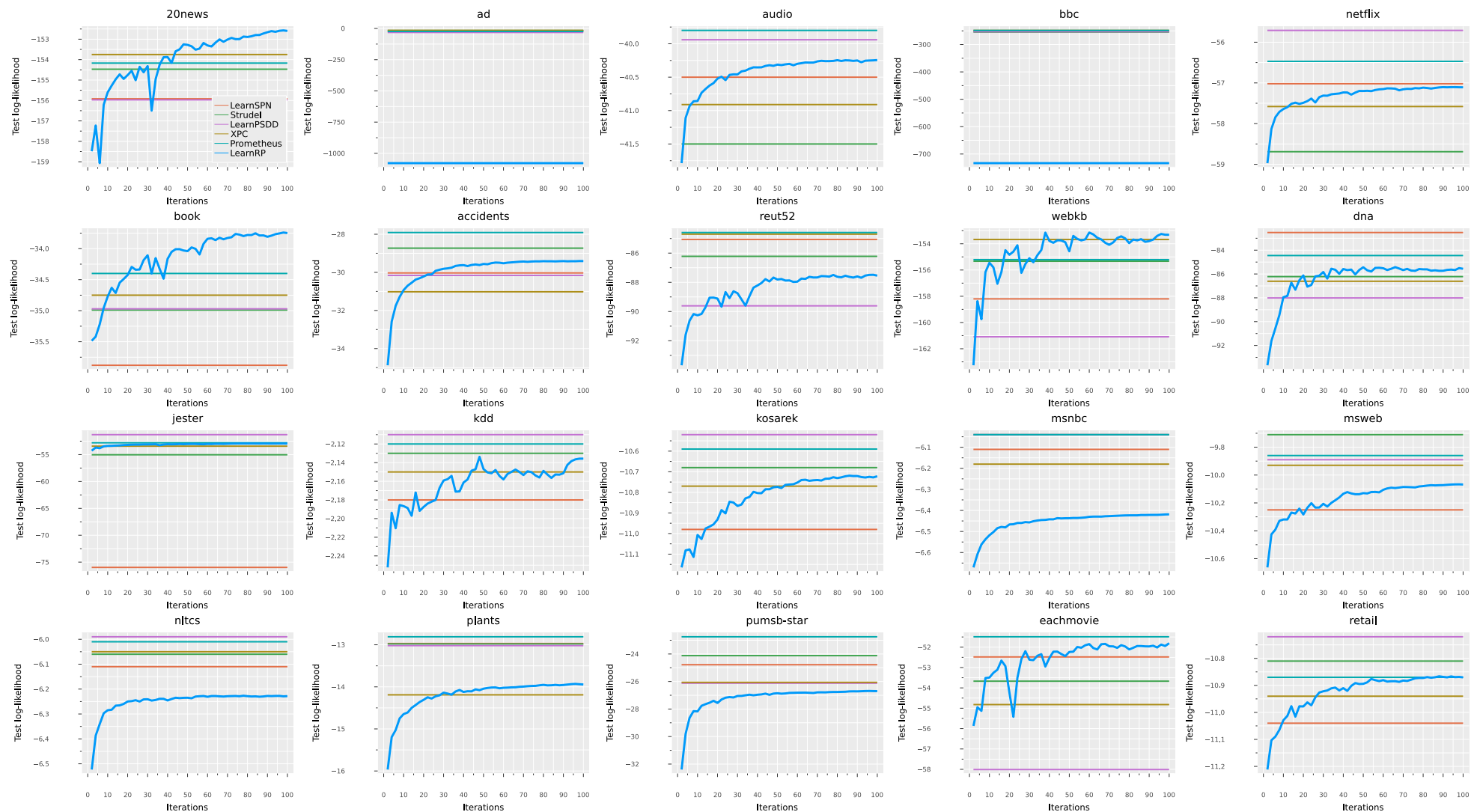
# Experiments



# LEARNRP – Learning Curves



# LEARNRP – Random Initializations





# Experiments

Dataset	Vars	SRBMs	oSLRAU	GBMMs	iGMMs	GMMs	PROMETHEUS	iSPTs	LEARNRP	Size
ABALONE	8	-2.28	<u>-0.94</u>	-1.17	—	<b>-0.59</b>	<u>-0.85</u>	—	-6.13	317
CA	22	-4.95	<u>21.19</u>	3.42	—	-1.08	<b>27.82</b>	—	-5.84	2765
QUAKE	4	-2.38	<u>-1.21</u>	-3.76	—	<b>-0.58</b>	<u>-1.50</u>	—	-3.76	79
SENSORLESS	48	-26.91	<u>60.72</u>	8.56	—	-1.39	<b>62.03</b>	—	-38.46	12589
BANKNOTE	4	-2.76	<u>-1.39</u>	-4.64	—	<b>-1.05</b>	<u>-1.96</u>	—	-6.06	79
FLOWSIZE	3	-0.79	<u>15.32</u>	5.72	—	-36.50	<b>18.03</b>	—	2.20	49
KINEMATICS	8	<b>-5.55</b>	-11.13	-11.20	—	<u>-6.11</u>	-11.12	—	<u>-11.02</u>	319
IRIS	4	—	—	—	-3.94	<b>0.20</b>	<u>-1.06</u>	-3.74	<u>-3.47</u>	79
OLDFAITH	2	—	—	—	<u>-1.73</u>	-2.09	<b>-1.48</b>	<u>-1.70</u>	-4.33	19
CHEMDIABET	3	—	—	—	-3.02	<b>-0.58</b>	<u>-2.59</u>	<u>-2.88</u>	-18.68	48

In conclusion

# Contributions

## Literature review

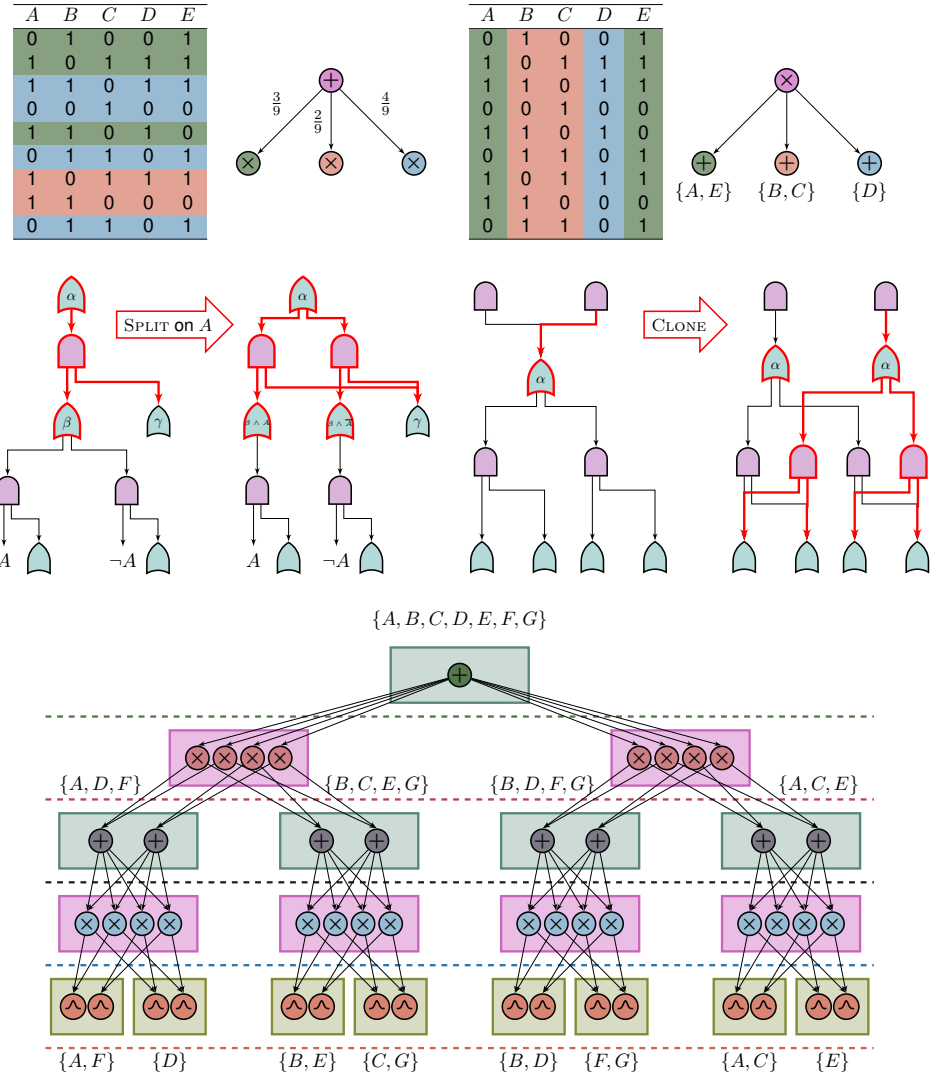
- Systematic review of literature;
- Taxonomy of popular algorithms;
- Complexity analysis;
- Pros and cons.

## SAMPLEPSDD

- Consistent with a relaxation of a formula;
- Relaxation as a function of vtree and sampling;
- Compromise between tractability and consistency;
- Ensembles mitigate relaxation.

## LEARNRP

- Simple strategy;
- Inspiration from known DET literature;
- Orders of magnitude faster;
- Competitive performance.



# Contributions

## Literature review

- Systematic review of literature;
- Taxonomy of popular algorithms;
- Complexity analysis;
- Pros and cons.

## SAMPLEPSDD

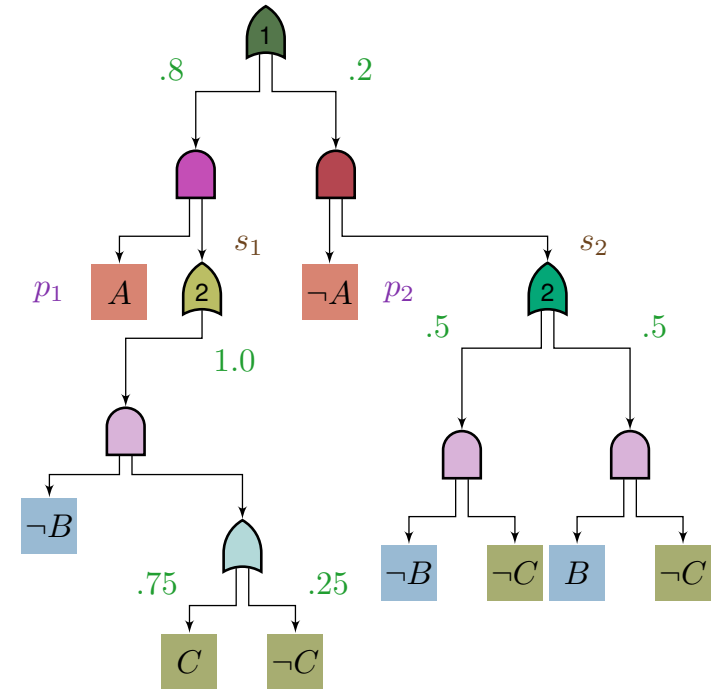
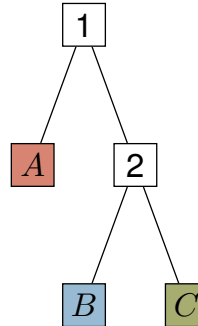
- Consistent with a relaxation of a formula;
- Relaxation as a function of vtree and sampling;
- Compromise between tractability and consistency;
- Ensembles mitigate relaxation.

## LEARNRP

- Simple strategy;
- Inspiration from known DET literature;
- Orders of magnitude faster;
- Competitive performance.

$A$	$B$	$C$	$p(\mathbf{x})$
0	0	0	0.1
0	1	0	0.1
1	0	0	0.2
1	0	1	0.6

$$\phi(A, B, C) = (A \rightarrow \neg B) \wedge (C \rightarrow A)$$



# Contributions

## Literature review

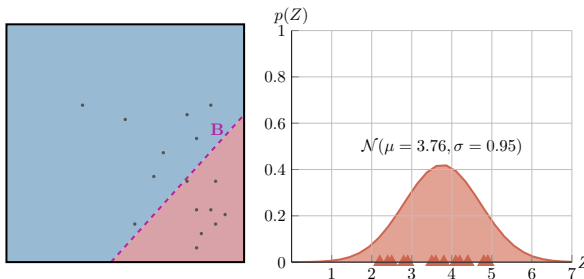
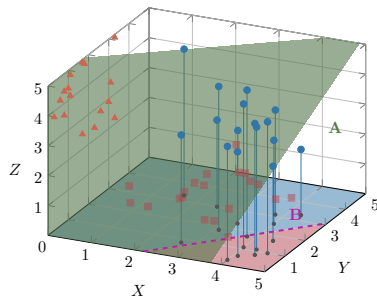
- Systematic review of literature;
- Taxonomy of popular algorithms;
- Complexity analysis;
- Pros and cons.

## SAMPLEPSDD

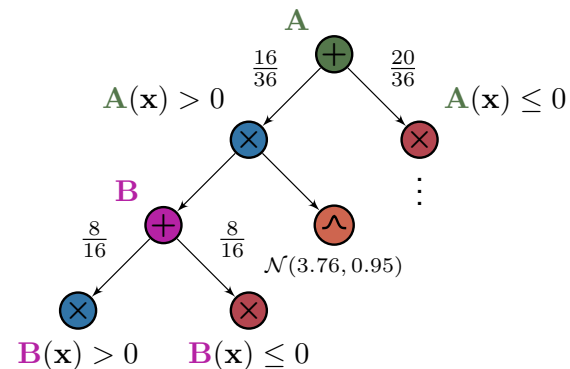
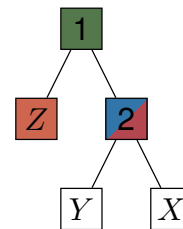
- Consistent with a relaxation of a formula;
- Relaxation as a function of vtree and sampling;
- Compromise between tractability and consistency;
- Ensembles mitigate relaxation.

## LEARNRP

- Simple strategy;
- Inspiration from known DET literature;
- Orders of magnitude faster;
- Competitive performance.



$$\mathbf{B}(x, y) = \begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} 1.10 \\ -1.00 \end{bmatrix} - 2.43$$



# Supplemental Material

# LEARNRP – Binary Benchmark

Dataset	LEARNSPN	STRUDEL	LEARNSDD	XPC	PROMETHEUS	LEARNRP-F	LEARNRP-100	LEARNRP-30	LEARNRP-20	LEARNRP-10
ACCIDENTS	-30.03	<u> -28.73 </u>	-30.16	-31.02	<b>-27.91</b>	<u> -28.65 </u>	-28.87	-29.38	-29.58	-29.99
AD	-19.73	<u> -16.38 </u>	-31.78	<b>-15.50</b>	-23.96	<u> -19.20 </u>	-20.32	-21.42	-21.44	-21.94
AUDIO	-40.50	-41.50	<u> -39.94 </u>	-40.91	<b>-39.80</b>	<u> -40.18 </u>	-40.23	-40.46	-40.63	-40.94
BBC	<u> -250.68 </u>	-254.41	-253.19	<b>-248.34</b>	<u> -248.50 </u>	-254.97	-255.55	-262.35	-257.67	-262.39
NETFLIX	<u> -57.02 </u>	-58.69	<b>-55.71</b>	-57.58	<u> -56.47 </u>	-57.07	-57.05	-57.29	-57.48	-57.66
BOOK	-35.88	-34.99	-34.97	-34.75	-34.40	<u> -33.57 </u>	<b>-33.52</b>	-34.34	<u> -34.24 </u>	-34.73
20-NEWSGRP	-155.92	-154.47	-155.97	<u> -153.75 </u>	-154.17	<u> -152.78 </u>	<b>-152.76</b>	-154.32	-155.03	-156.26
REUTERS-52	<u> -85.06 </u>	-86.22	-89.61	<u> -84.70 </u>	<b>-84.59</b>	-85.73	-85.47	-87.41	-87.05	-89.26
WEBKB	-158.20	-155.33	-161.09	<u> -153.67 </u>	-155.21	-154.43	<b>-152.60</b>	-154.83	<u> -154.33 </u>	-158.01
DNA	<b>-82.52</b>	-86.22	-88.01	-86.61	-84.45	<u> -83.03 </u>	<u> -83.85 </u>	-84.77	-84.98	-85.40
JESTER	-75.98	-55.03	<b>-51.29</b>	-53.43	<u> -52.80 </u>	-52.92	<u> -52.89 </u>	-53.23	-53.22	-53.54
KDD	-2.18	<u> -2.13 </u>	<b>-2.11</b>	-2.15	<u> -2.12 </u>	<u> -2.13 </u>	-2.14	-2.17	-2.16	-2.20
KOSAREK	-10.98	-10.68	<b>-10.52</b>	-10.77	<u> -10.59 </u>	<u> -10.65 </u>	-10.67	-10.79	-10.86	-11.00
MSNBC	<u> -6.11 </u>	<b>-6.04</b>	<b>-6.04</b>	<u> -6.18 </u>	<b>-6.04</b>	-6.31	-6.36	-6.40	-6.41	-6.44
MSWEB	-10.25	<b>-9.71</b>	-9.89	-9.93	<u> -9.86 </u>	<u> -9.85 </u>	-9.97	-10.06	-10.21	-10.27
NLTCS	-6.11	-6.06	<b>-5.99</b>	<u> -6.05 </u>	-6.01	-6.35	-6.23	-6.25	-6.27	-6.32
PLANTS	<u> -12.97 </u>	<u> -12.98 </u>	-13.02	-14.19	<b>-12.81</b>	-13.68	-14.00	-14.26	-14.40	-14.70
PUMSB-STAR	<u> -24.78 </u>	<u> -24.12 </u>	-26.12	-26.06	<b>-22.75</b>	-25.88	-26.19	-26.36	-26.54	-27.17
EACHMOVIE	-52.48	-53.67	-58.01	-54.82	<u> -51.49 </u>	<u> -51.37 </u>	<b>-51.06</b>	-51.55	-52.86	-52.21
RETAIL	-11.04	<u> -10.81 </u>	<b>-10.72</b>	-10.94	-10.87	<u> -10.85 </u>	-10.86	-10.93	-10.97	-11.04
<b>Avg. Rank</b>	6.08 ± 3.03 4.80 ± 1.91	5.28 ± 2.97 4.22 ± 1.81	5.20 ± 3.86 4.05 ± 2.56	5.55 ± 2.76 4.60 ± 1.93	<b>2.90 ± 2.07</b> <b>2.55 ± 1.43</b>	3.83 ± 1.98 3.62 ± 1.56	4.15 ± 2.03 4.15 ± 2.03	6.35 ± 1.50	6.95 ± 1.70	8.72 ± 1.50
<b>Pos. (mean)</b>	7th 7th	5th 5th	4th <u>3rd</u>	6th 6th	<b>1st</b> <b>1st</b>	<u>2nd</u> <u>2nd</u>	<u>3rd</u> 4th	8th	9th	10th