Scalable Learning of Probabilistic Circuits

Motivation

Given a selection of sushi...

...and people's preferences...

Alice: 

Bob: 

Carol: 

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

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

$p(2^{\text{nd}} =$  $| 1^{\text{st}} =$  $)$

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

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

...and extract meaningful queries from it?

Kamishima [2003]

Motivation

Given a selection of sushi...

...and people's preferences...

Alice: 

Bob: 

Carol: 

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

$p(1^{st} =$  $, 3^{rd} =$  $)$ **Marginals**

$p(2^{nd} =$  $| 1^{st} =$  $)$ **Conditionals**

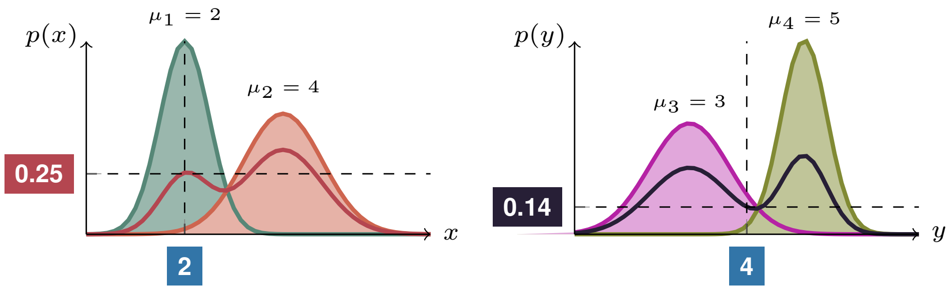
$\arg\max p(1^{st} = ?, 2^{nd} = ?, 3^{rd} = ?, 4^{th} =$  $, 5^{th} =$  $)$ **MPE**

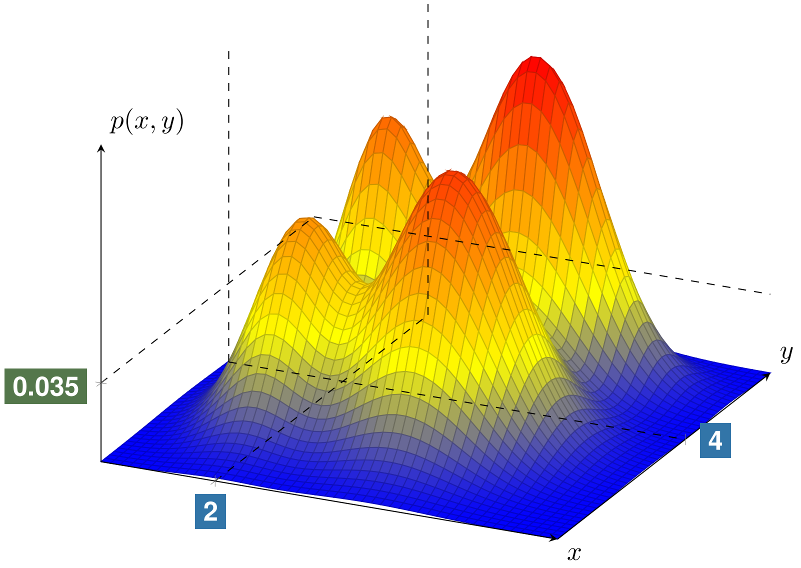
$p((3^{rd} =$  $\rightarrow 1^{st} =$  $) \vee 2^{nd} =$  $)$ **Logical events**

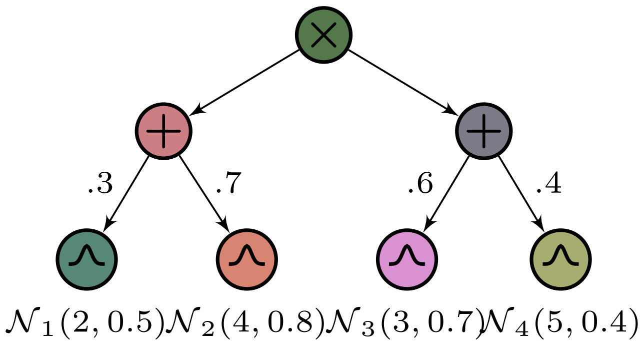
...and extract meaningful queries from it?

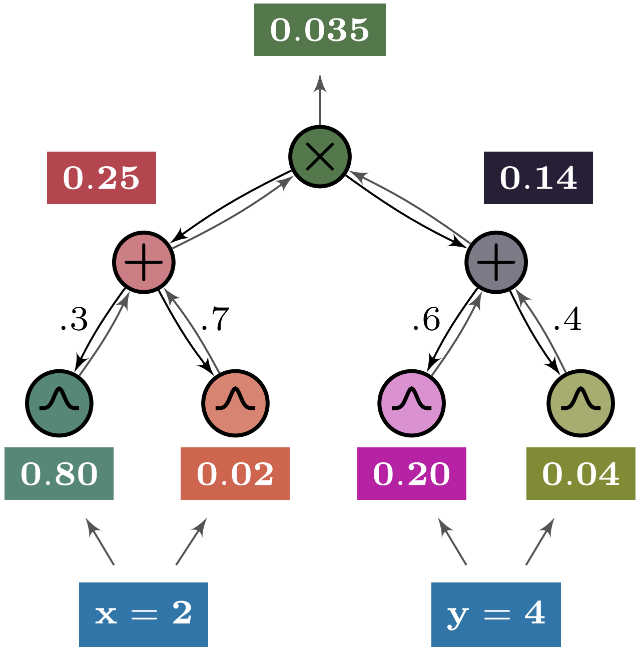
Kamishima [2003]

Probabilistic Circuits

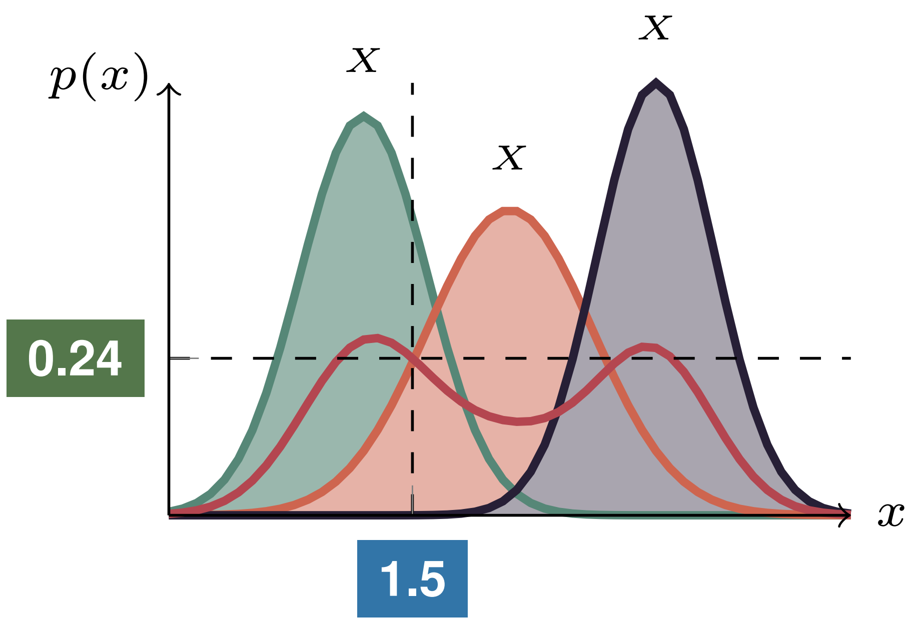


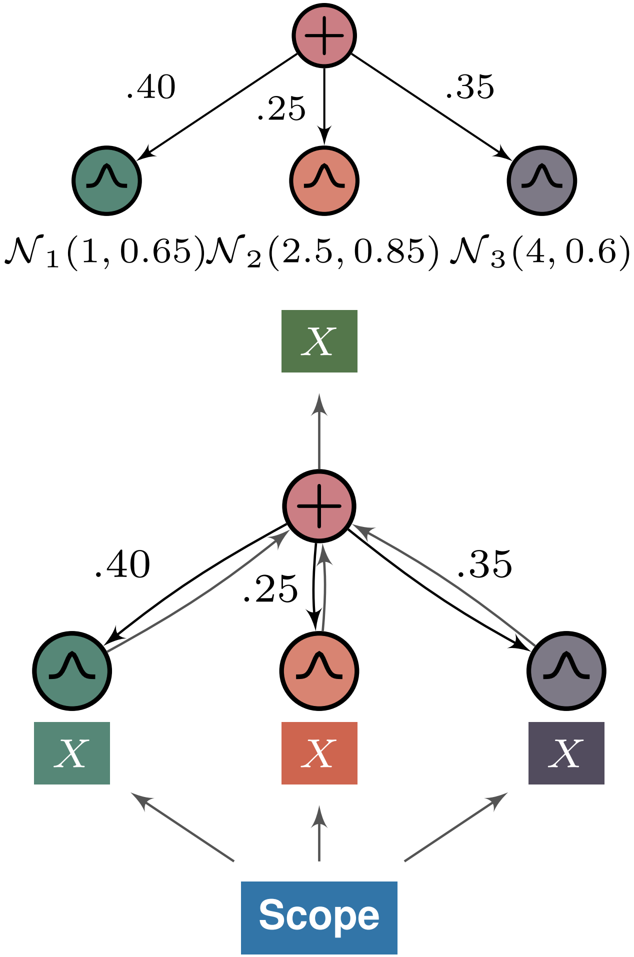






Probabilistic Circuits – Smoothness

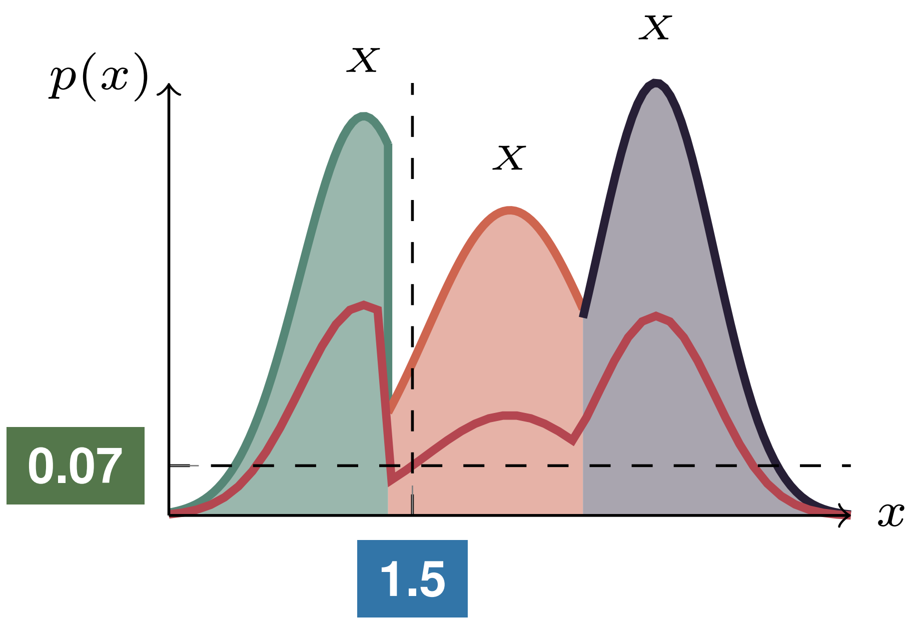


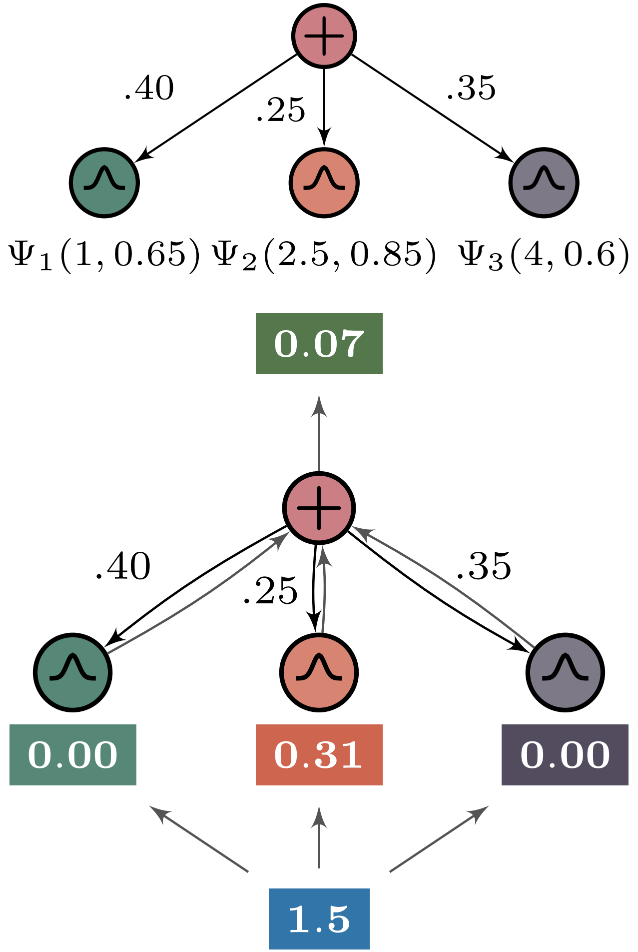


Definition 1 (Smoothness)**.**
Every sum node child mentions the same variables.

Darwiche [2001a]

Probabilistic Circuits – Determinism

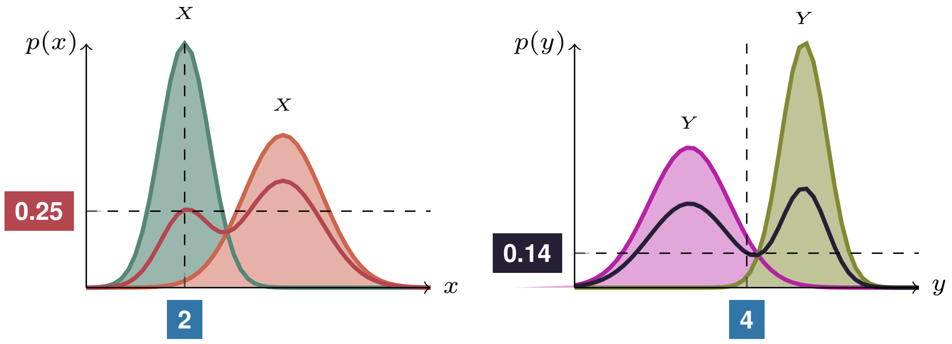




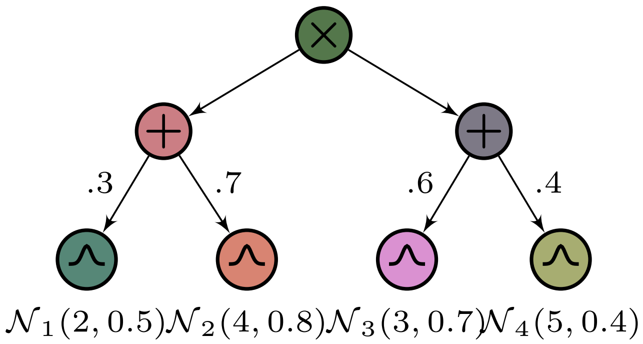
Definition 2 (Determinism)**.**
<u>At most one</u> sum node child has a positive value.

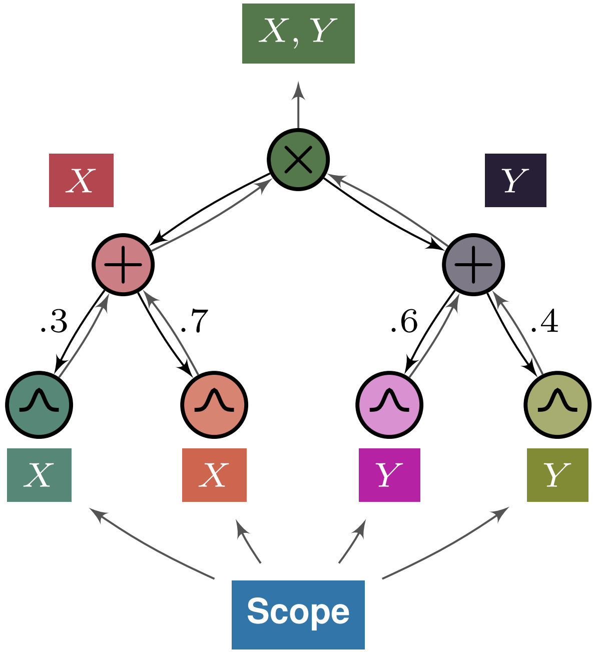
Darwiche [2001a]

Probabilistic Circuits – Decomposability



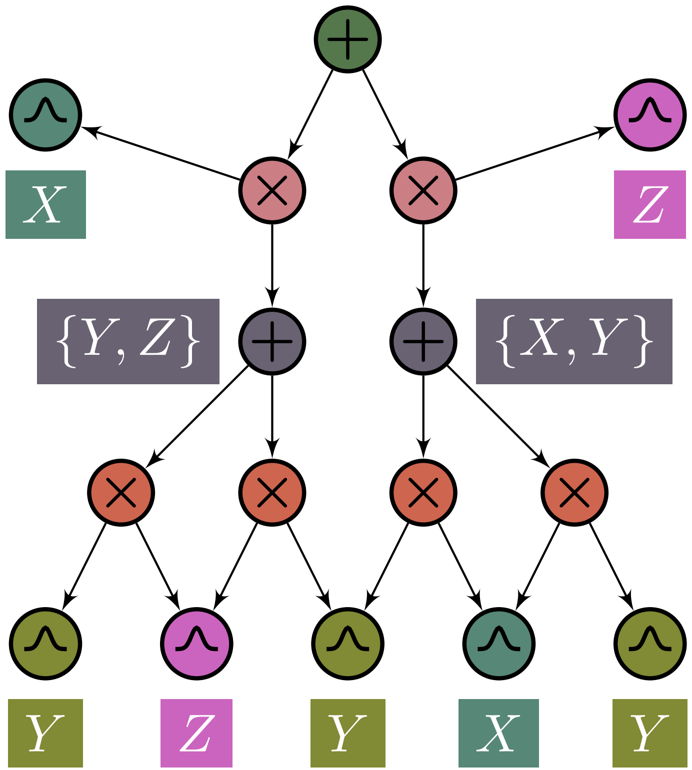
Definition 3 (Decomposability)**.**
Every product node child mentions *<u>different</u>* *variables.*

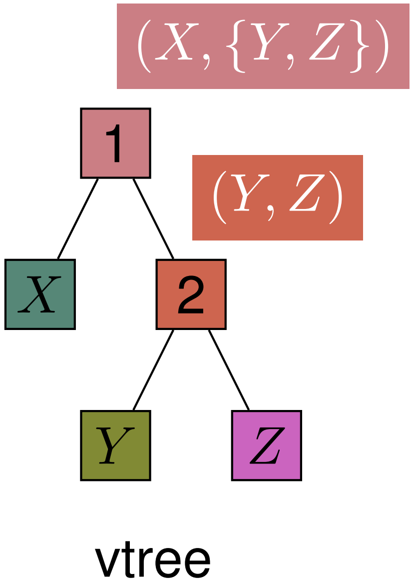


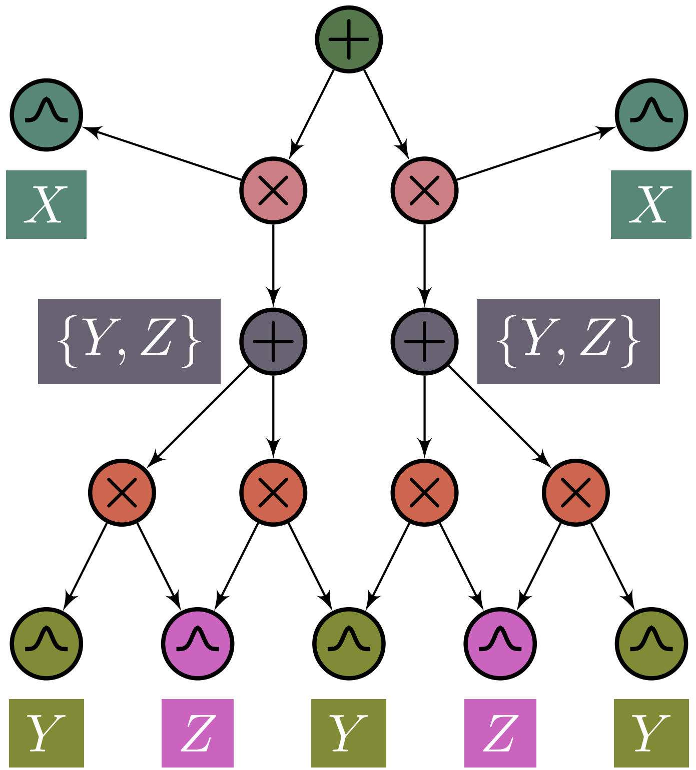


Darwiche [1999, 2001b]

Probabilistic Circuits – Structured Decomposability







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

Pipatsrisawat and Darwiche [2008]

Probabilistic Circuits – Tractability

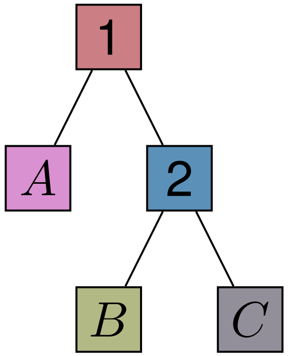
Query	+Sm?	+Dec?	+Det?	+Str Dec?
Evidence	✓	✓	✓	✓
Marginals	✗	✓	✓	✓
Conditionals	✗	✓	✓	✓
MPE	✗	✗	✓	✓
Shannon Entropy*	✗	✗	✓	✓
Rényi Entropy*	✗	✗	✓	✓
Cross Entropy*	✗	✗	✗	✓
Kullback-Leibler Div*	✗	✗	✗	✓
Rényi's Alpha Div*	✗	✗	✗	✓
Cauchy-Schwarz Div*	✗	✗	✗	✓
Logical Events	✗	✗	✗	✓
Mutual Information*	✗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

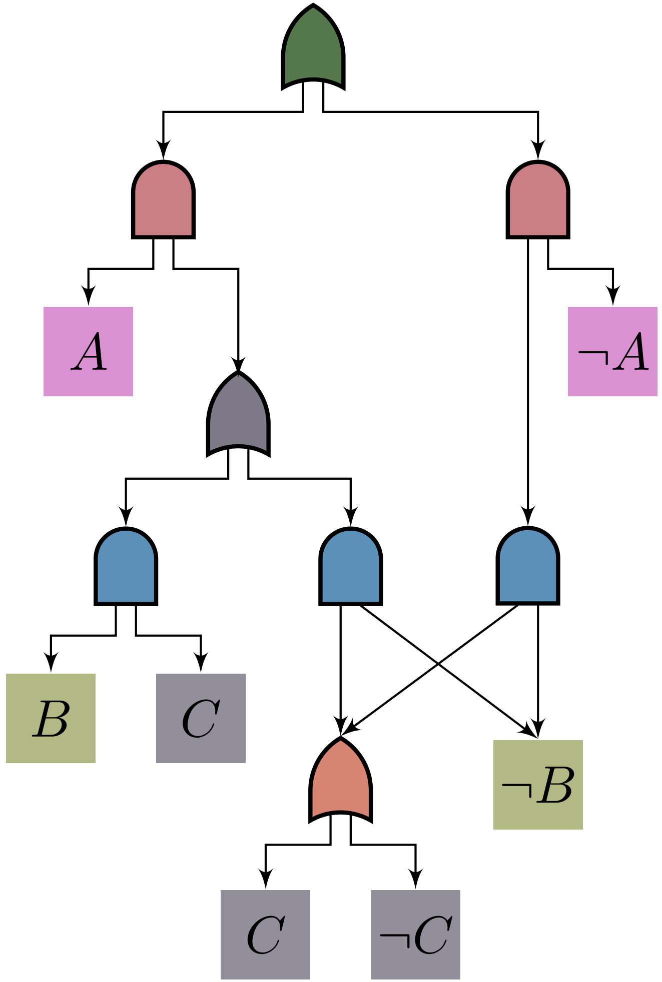
Vergari et al. [2021], Poon and Domingos [2011], Peharz et al. [2016]

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

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



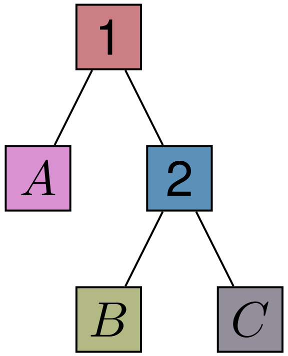


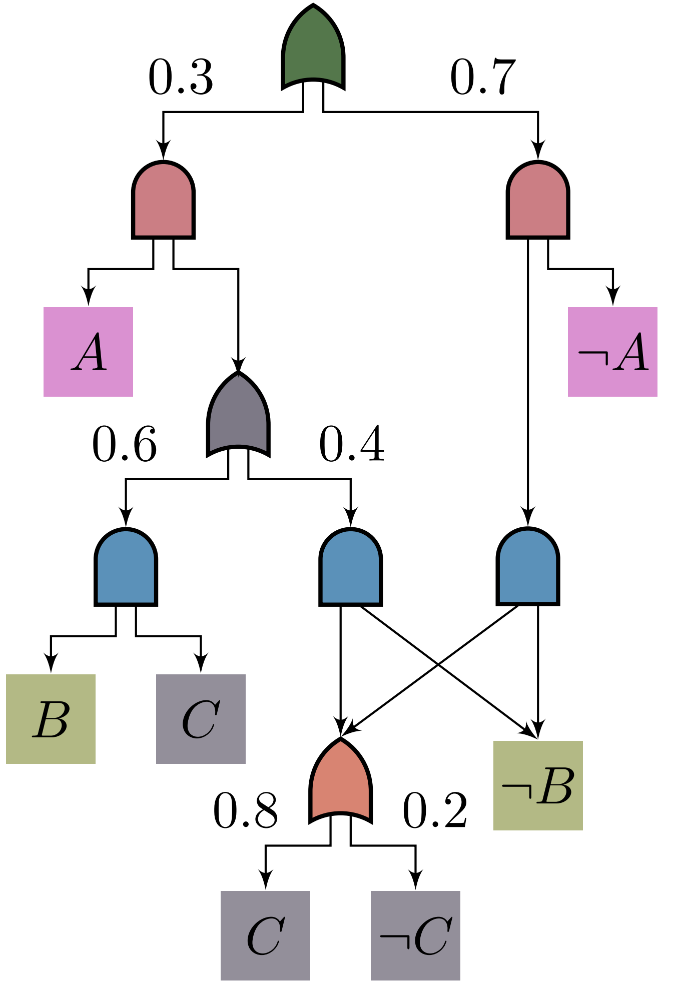
Darwiche [2011]

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

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





Kisa et al. [2014]

Learning Probabilistic Circuits

Divide-and-Conquer Approaches (DIV)

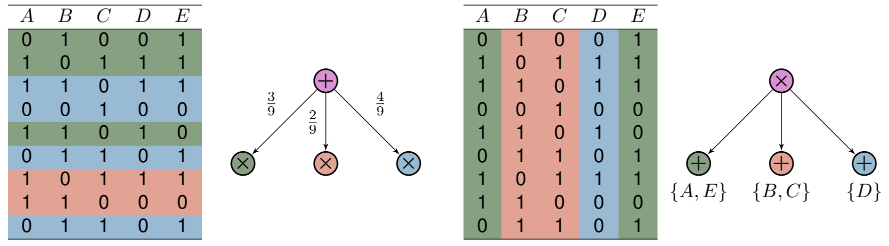
- Usually recursive;
- Splits data by similarity and stat dep;
- Stat dep usually costly;
- Usually tree-shaped.

Incremental Approaches (INCR)

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

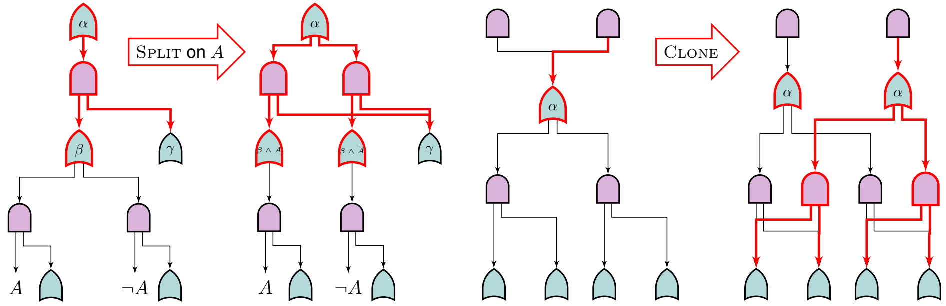
Random Approaches (RAND)

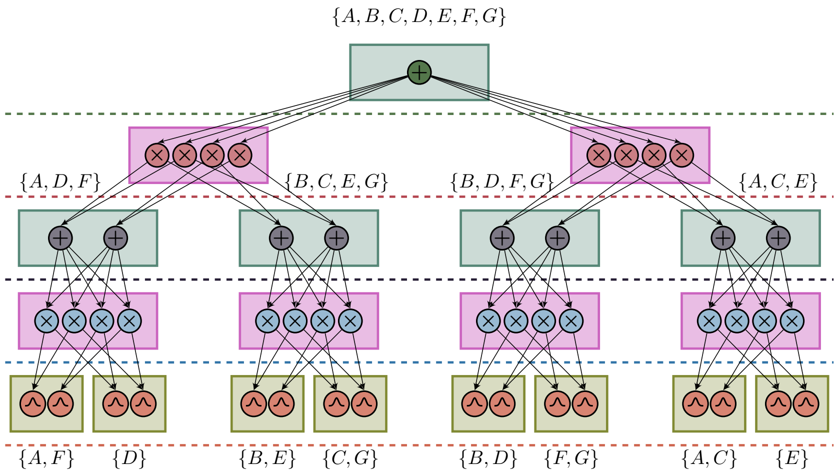
- Fast;
- Randomly generates circuits;
- Data blind and data guided approaches exist;
- Usually relies on many hyperparams;
- Worse performance.



A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1

A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1





Learning Probabilistic Circuits

Divide-and-Conquer Approaches (DIV)

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

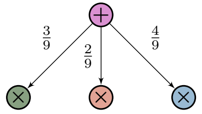
Incremental Approaches (INCR)

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

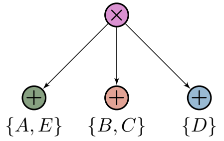
Random Approaches (RAND)

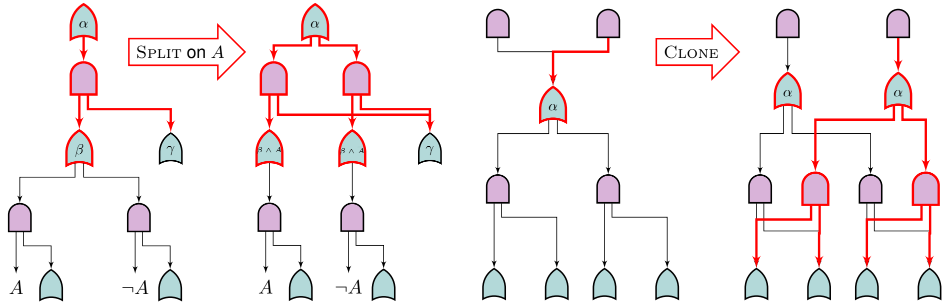
- Fast;
- Randomly generates circuits;
- Data blind and data guided approaches exist;
- Usually relies on many hyperparams;
- Worse performance.

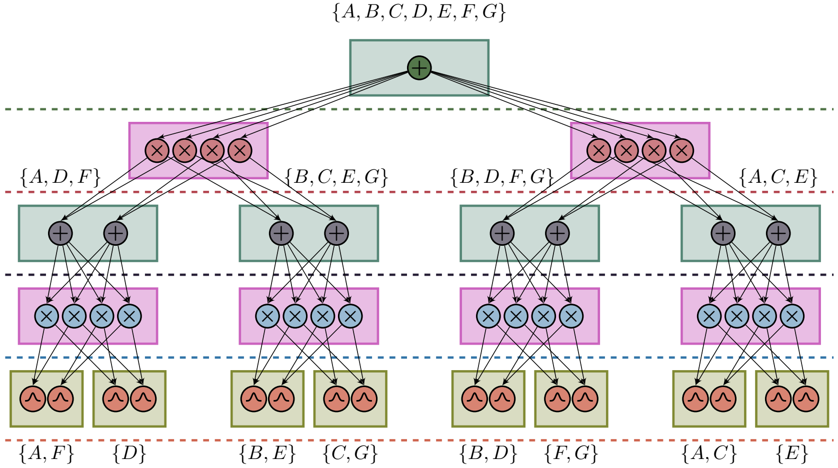
A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1



A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1







Learning Probabilistic Circuits

Divide-and-Conquer Approaches (DIV)

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

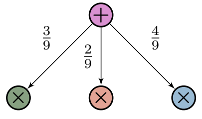
Incremental Approaches (INCR)

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

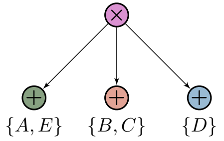
Random Approaches (RAND)

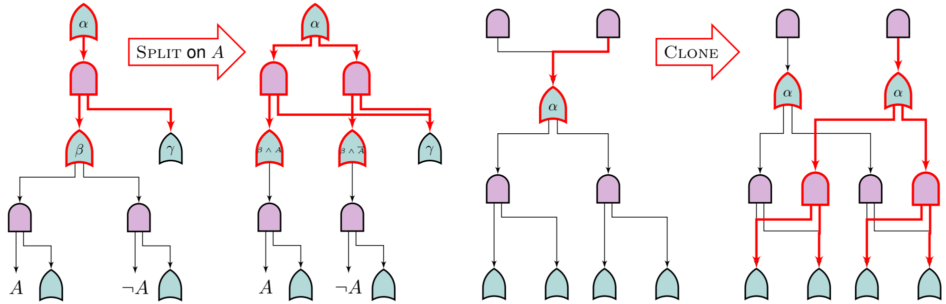
- Fast;
- Randomly generates circuits;
- Data blind and data guided approaches exist;
- Usually relies on many hyperparams;
- Worse performance.

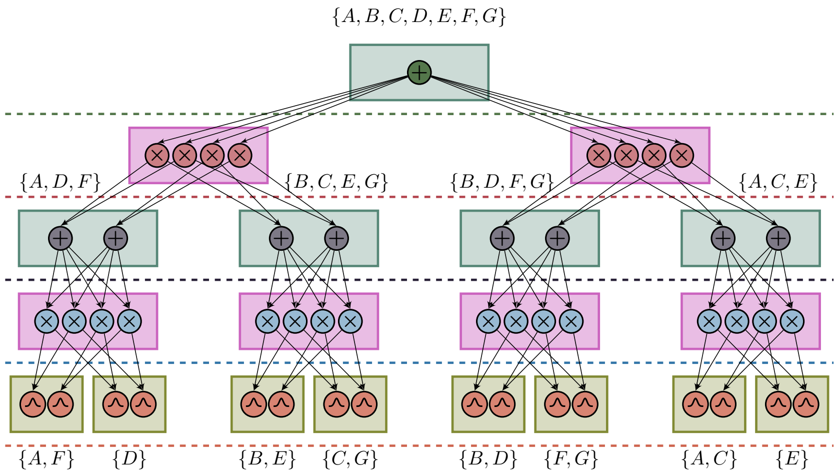
A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1



A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1







Learning Probabilistic Circuits – Where are we right now?

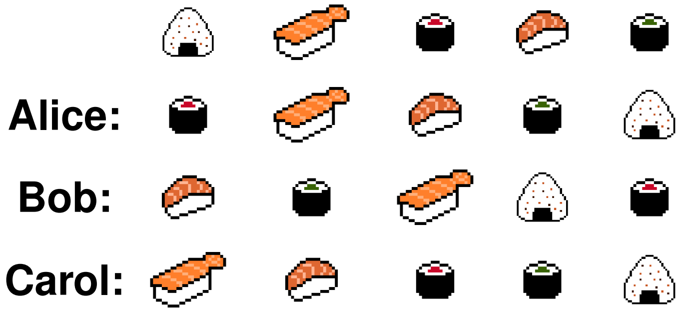
Name	Class	Time Complexity	# hyperparams	Accepts logic?	Sm?	Dec?	Det?	Str Dec?	$\{0,1\}$?	$\mathbb{N}$?	$\mathbb{R}$?	Reference
LearnSPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(nm^3) & \text{, if product} \end{cases}$	≥ 2	✗	✓	✓	✗	✗	✓	✓	✓	Gens and Domingos [2013]
ID-SPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(nm^3) & \text{, if product} \\ \mathcal{O}(ic(rn+m)) & \text{, if input} \end{cases}$	$\geq 2+3$	✗	✓	✓	✗	✗	✓	✓	✗	Rooshenas and Lowd [2014]
Prometheus	DIV	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(m(\log m)^2) & \text{, if product} \end{cases}$	≥ 1	✗	✓	✓	✗	✗	✓	✓	✓	Jaini et al. [2018a]
LearnPSDD	INCR	$\begin{cases} \mathcal{O}(m^2) & \text{, top-down vtree} \\ \mathcal{O}(m^4) & \text{, bottom-up vtree} \\ \mathcal{O}(i\lvert\mathcal{C}\rvert^2) & \text{, circuit structure} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Liang et al. [2017]
Strudel	INCR	$\begin{cases} \mathcal{O}(m^2 n) & \text{, CLT + vtree} \\ \mathcal{O}(i(\lvert\mathcal{C}\rvert n+m^2)) & \text{, circuit structure} \end{cases}$	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]
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	✓	✓	✓	✓	✓	✓	✗	✗	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	✗	✓	✓	✗	✓	✓	✓	✓	To appear

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	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(nm^3) & \text{, if product} \end{cases}$	≥ 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	$\begin{cases} \mathcal{O}(nkmc) & \text{, if sum} \\ \mathcal{O}(m(\log m)^2) & \text{, if product} \end{cases}$	≥ 1	✗	✓	✓	✗	✗	✓	✓	✓	Jaini et al. [2018a]
LEARNPSDD	INCR	$\begin{cases} \mathcal{O}(m^2) & \text{, top-down vtree} \\ \mathcal{O}(m^4) & \text{, bottom-up vtree} \\ \mathcal{O}(i\lvert\mathcal{C}\rvert^2) & \text{, circuit structure} \end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Liang et al. [2017]
STRUDEL	INCR	$\begin{cases} \mathcal{O}(m^2 n) & \text{, CLT + vtree} \\ \mathcal{O}(i(\lvert\mathcal{C}\rvert n + m^2)) & \text{, circuit structure} \end{cases}$	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]
$\Rightarrow$ 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	✓	✓	✓	✓	✓	✓	✗	✗	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	✗	✓	✓	✗	✓	✓	✓	✓	To appear

A Logical Perspective

Motivation



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!

Example:

$n = 3,\ k = 3$

X_{11}	X_{12}	X_{13}	X_{21}	$\cdots$	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: $3! = 6$

Choi et al. [2015]

Motivation

Existing approaches:

LearnPSDD (Liang et al. [2017]):

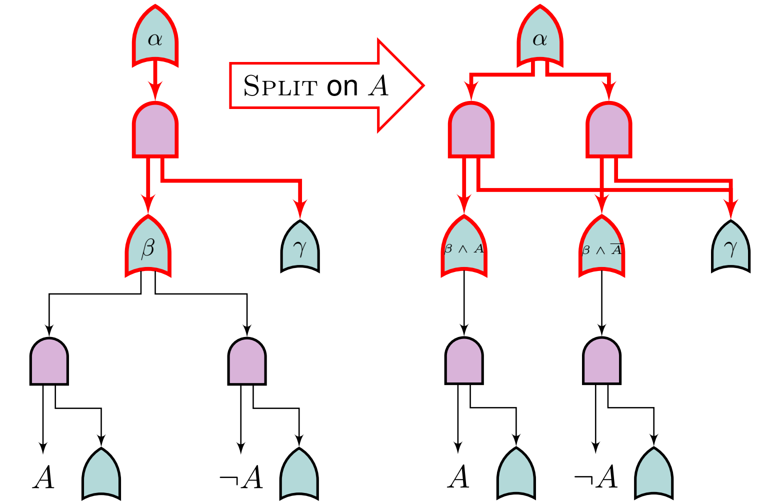
- ☒ Requires initial logic circuit encoding the support...
- ☒ Scales poorly to complex formulae and/or high dimension...
- ☒ Costly whole circuit evaluation at every iteration...
- ☑ Very good performance!

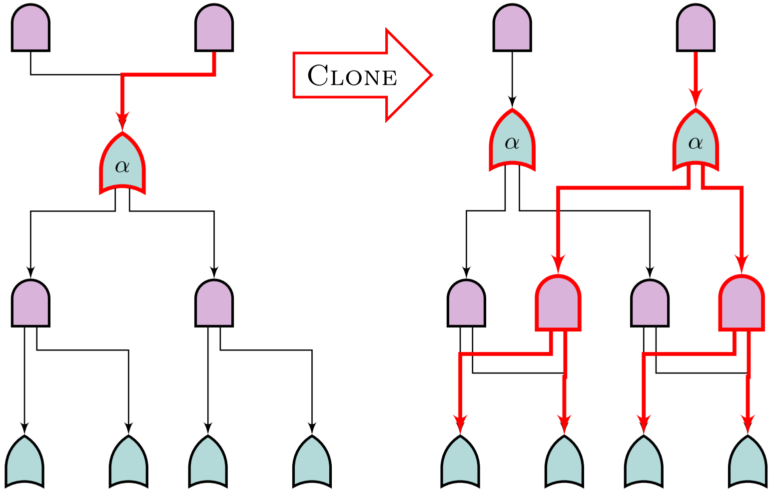
Strudel (Dang et al. [2020]):

- ☑ 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...



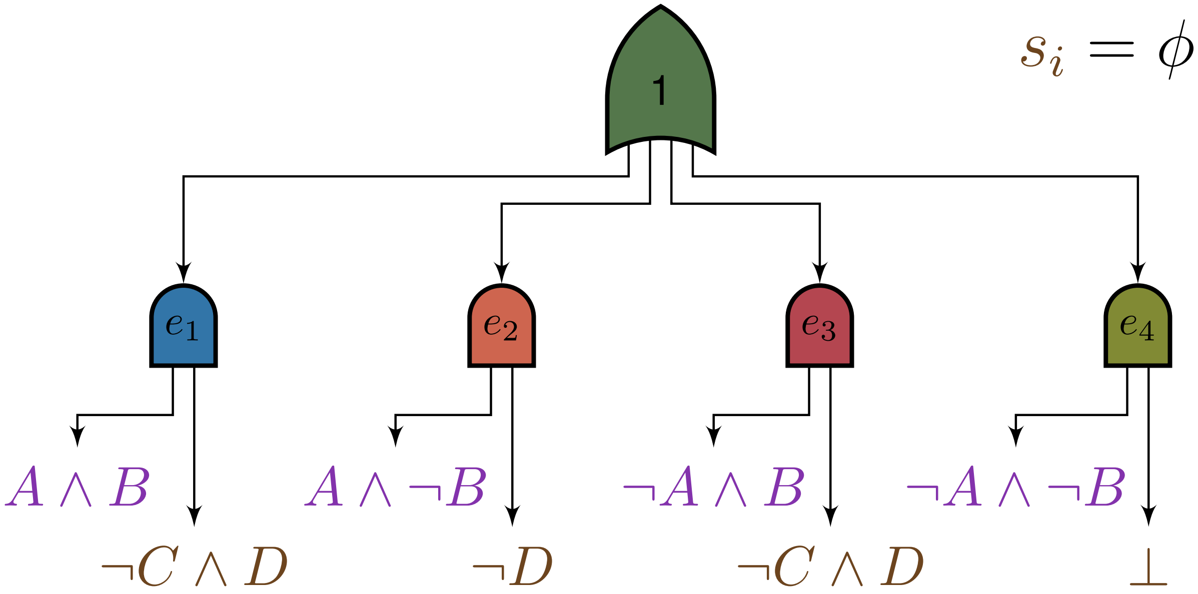


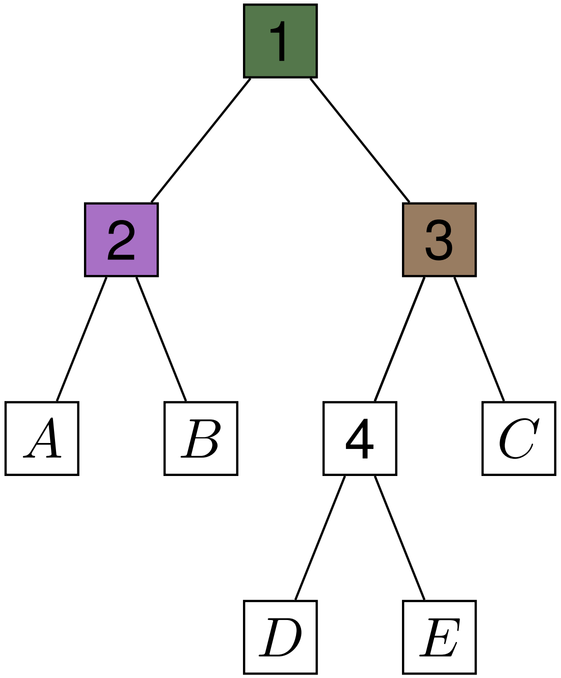
SamplePSDD

Common assumption: p_i are conjunctions of literals.

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

$$s_i = \phi|_{p_i}$$



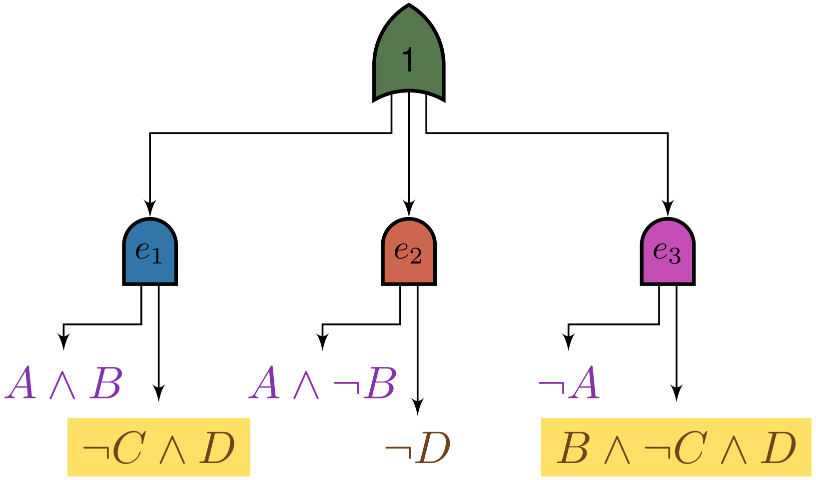


Problem: size of circuit is exponential in the size of p_i's scope.

SamplePSDD

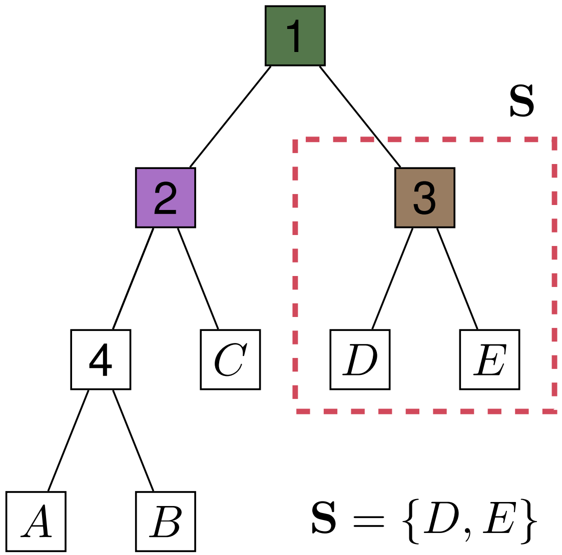
Solution: randomly sample a bounded number (k) of p_i

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



$$s_i = \phi|_{p_i}$$

$\mathrm{Sc}(s_3) \not\subseteq \mathbf{S}$



$\mathbf{S} = \{D, E\}$

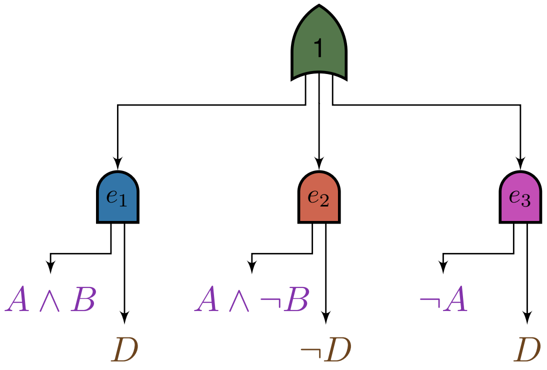
But: this violates structured decomposability:

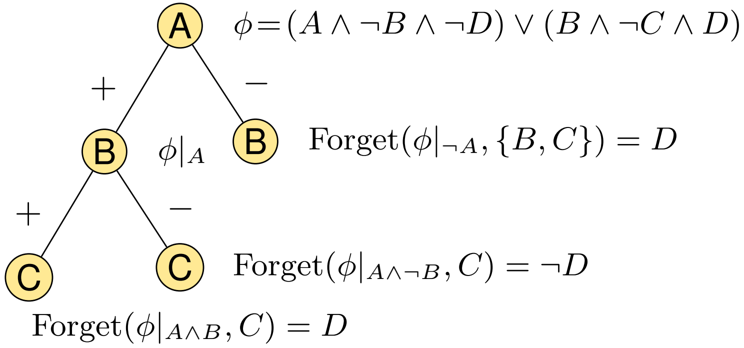
$\neg C \wedge D$ contains C, and $C \notin \mathbf{S}$

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

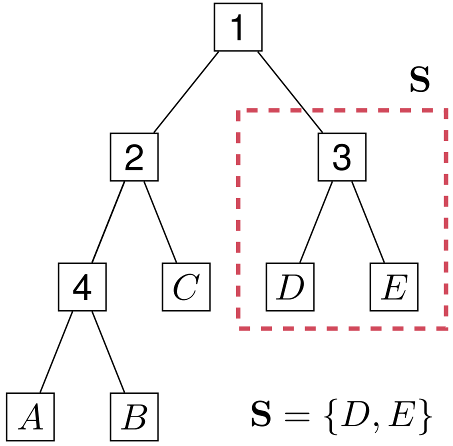
SamplePSDD

New solution: relax logical constraints ϕ



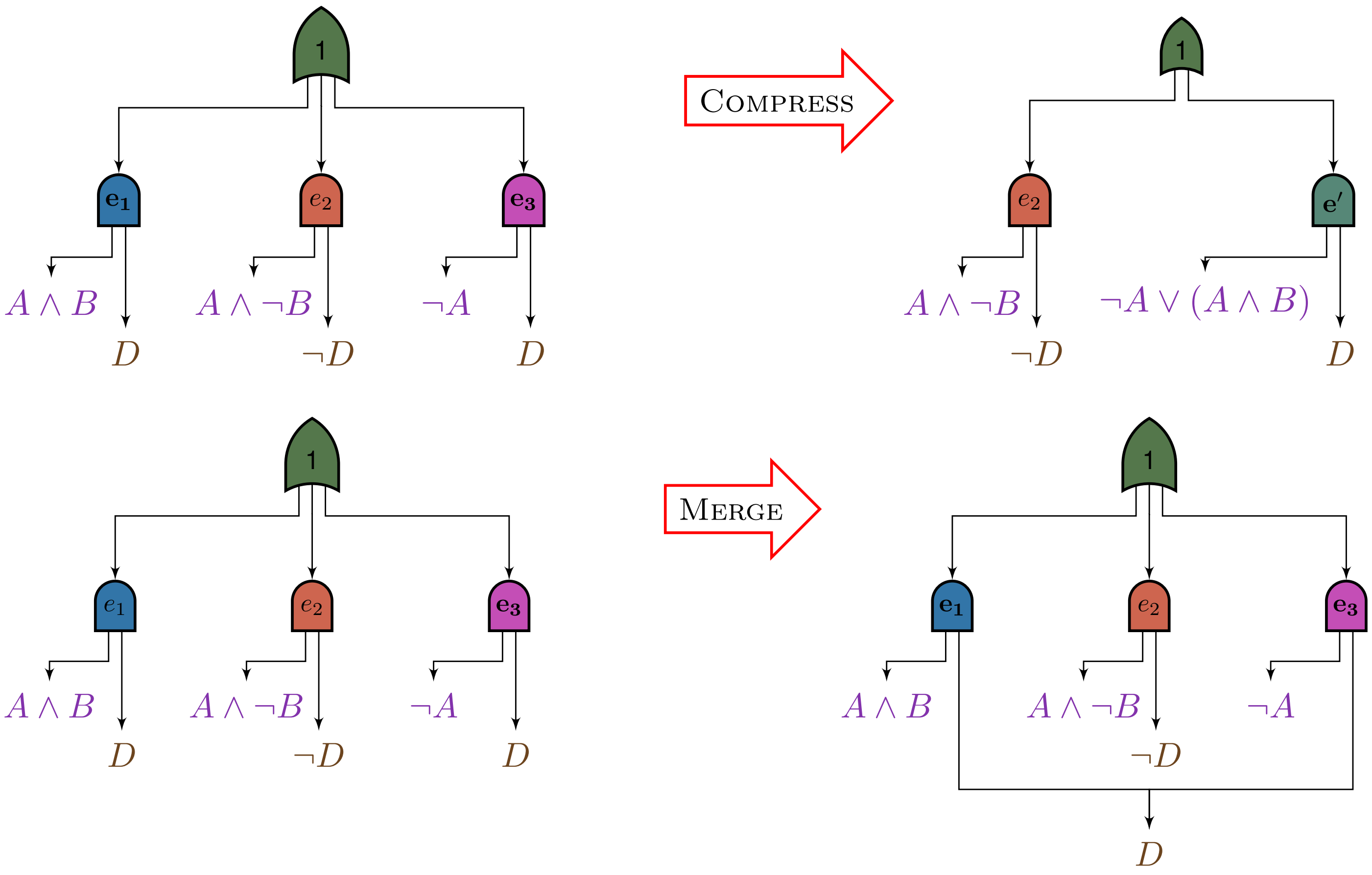


Now all s_i respect **S**



SamplePSDD

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



Darwiche [2011]

Experiments

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

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

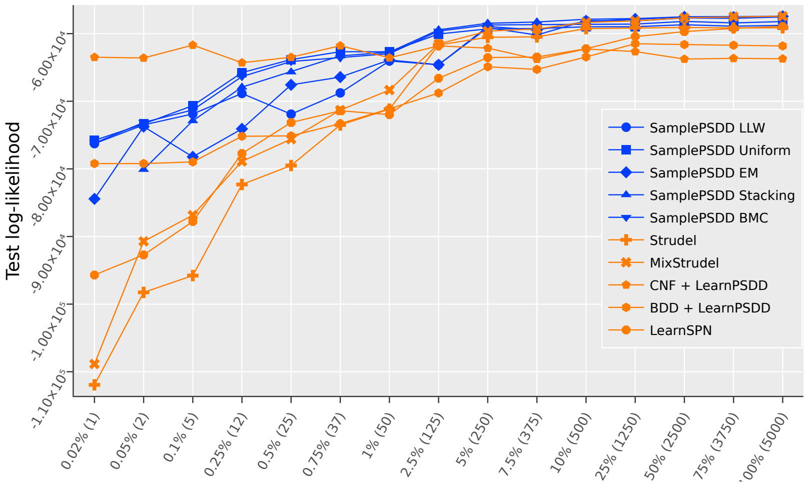
comparing against STRUDEL, LEARNPSDD and LEARNSPN.

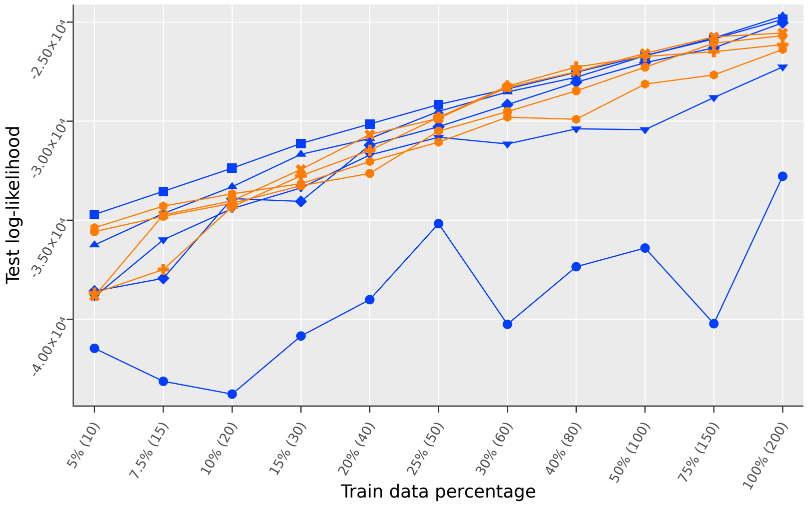
Datasets: we evaluate with 5 data + knowledge as logic constraints:

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

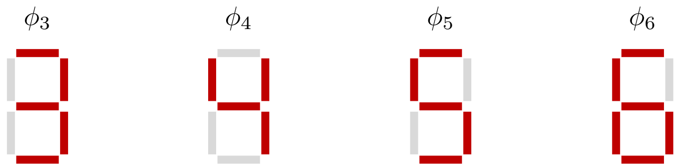
Our approach fares **better** with **fewer** data, yet remains **competitive** under **lots of data**.

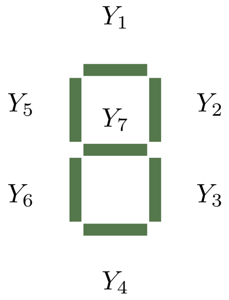
Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]





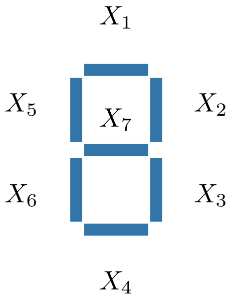
Experiments – LED

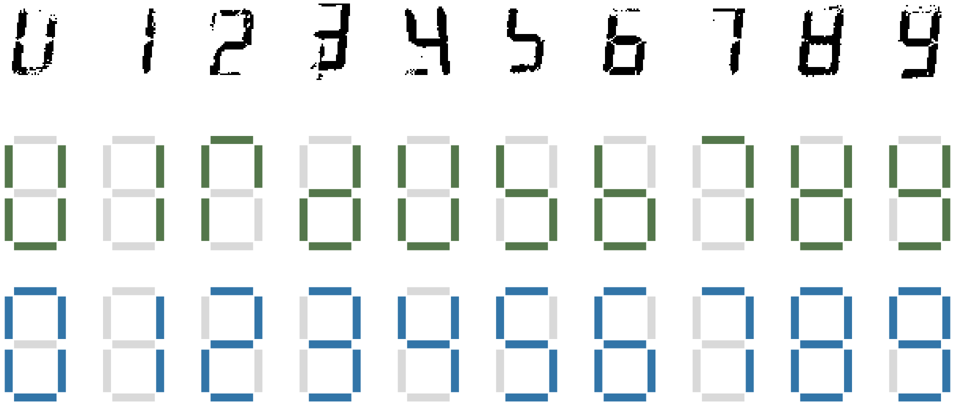




$$
\begin{aligned}
\phi_3 &= X_1 \wedge X_2 \wedge X_3 \wedge X_4 \wedge \neg X_5 \wedge \neg X_6 \wedge X_7 \\
\phi_4 &= \neg X_1 \wedge X_2 \wedge X_3 \wedge \neg X_4 \wedge \neg X_5 \wedge X_6 \wedge X_7 \\
\phi_5 &= X_1 \wedge \neg X_2 \wedge X_3 \wedge X_4 \wedge \neg X_5 \wedge X_6 \wedge X_7 \\
\phi_6 &= X_1 \wedge \neg X_2 \wedge X_3 \wedge X_4 \wedge X_5 \wedge X_6 \wedge X_7
\end{aligned}
$$

$$\phi = \bigvee_{i=0}^{9} \phi_i$$





Mattei et al. [2020]

Experiments

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

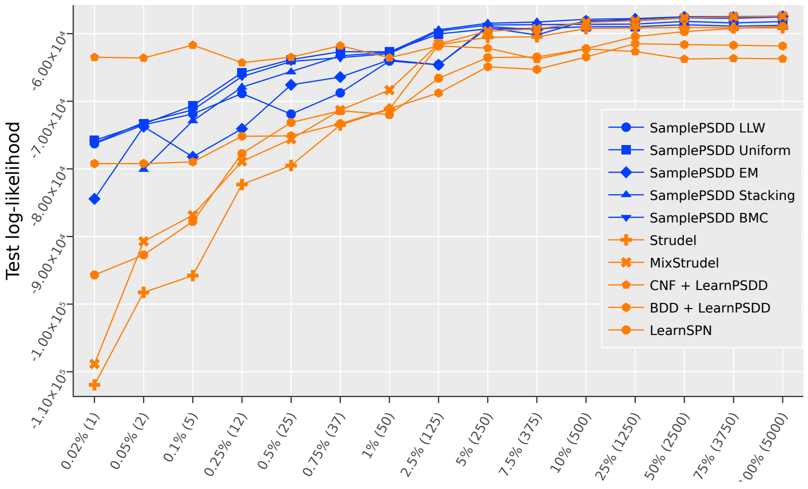
- ● **Likelihood weighting (LLW)**,
- ■ **Uniform weights**,
- ◆ **Expectation-Maximization (EM)**,
- ▲ **Stacking**,
- ▼ **Bayesian Model Combination (BMC)**;

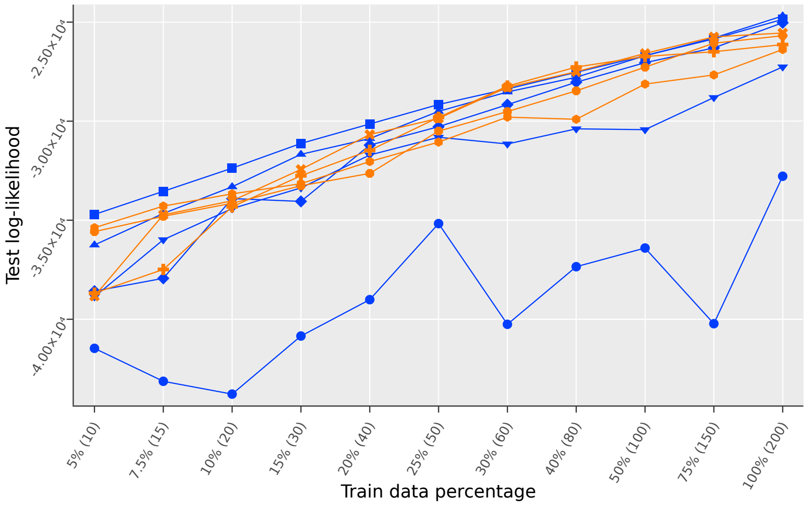
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

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

	Dataset	#vars	#train	ϕ's size
$\Rightarrow$	LED	14	5000	23
$\Rightarrow$	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)**;

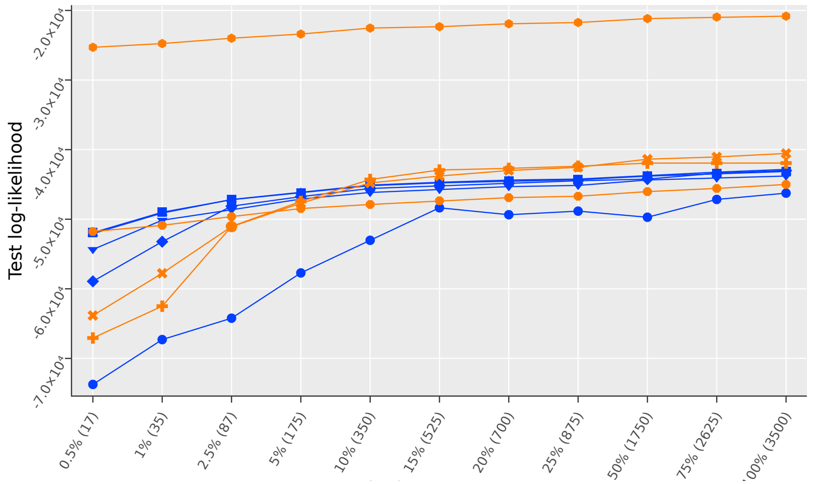
comparing against **Strudel**, **LearnPSDD** and **LearnSPN**.

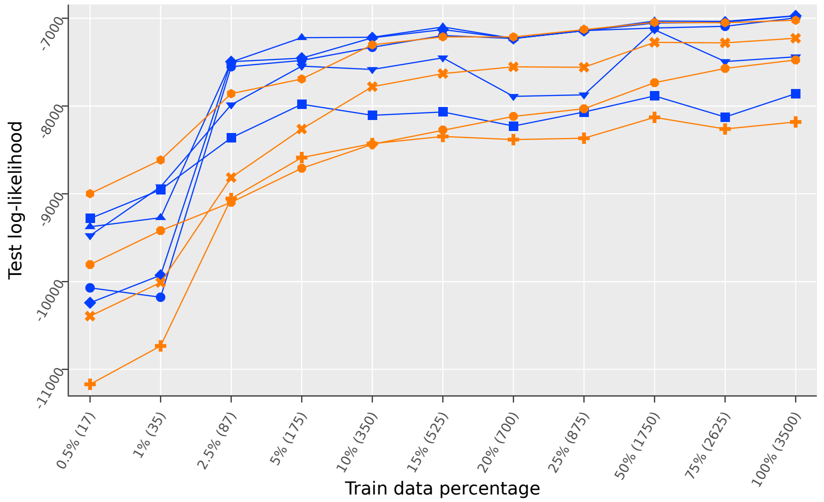
Datasets: we evaluate with 5 data + knowledge as logic constraints:

	Dataset	#vars	#train	ϕ's size
	LED	14	5000	23
	LED + Images	157	700	39899
$\Rightarrow$	Sushi Ranking	100	3500	17413
$\Rightarrow$	Sushi Top 5	10	3500	37
	Dota 2 Games	227	92650	1308

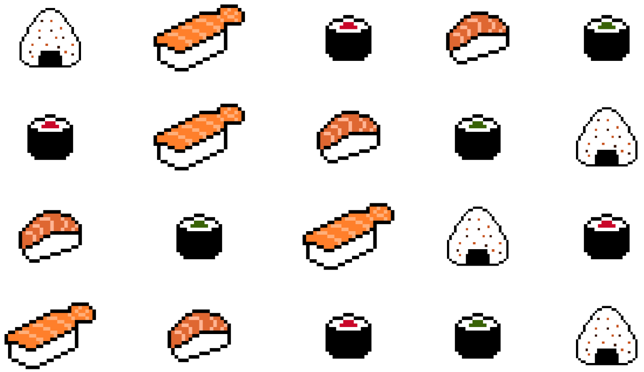
Our approach fares **better** with **fewer** data, yet remains **competitive** under **lots of data**.

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



Alice:

Bob:

Carol:

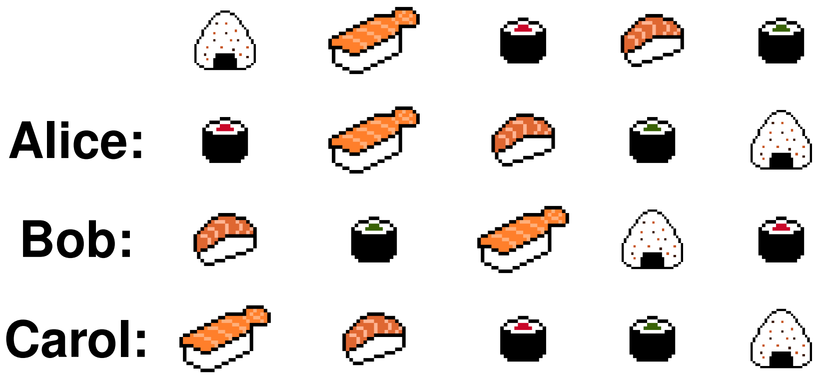
n sushi types and k rank positions

$$\alpha = \underbrace{\begin{aligned} &(\ \ X_{i1} \wedge \neg X_{i2} \wedge \cdots \wedge \neg X_{ik}) \\ \vee&(\neg X_{i1} \wedge \ \ X_{i2} \wedge \cdots \wedge \neg X_{ik}) \\ &\qquad\qquad\vdots \\ \vee&(\neg X_{i1} \wedge \neg X_{i2} \wedge \cdots \wedge \ \ X_{ik}) \end{aligned}}_{\text{Rank position}} \qquad \beta = \underbrace{\begin{aligned} &(\ \ X_{1j} \wedge \neg X_{2j} \wedge \cdots \wedge \neg X_{nj}) \\ \vee&(\neg X_{1j} \wedge \ \ X_{2j} \wedge \cdots \wedge \neg X_{nj}) \\ &\qquad\qquad\vdots \\ \vee&(\neg X_{1j} \wedge \neg X_{2j} \wedge \cdots \wedge \ \ X_{nj}) \end{aligned}}_{\text{Type uniqueness}}$$

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

Kamishima [2003], Shen et al. [2017]

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 $\text{Exactly}(k, n)$

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

Kamishima [2003], Choi et al. [2015]

Experiments

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

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

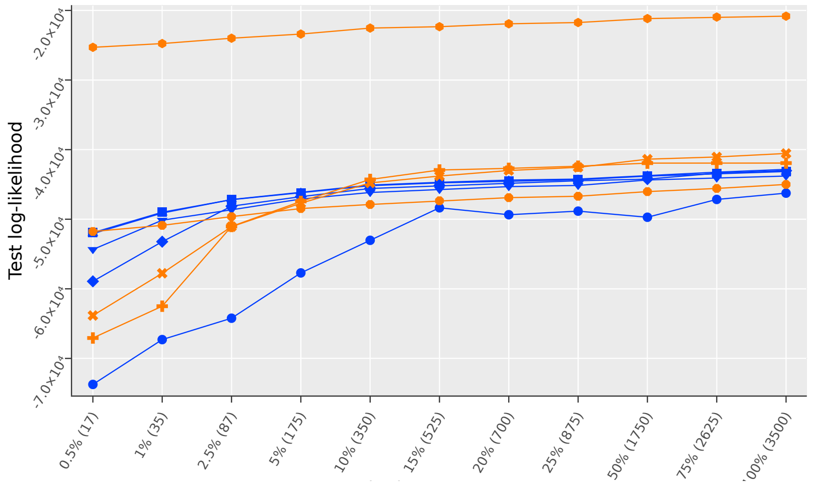
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

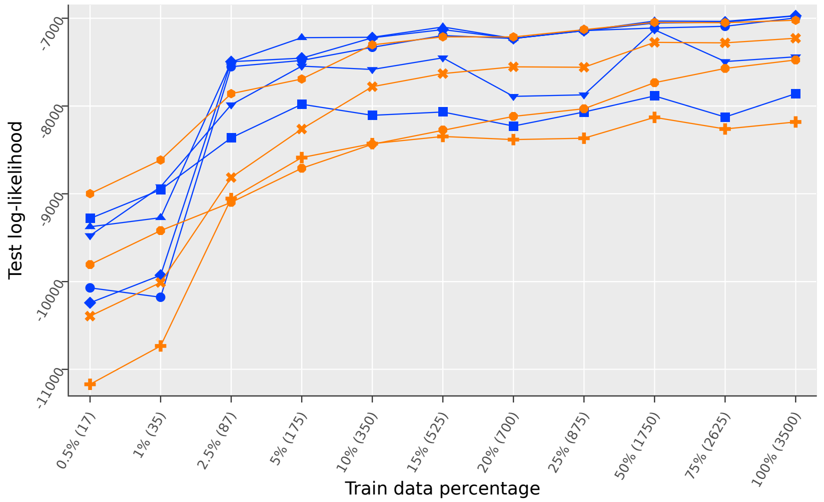
Datasets: we evaluate with 5 data + knowledge as logic constraints:

	Dataset	#vars	#train	ϕ's size
	LED	14	5000	23
	LED + IMAGES	157	700	39899
$\Rightarrow$	SUSHI RANKING	100	3500	17413
$\Rightarrow$	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)**;

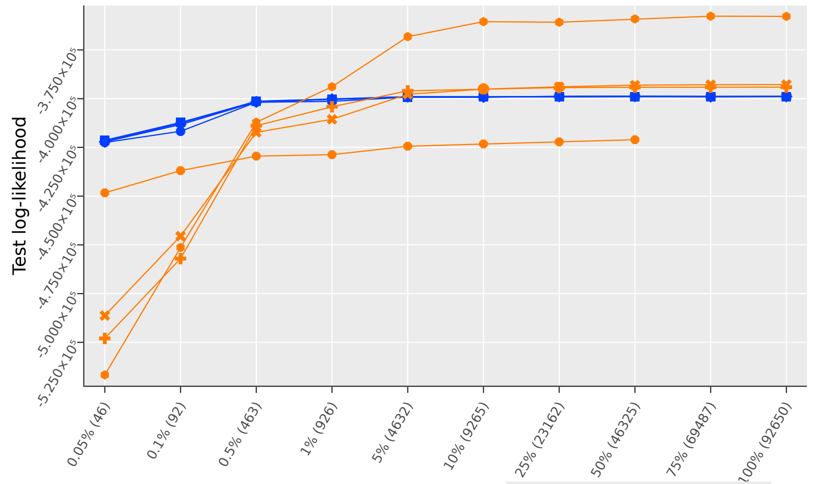
comparing against **Strudel**, **LearnPSDD** and **LearnSPN**.

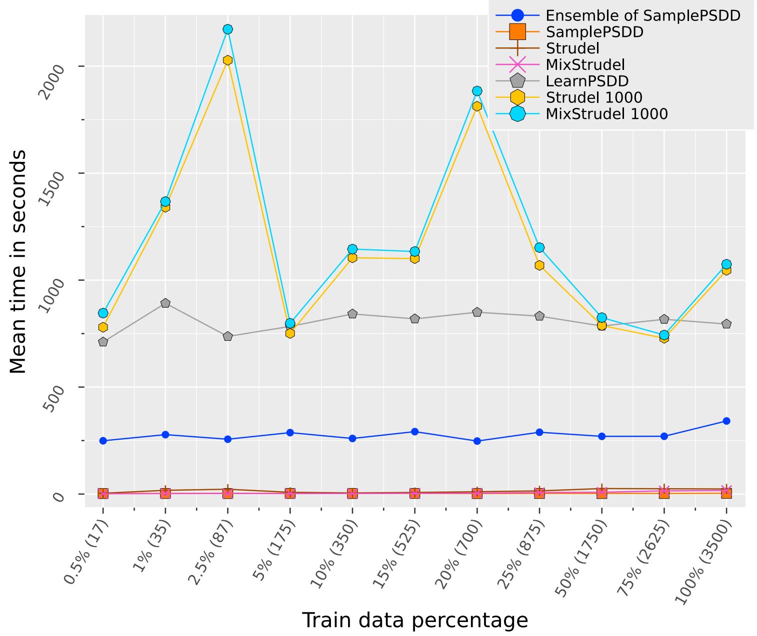
Datasets: we evaluate with 5 data + knowledge as logic constraints:

	Dataset	#vars	#train	ϕ's size
	LED	14	5000	23
	LED + Images	157	700	39899
	Sushi Ranking	100	3500	17413
	Sushi Top 5	10	3500	37
$\Rightarrow$	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

n characters, k for each team

Y_5	Y_4	Y_3	Y_2	Y_1
25	20	13	8	5

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

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

$$\underbrace{\gamma = X_i \neq Y_j, \forall X_i, Y_j}_{\text{Intractable as BDD}}$$

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

Tridgell [2016], Sinz [2005], Nishino et al. [2016], Eén and Sörensson [2006]

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.

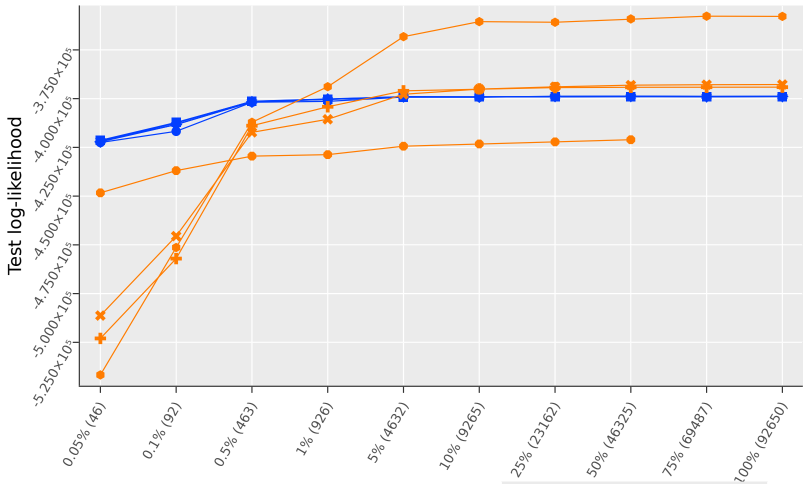
Datasets: we evaluate with 5 data + knowledge as logic constraints:

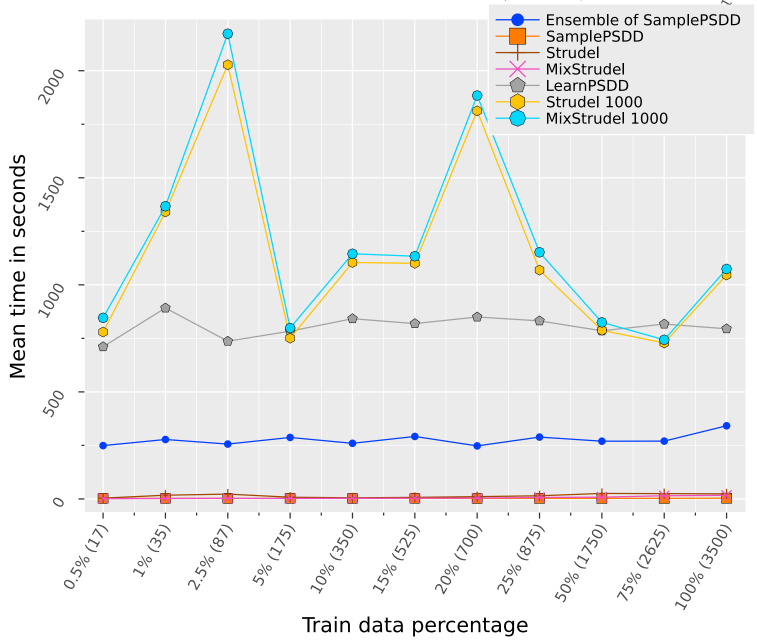
	Dataset	#vars	#train	ϕ's size
	LED	14	5000	23
	LED + IMAGES	157	700	39899
	SUSHI RANKING	100	3500	17413
	SUSHI TOP 5	10	3500	37
$\Rightarrow$	DOTA 2 GAMES	227	92650	1308

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]

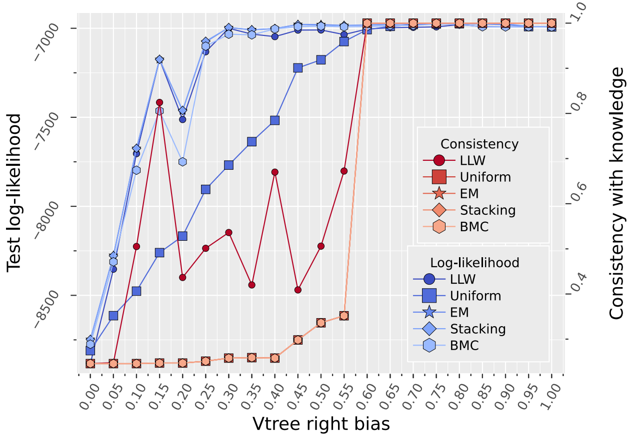


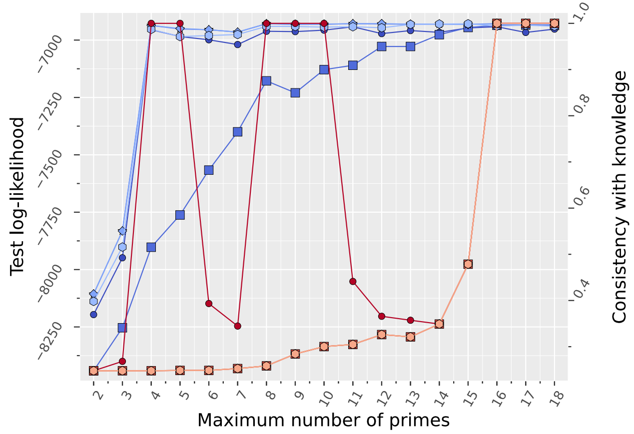


SamplePSDD – Experiments

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

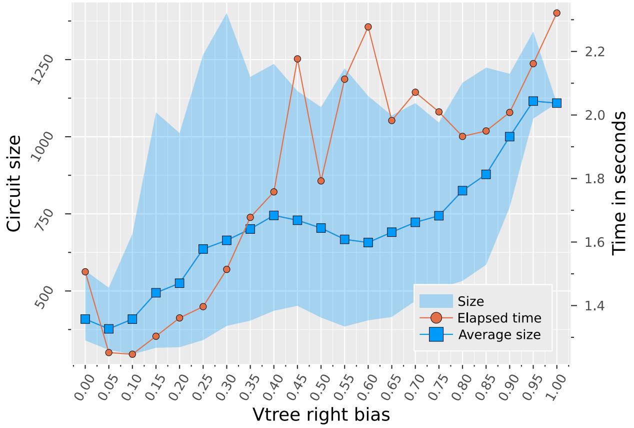
in log-likelihood and consistency?

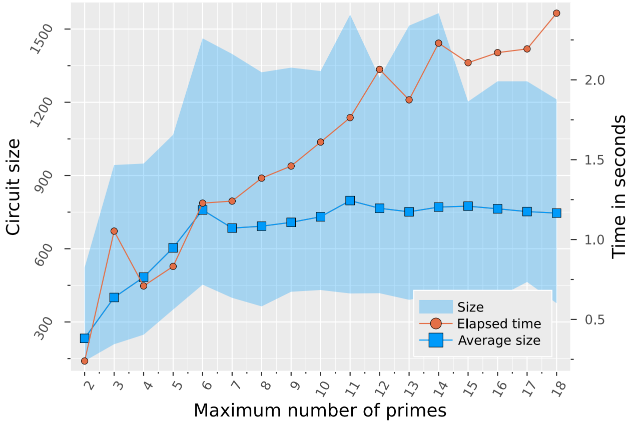




Samples perform **better** with higher k's and right-leaning vtrees...

...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	$\begin{cases}\mathcal{O}(nkmc) & \text{, if sum}\\ \mathcal{O}(nm^3) & \text{, if product}\end{cases}$	≥ 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	$\begin{cases}\mathcal{O}(nkmc) & \text{, if sum}\\ \mathcal{O}(m(\log m)^2) & \text{, if product}\end{cases}$	≥ 1	✗	✓	✓	✗	✗	✓	✓	✓	Jaini et al. [2018a]
LearnPSDD	INCR	$\begin{cases}\mathcal{O}(m^2) & \text{, top-down vtree}\\ \mathcal{O}(m^4) & \text{, bottom-up vtree}\\ \mathcal{O}(i\lvert\mathcal{C}\rvert^2) & \text{, circuit structure}\end{cases}$	1	✓	✓	✓	✓	✓	✓	✗	✗	Liang et al. [2017]
Strudel	INCR	$\begin{cases}\mathcal{O}(m^2 n) & \text{, CLT + vtree}\\ \mathcal{O}(i(\lvert\mathcal{C}\rvert n+m^2)) & \text{, circuit structure}\end{cases}$	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]
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	✓	✓	✓	✓	✓	✓	✗	✗	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	✗	✓	✓	✗	✓	✓	✓	✓	To appear

A Data Perspective

Motivation

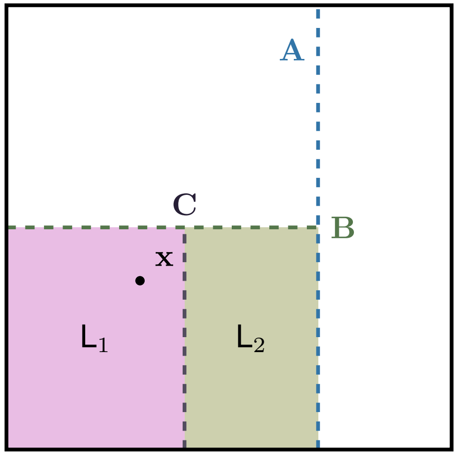
Density Estimation Trees...

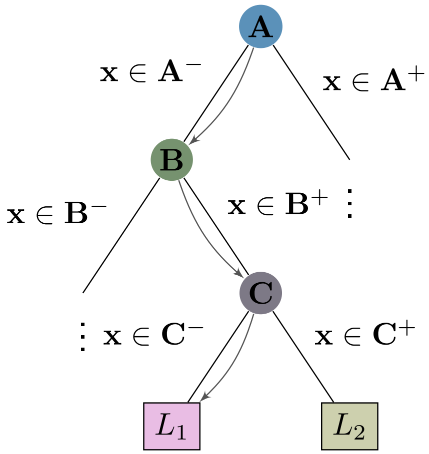
- ☑ ...are fast;
- ☑ ...are interpretable;
- ☑ ...are (somewhat) explainable;
- ☑ ...have extensive literature coverage;
- ☒ ...are not so expressive;
- ☒ ...only accept marginalization queries;
- ☒ ...are not so accurate;

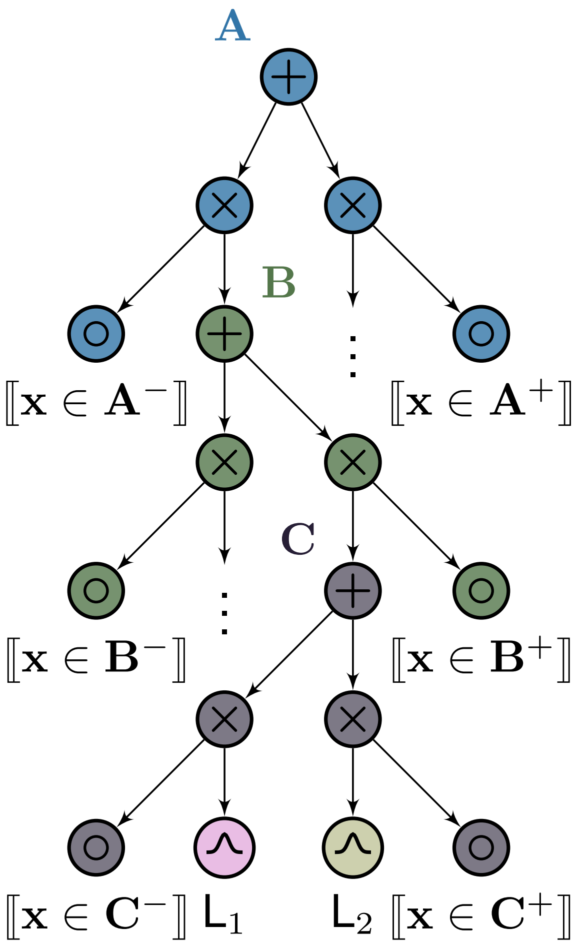
...but are subsumed by circuits!

Learn DETs $\subseteq$ Learn PCs?

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

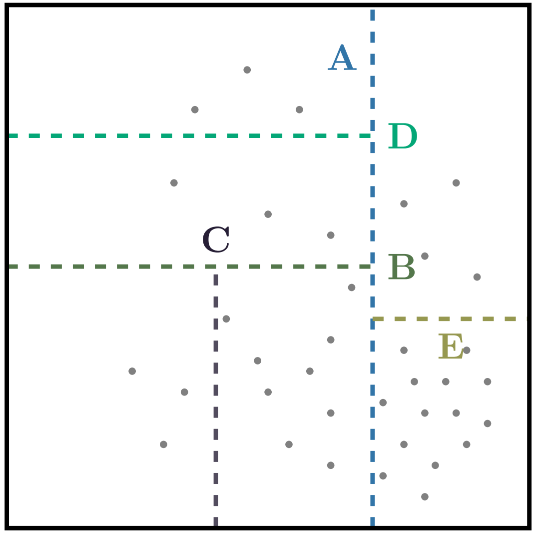




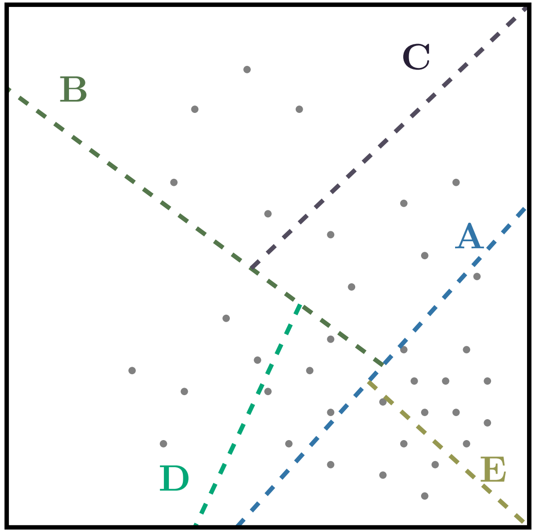


Correia et al. [2020], Ram and Gray [2011]

Random Projections



Axis-aligned projections



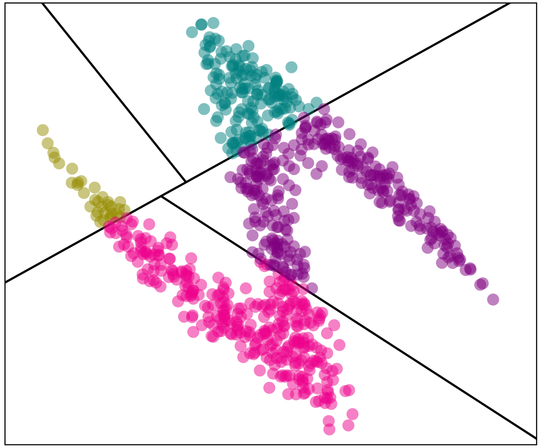
Random projections

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.

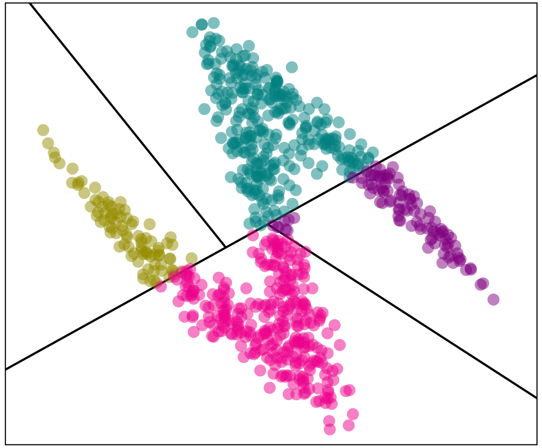
Freund et al. [2008], Dasgupta and Freund [2008]

Random Projections

SPLITMAX



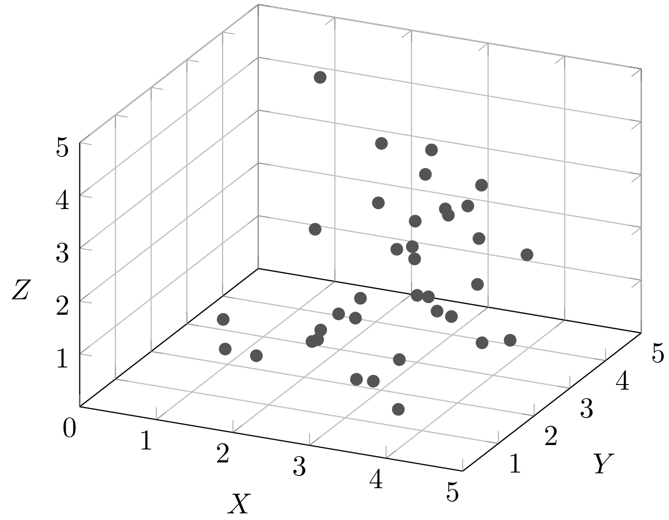
SPLITSID

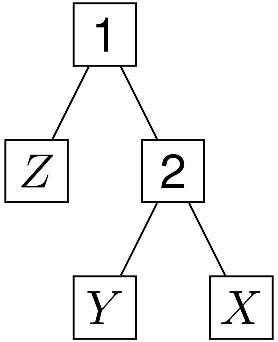


If the data has *intrinsic dimension* d, then with constant probability the part of the data at level d or higher of the tree has average diameter less than half of the data.

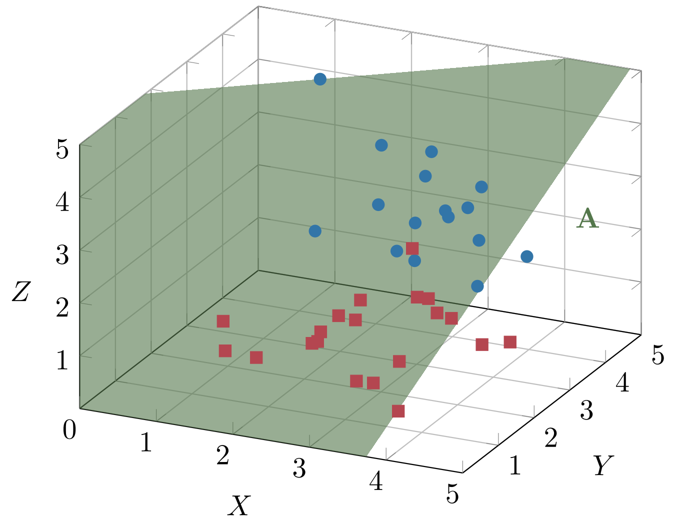
Freund et al. [2008], Dasgupta and Freund [2008]

LearnRP

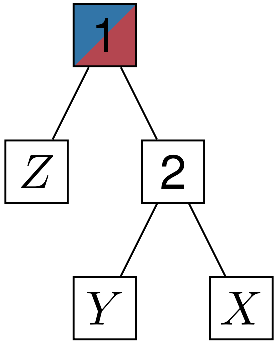


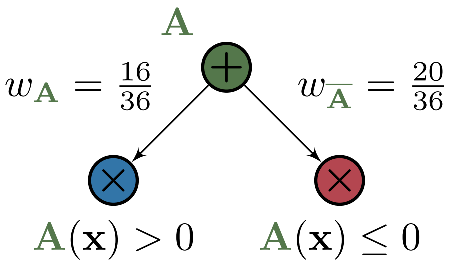


LearnRP



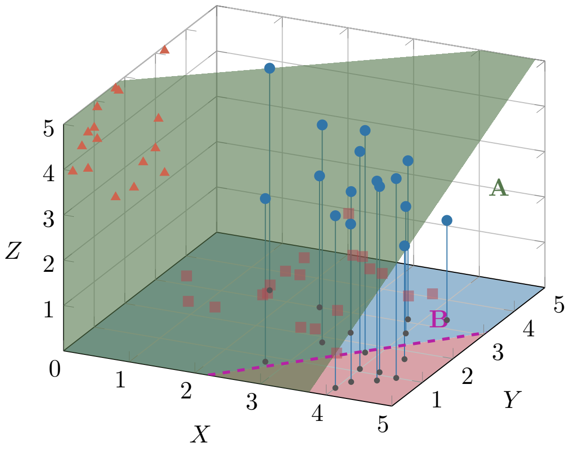
$$\mathbf{A}(x, y, z) = \begin{bmatrix} x & y & z \end{bmatrix} \cdot \underbrace{\begin{bmatrix} -0.31 \\ -0.40 \\ 0.85 \end{bmatrix}}_{a} + \underbrace{1}_{\theta}$$

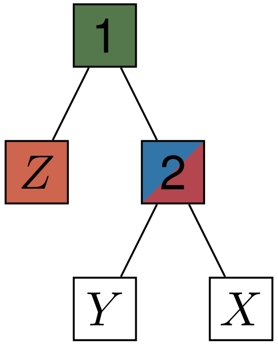


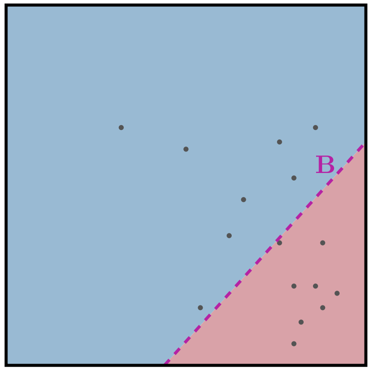


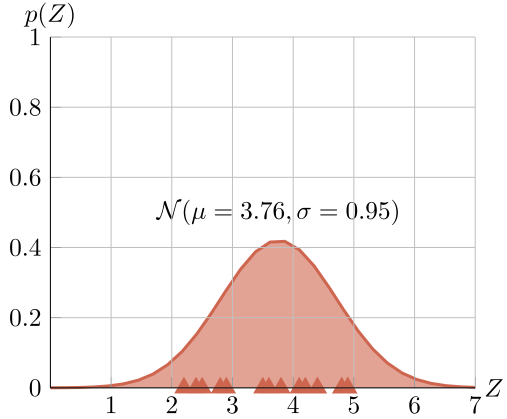
$w_{\mathbf{A}}$: probability of $\mathbf{A}(\mathbf{x}) > 0$

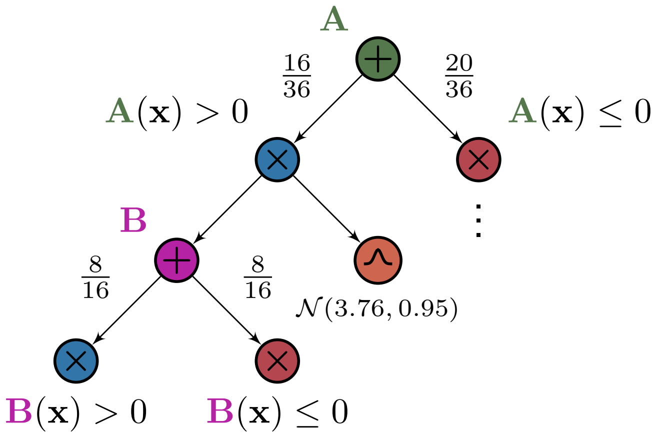
LearnRP











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

Parameter Optimization

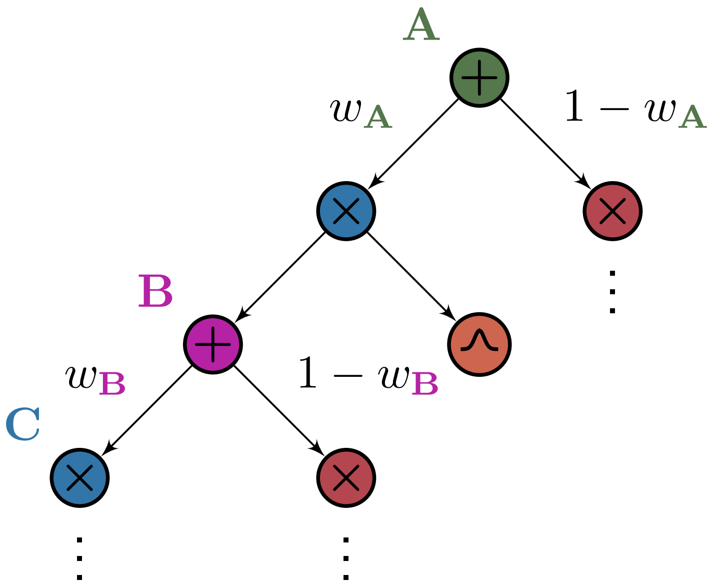
Expectation-Maximization (EM)

- 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

Dempster et al. [1977], Liu and Van den Broeck [2021]

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	$\mathbb{R}$

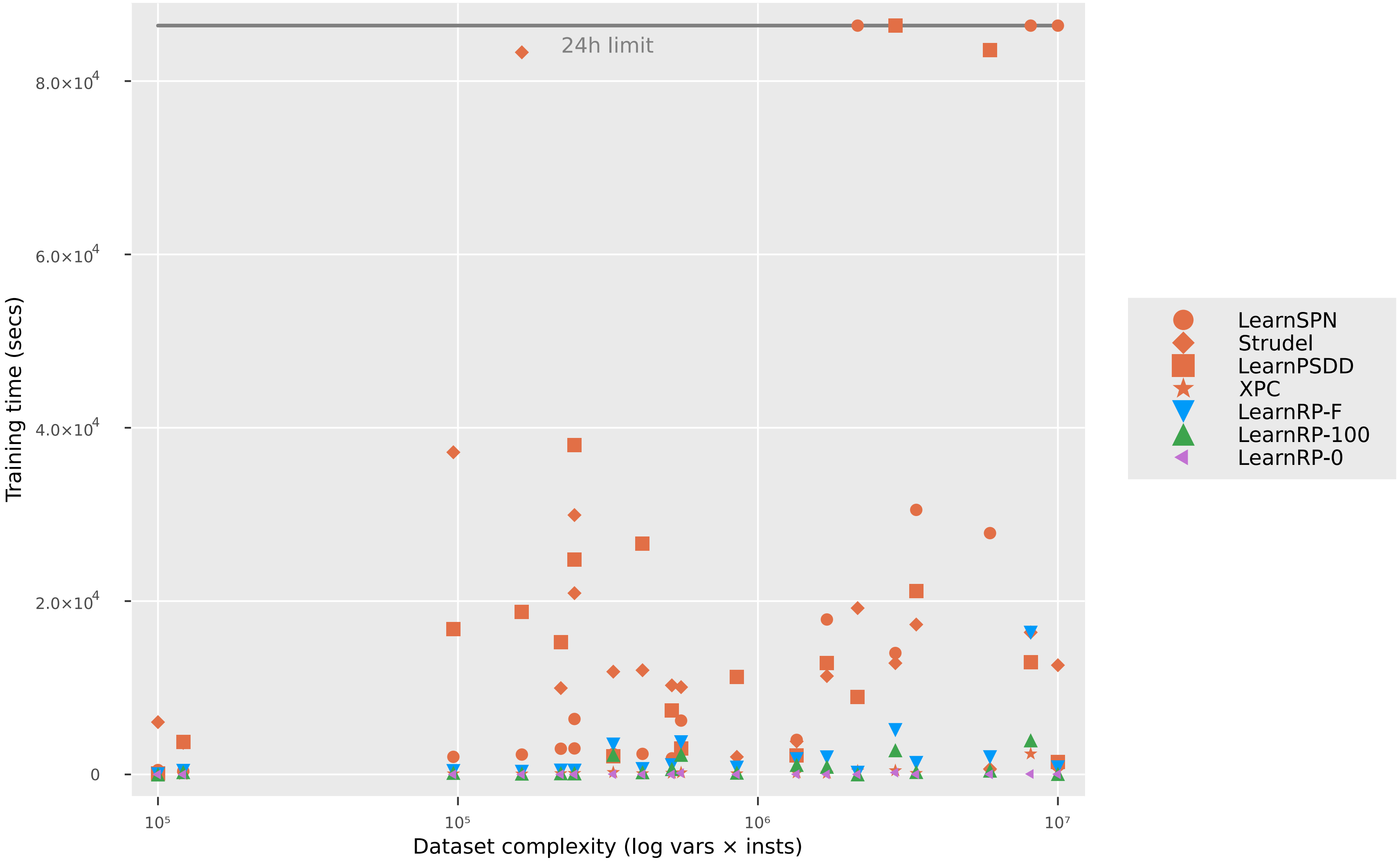
Lowd and Davis [2010], Van Haaren and Davis [2012], Dua and Graff [2017], Güvenir and Uysal [2000]

Experiments

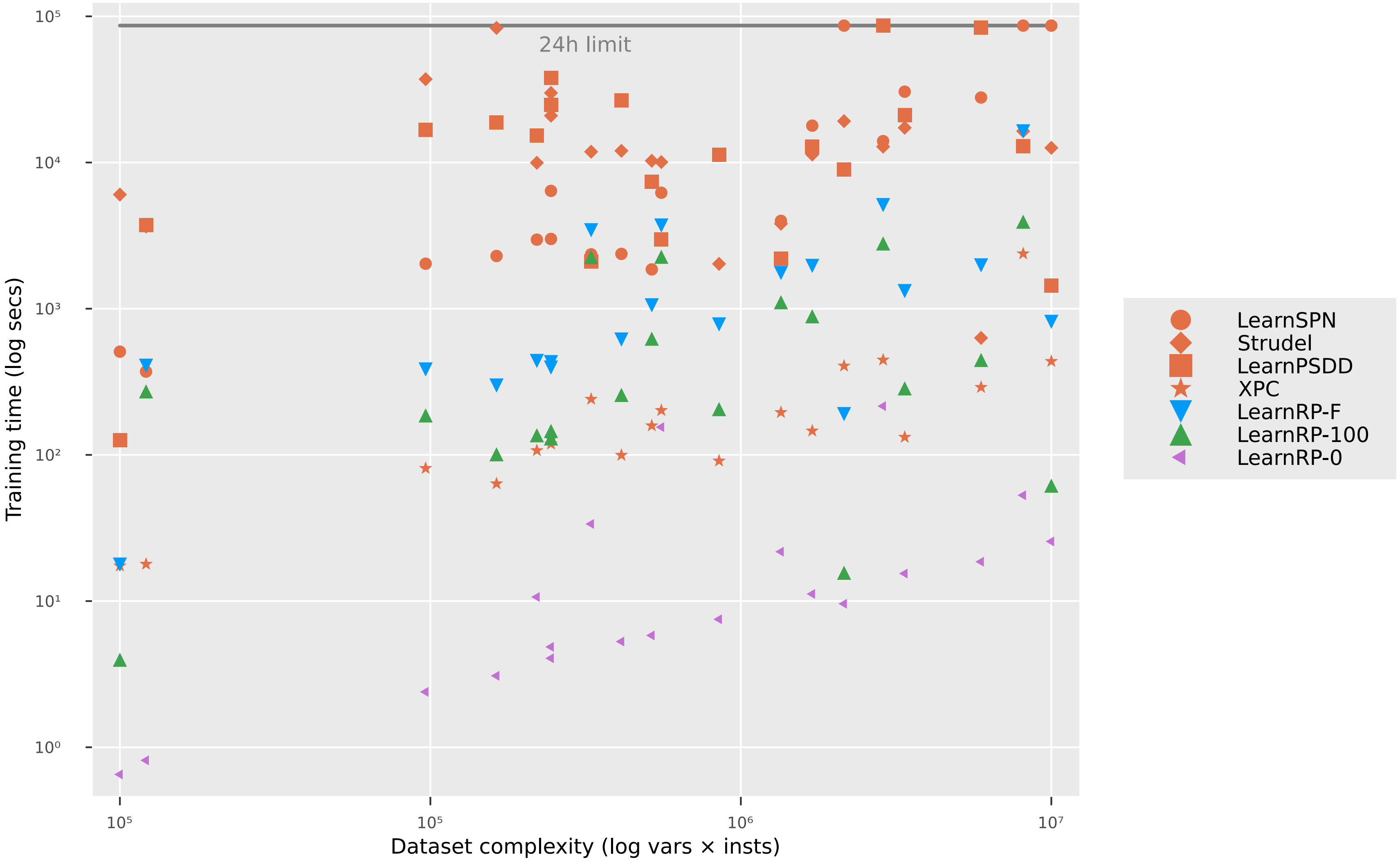
Dataset	LEARNSPN	STRUDEL	LEARNPSDD	XPC	PROMETHEUS	LEARNRP-F	LEARNRP-100
ACCIDENTS	-30.03	|-28.73|	-30.16	-31.02	**-27.91**	-28.66	-28.81
AD	-19.73	-16.38	-31.78	**-15.50**	-23.96	|-19.26|	-19.99
AUDIO	-40.50	-41.50	-39.94	-40.91	**-39.80**	|-40.27|	-40.30
BBC	|-250.68|	-254.41	-253.19	**-248.34**	-248.50	-254.15	-251.57
NETFLIX	|-57.02|	-58.69	**-55.71**	-57.58	-56.47	|-57.02|	-57.03
BOOK	-35.88	-34.99	-34.97	-34.75	|-34.40|	-33.56	**-33.41**
20-NEWSGRP	-155.92	-154.47	-155.97	|-153.75|	-154.17	-152.63	**-152.34**
REUTERS-52	|-85.06|	-86.22	-89.61	-84.70	**-84.59**	-85.69	-85.76
WEBKB	-158.20	-155.33	-161.09	|-153.67|	-155.21	-153.52	**-151.80**
DNA	**-82.52**	-86.22	-88.01	-86.61	-84.45	-83.57	|-83.62|
JESTER	-75.98	-55.03	**-51.29**	-53.43	-52.80	-52.92	|-52.86|
KDD	-2.18	|-2.13|	**-2.11**	-2.15	-2.12	-2.14	-2.14
KOSAREK	-10.98	-10.68	**-10.52**	-10.77	-10.59	|-10.62|	-10.66
MSNBC	-6.11	**-6.04**	**-6.04**	|-6.18|	**-6.04**	-6.33	-6.35
MSWEB	-10.25	**-9.71**	|-9.89|	-9.93	-9.86	-9.90	-9.93
NLTCS	-6.11	-6.06	**-5.99**	|-6.05|	-6.01	-6.22	-6.27
PLANTS	-12.97	|-12.98|	-13.02	-14.19	**-12.81**	-13.77	-13.81
PUMSB-STAR	|-24.78|	-24.12	-26.12	-26.06	**-22.75**	-26.12	-26.33
EACHMOVIE	-52.48	-53.67	-58.01	-54.82	|-51.49|	-51.41	**-50.95**
RETAIL	-11.04	-10.81	**-10.72**	-10.94	-10.87	|-10.84|	-10.86
Avg. Rank	4.83± 1.89	4.30± 1.92	|4.03± 2.57|	4.62± 1.88	**2.50± 1.43**	3.62± 1.47	4.10± 1.98
Pos. (mean)	7th	5th	|3rd|	6th	**1st**	2nd	4th

Gens and Domingos [2013], Dang et al. [2020], Liang et al. [2017], Mauro et al. [2021], Jaini et al. [2018a]

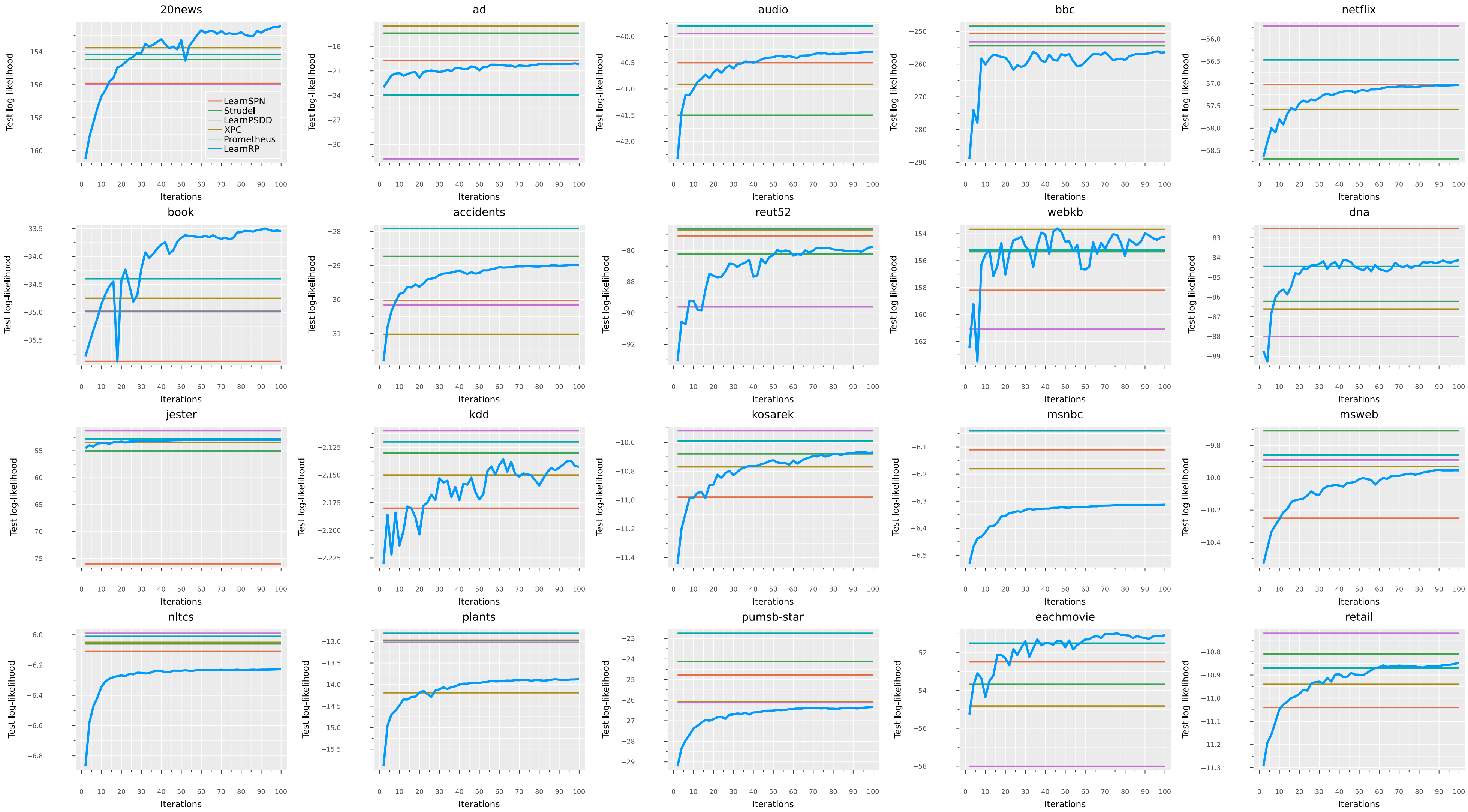
Experiments



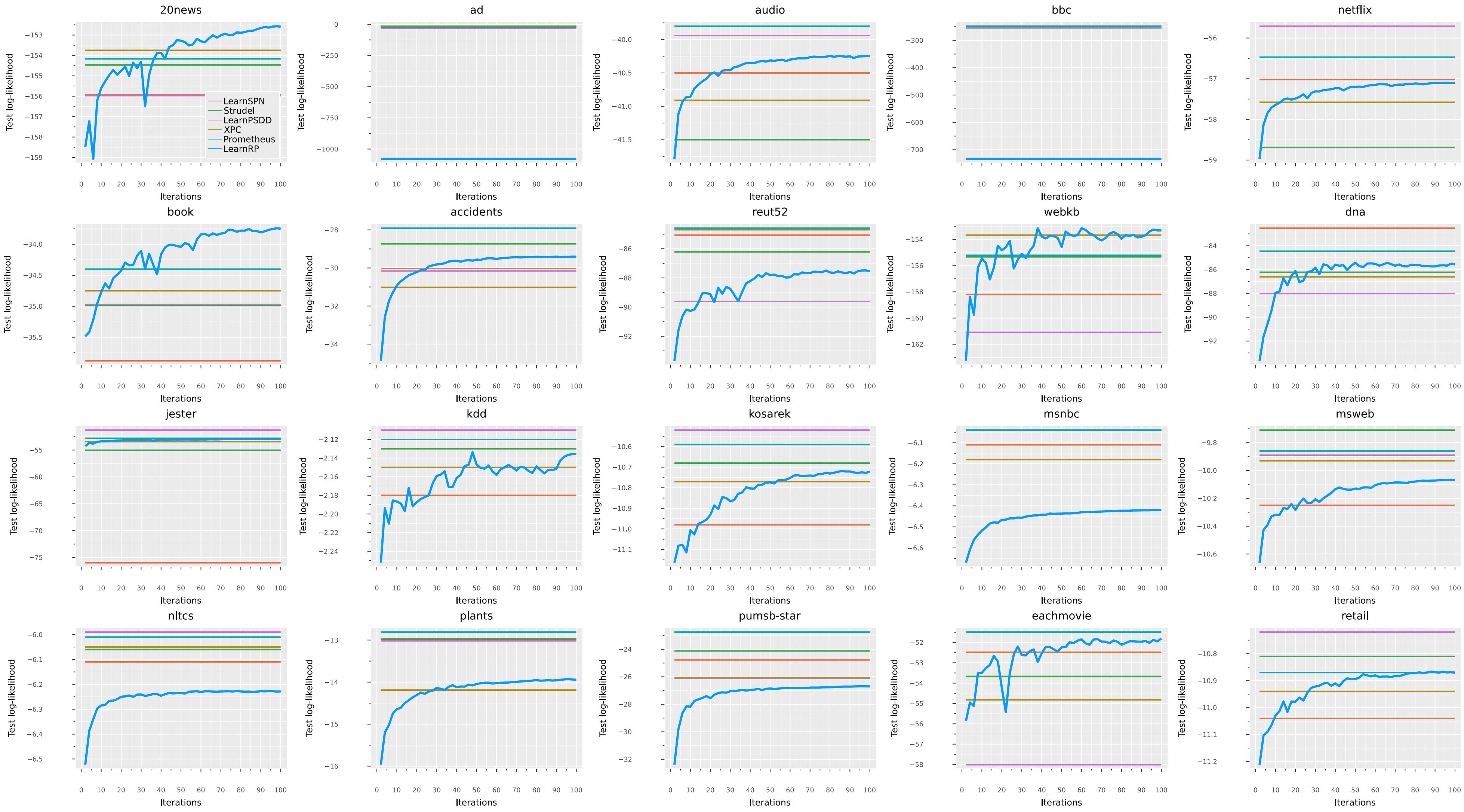
Experiments



LEARNRP – Learning Curves



LEARNRP – Random Initializations



Experiments

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

Jaini et al. [2018a], Salakhutdinov and Hinton [2009], Rasmussen [2000], Jaini et al. [2018b], Hsu et al. [2017], Trapp et al. [2016], Dua and Graff [2017], Güvenir and Uysal [2000]

In conclusion

Contributions

Literature review

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

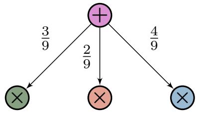
SamplePSDD

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

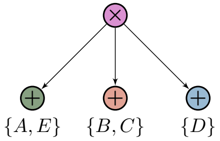
LearnRP

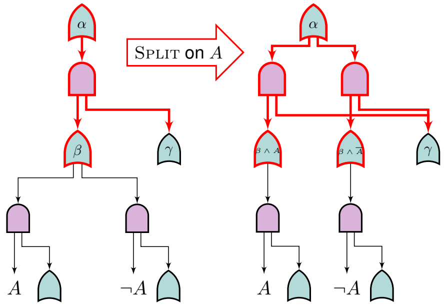
- Simple strategy;
- Inspiration from known DET literature;
- Orders of magnitude faster;
- Competitive performance.

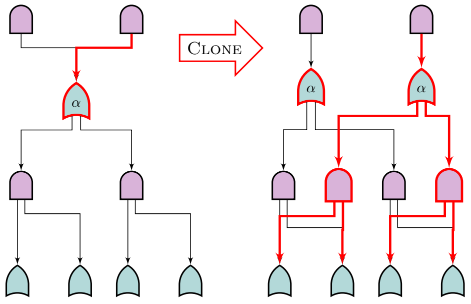
A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1

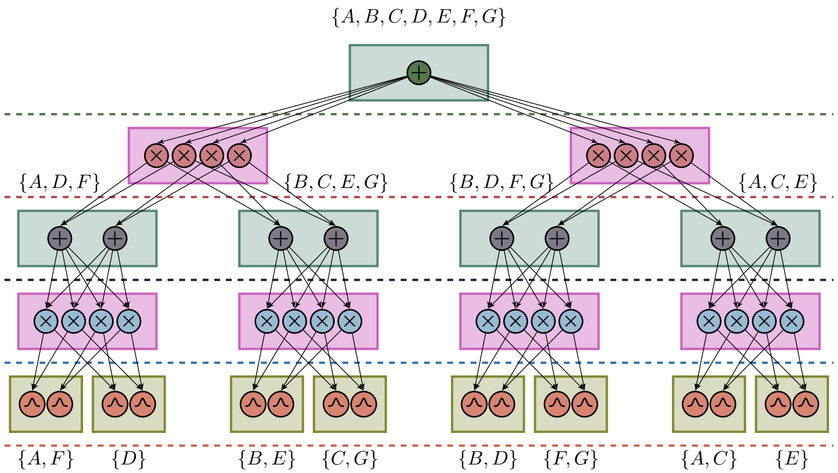


A	B	C	D	E
0	1	0	0	1
1	0	1	1	1
1	1	0	1	1
0	0	1	0	0
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	0	0
0	1	1	0	1









Contributions

Literature review

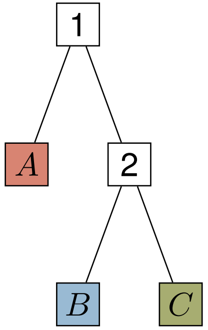
- Systematic review of literature;
- Taxonomy of popular algorithms;
- Complexity analysis;
- Pros and cons.

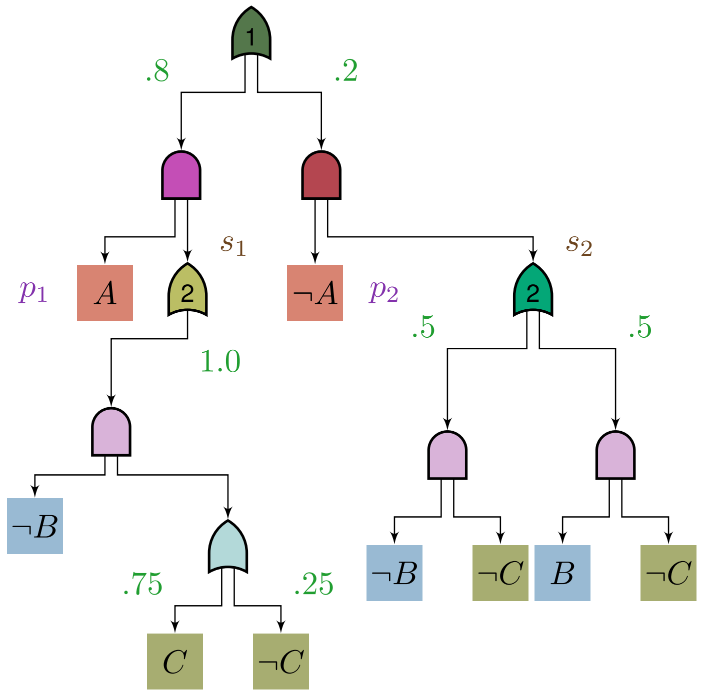
A	B	C	$p(\mathbf{x})$
0	0	0	0.1
0	1	0	0.1
1	0	0	0.2
1	0	1	0.6

SAMPLEPSDD

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

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





LEARNRP

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

Contributions

Literature review

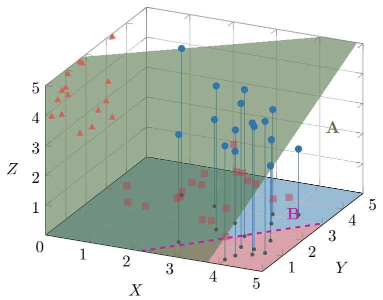
- Systematic review of literature;
- Taxonomy of popular algorithms;
- Complexity analysis;
- Pros and cons.

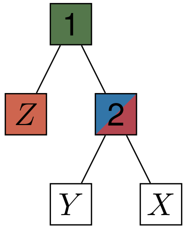
SamplePSDD

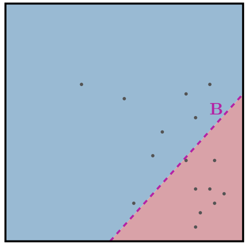
- Consistent with a relaxation of a formula;
- Relaxation as a function of vtree and sampling;
- Compromise between tractability and consistency;
- Ensembles mitigate relaxation.

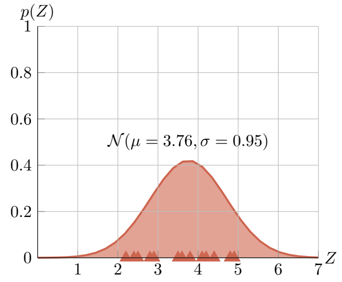
LearnRP

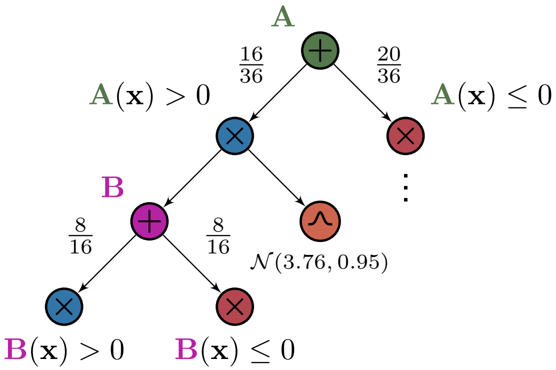
- Simple strategy;
- Inspiration from known DET literature;
- Orders of magnitude faster;
- Competitive performance.











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

Supplemental Material

LearnRP – Binary Benchmark

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

Gens and Domingos [2013], Dang et al. [2020], Liang et al. [2017], Mauro et al. [2021], Jaini et al. [2018a]