

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...

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

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

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

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

...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...

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

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

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

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

Marginals

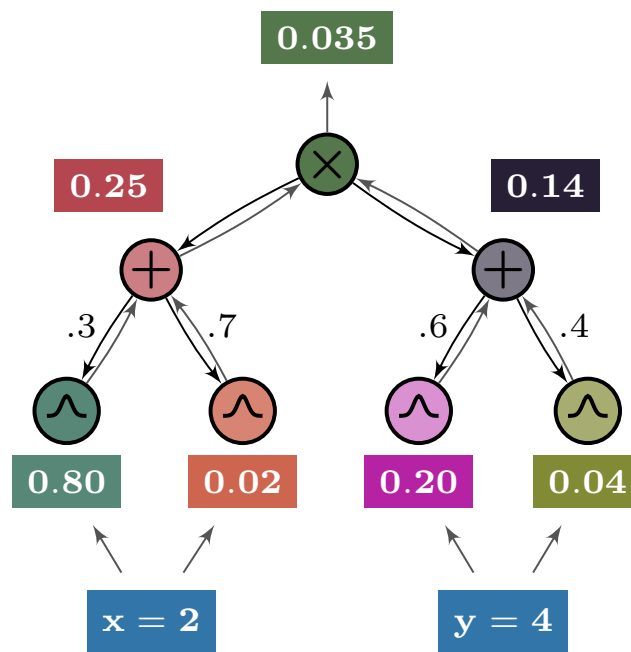
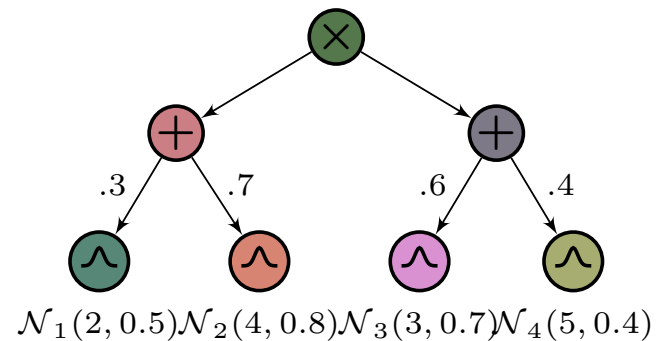
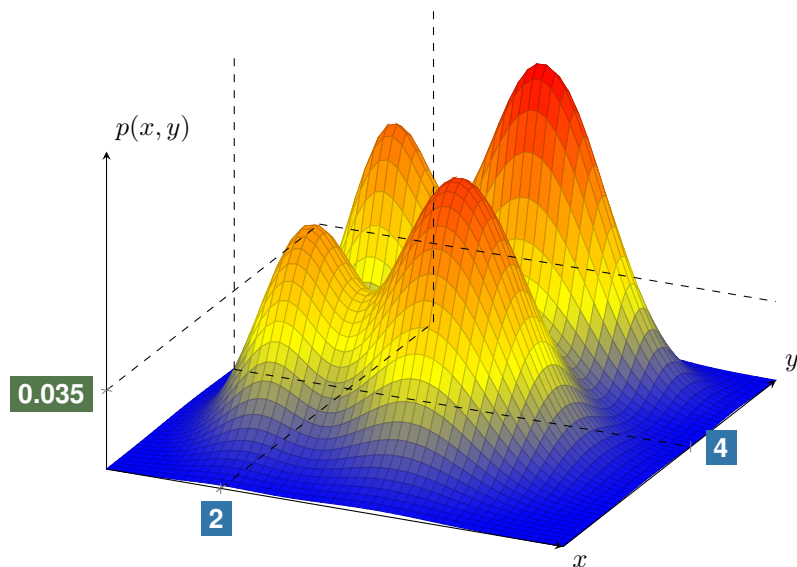
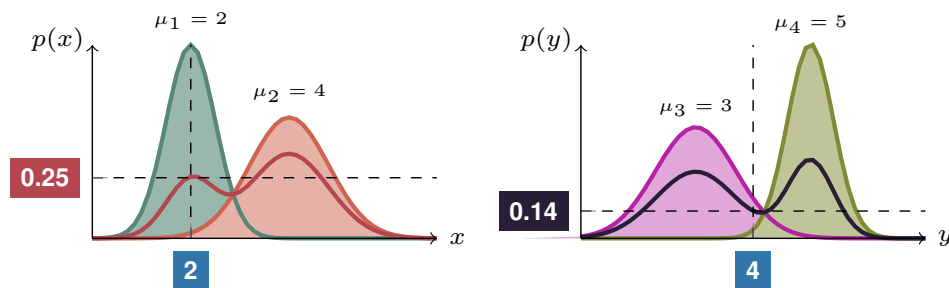
Conditionals

MPE

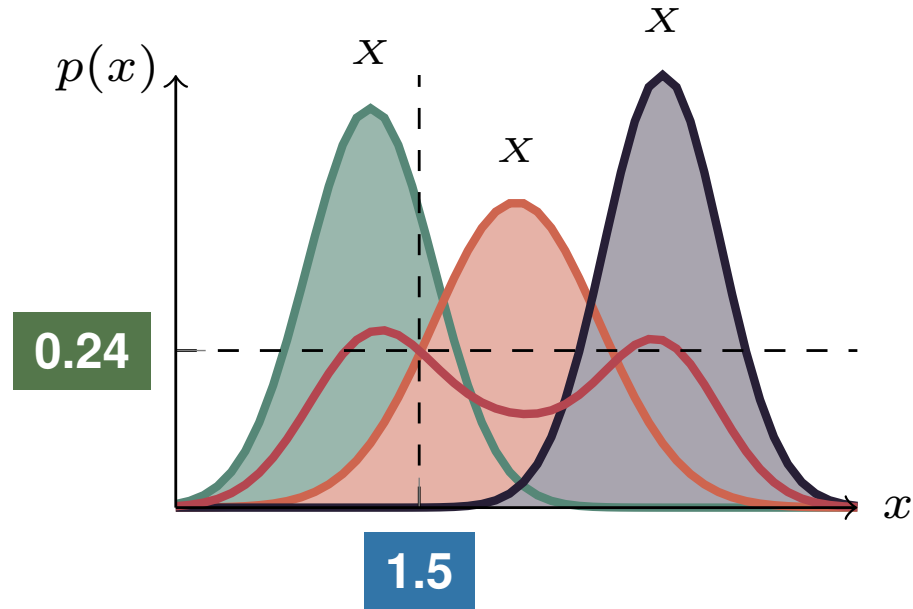
Logical events

...and extract meaningful queries from it?

Probabilistic Circuits

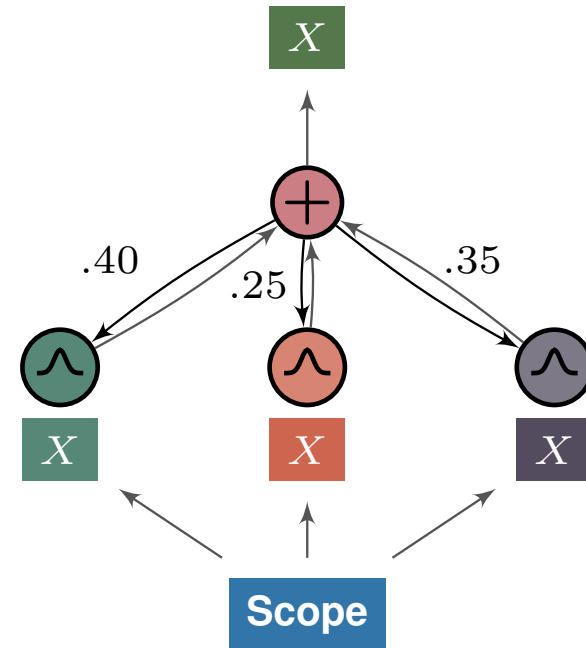
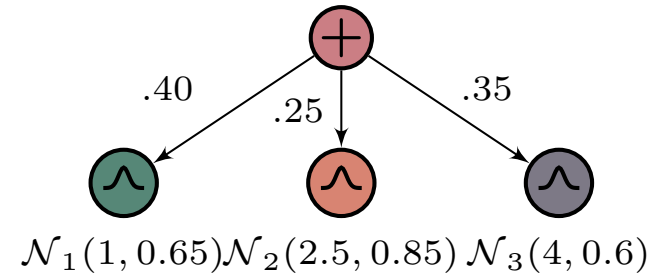


Probabilistic Circuits – Smoothness

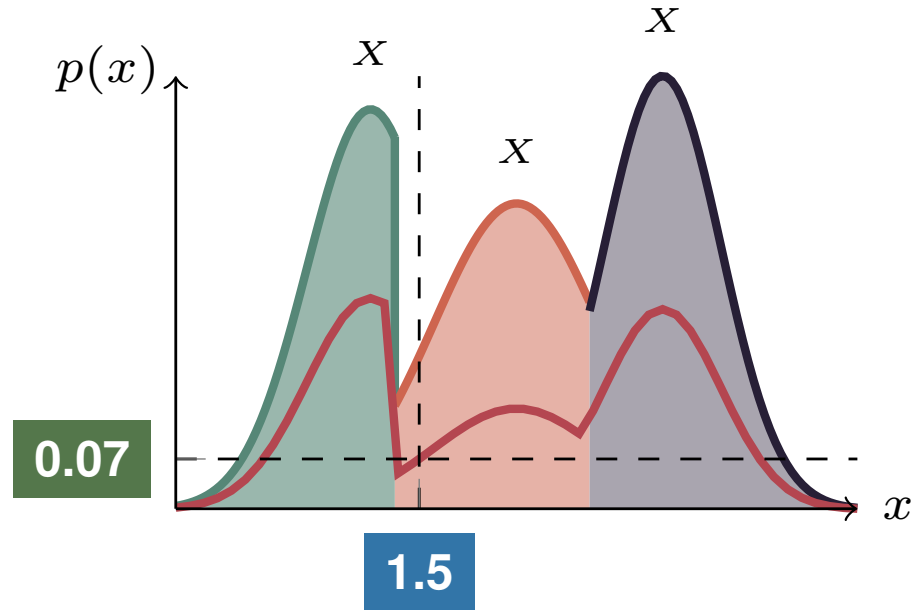


Definition 1 (Smoothness).

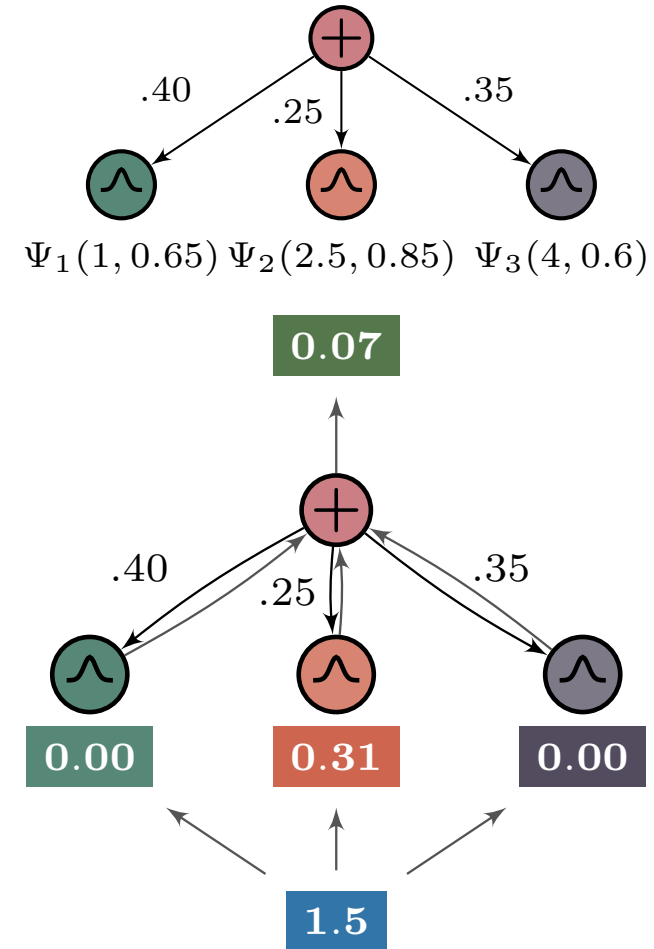
Every sum node child mentions the same variables.



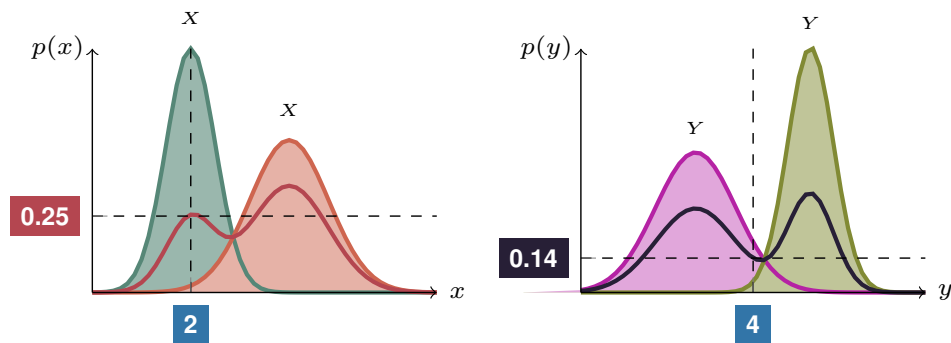
Probabilistic Circuits – Determinism



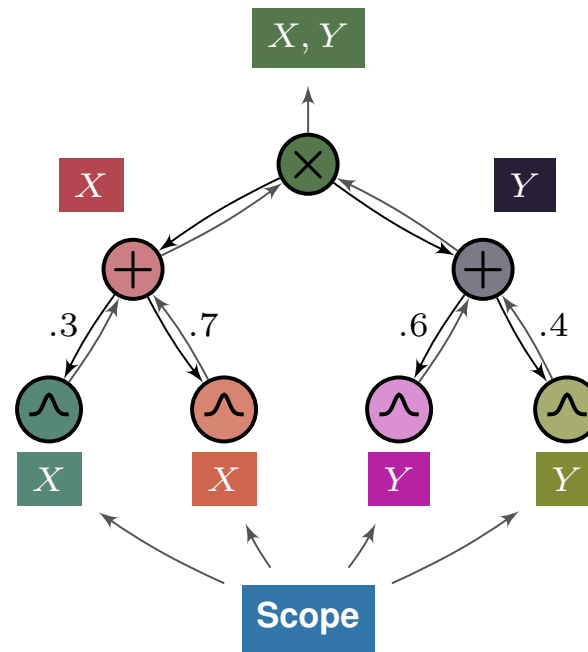
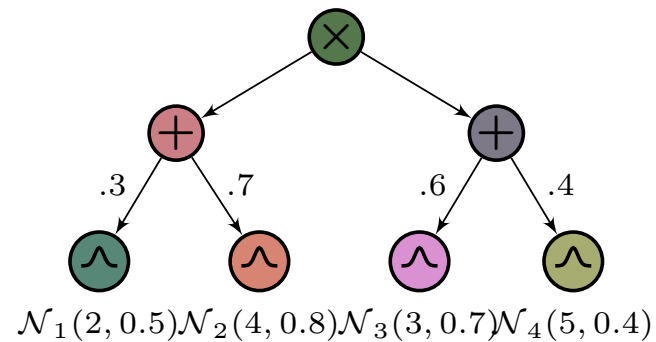
Definition 2 (Determinism).
At most one sum node child has a positive value.



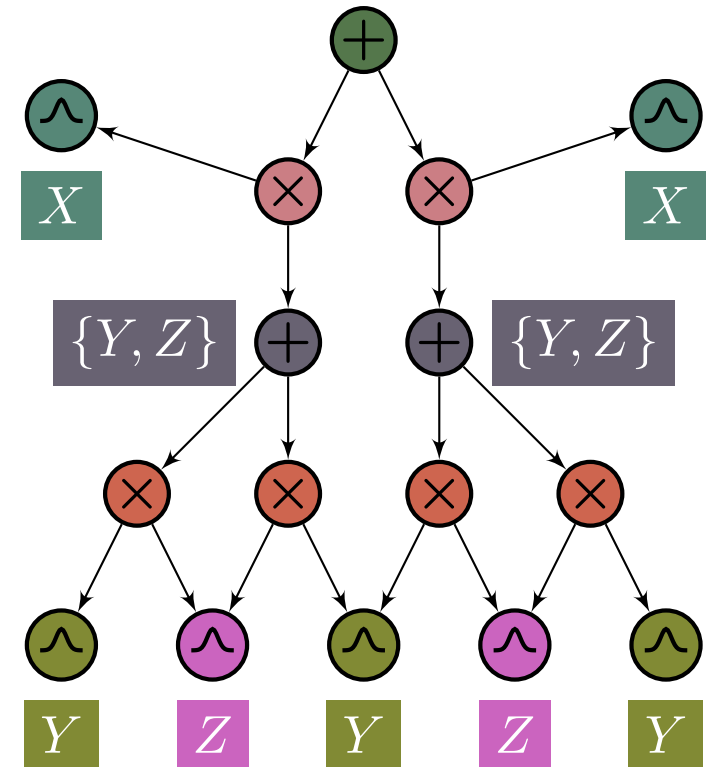
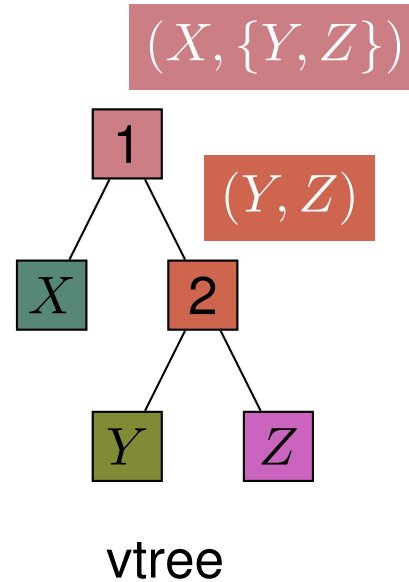
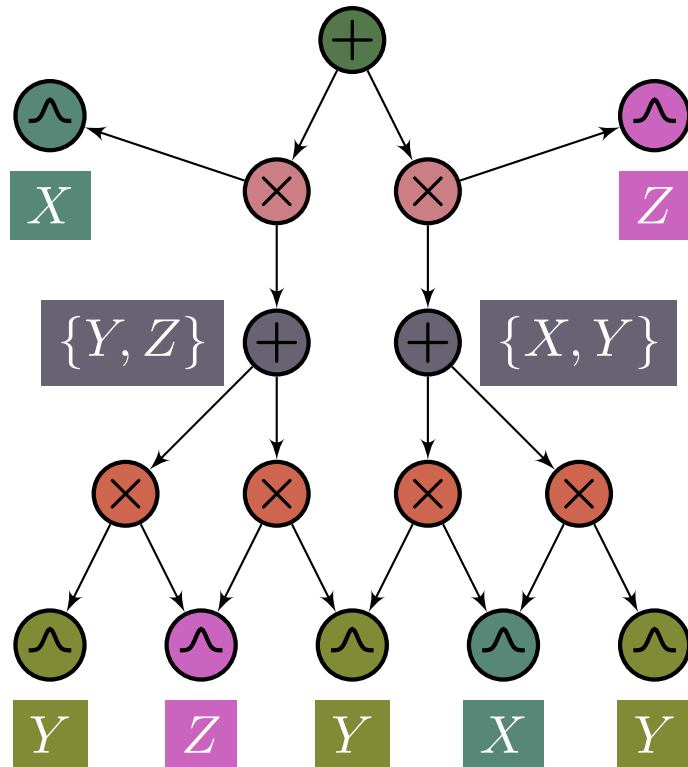
Probabilistic Circuits – Decomposability



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.*

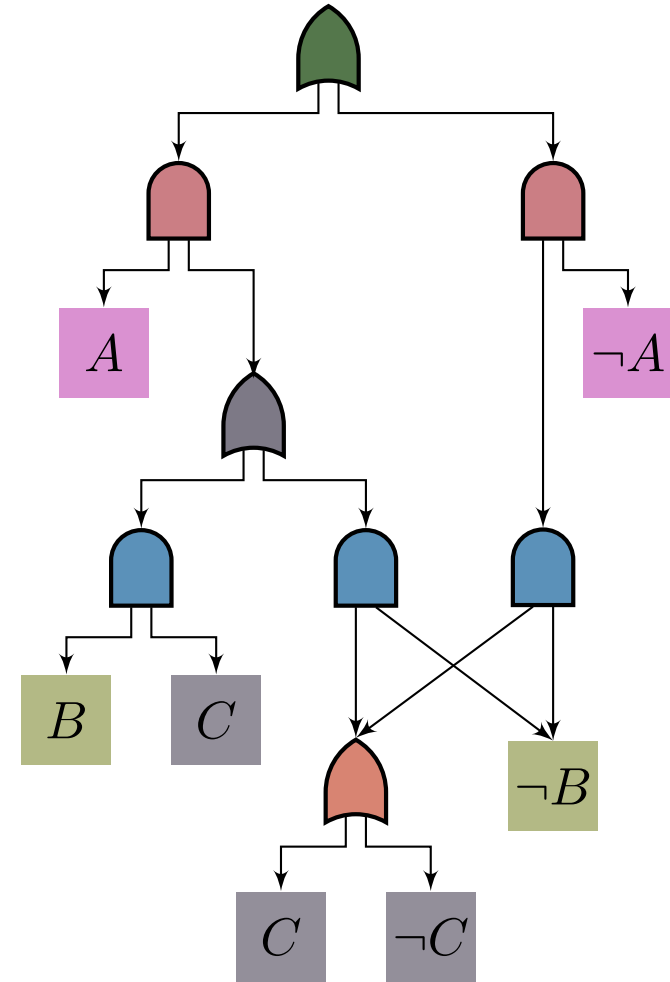
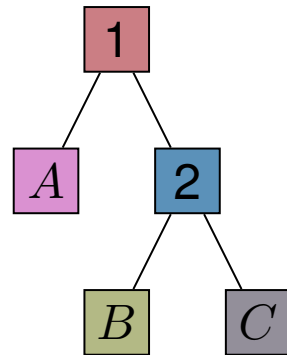
Probabilistic Circuits – Tractability

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

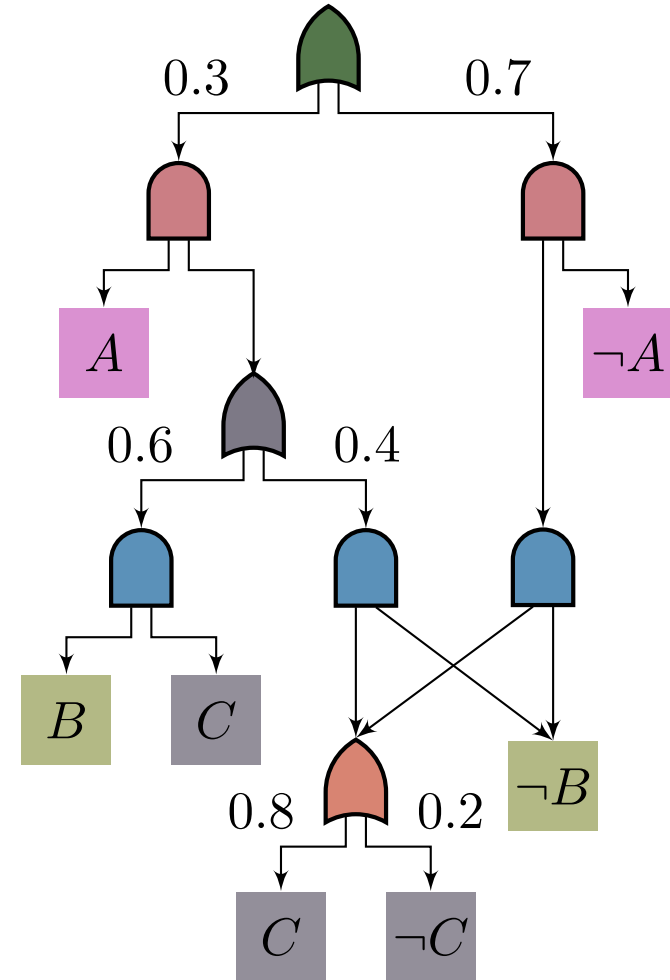
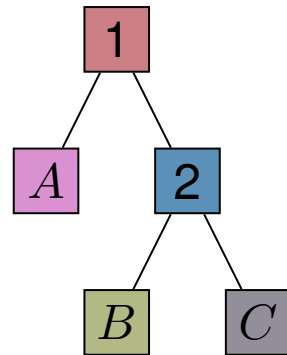
$$\phi(A, B, C) = (A \vee B) \wedge (\neg B \vee C)$$



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)

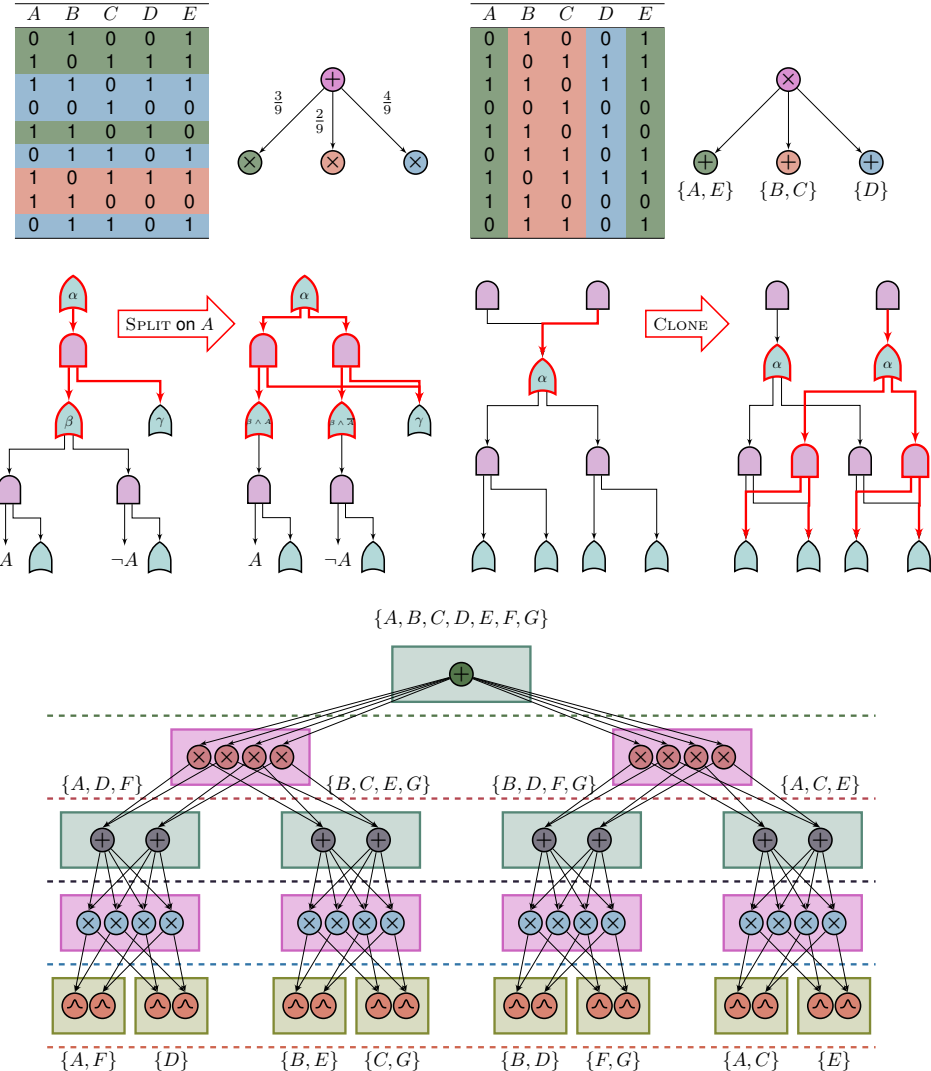
- Usually recursive;
- Splits data by similarity and stat dep;
- Stat dep usually costly;
- Usually tree-shaped.

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

Divide-and-Conquer Approaches (DIV)

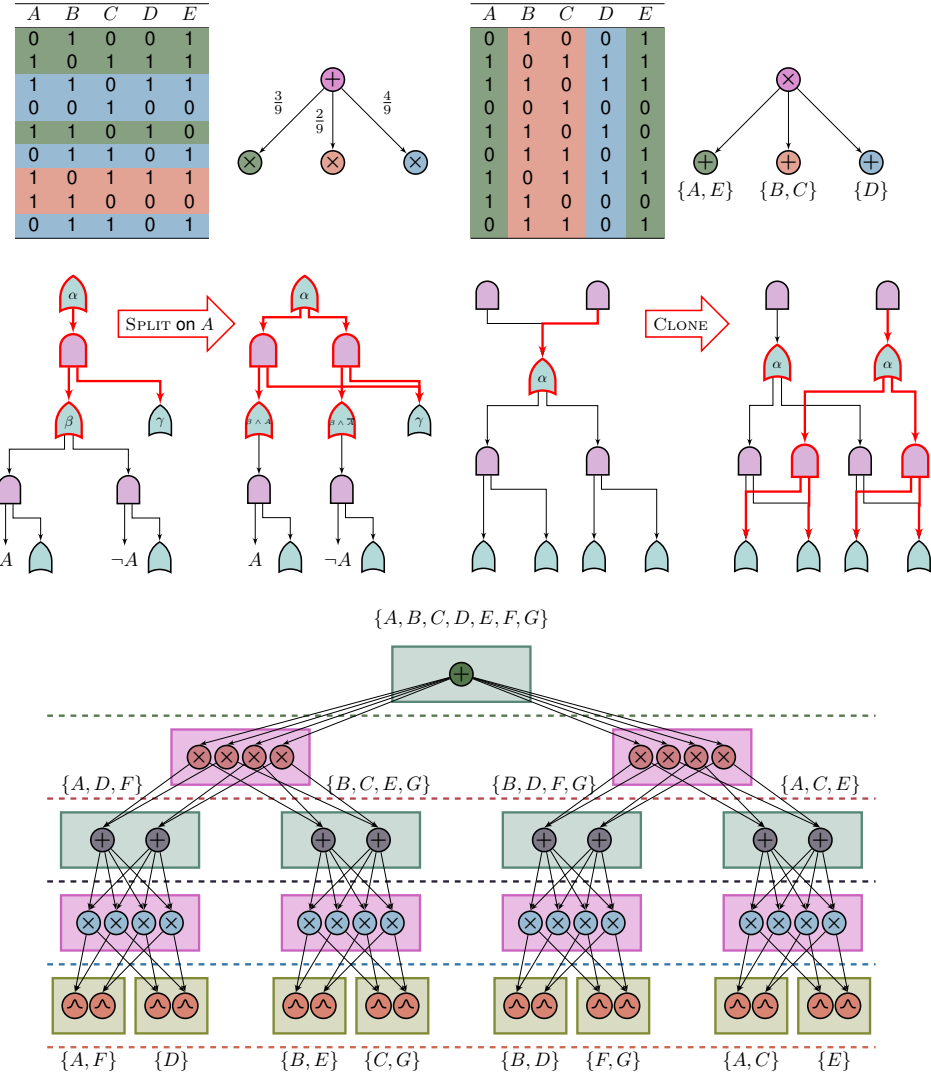
- Usually recursive;
- Splits data by similarity and stat dep;
- Stat dep usually costly;
- Usually tree-shaped.

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

Divide-and-Conquer Approaches (DIV)

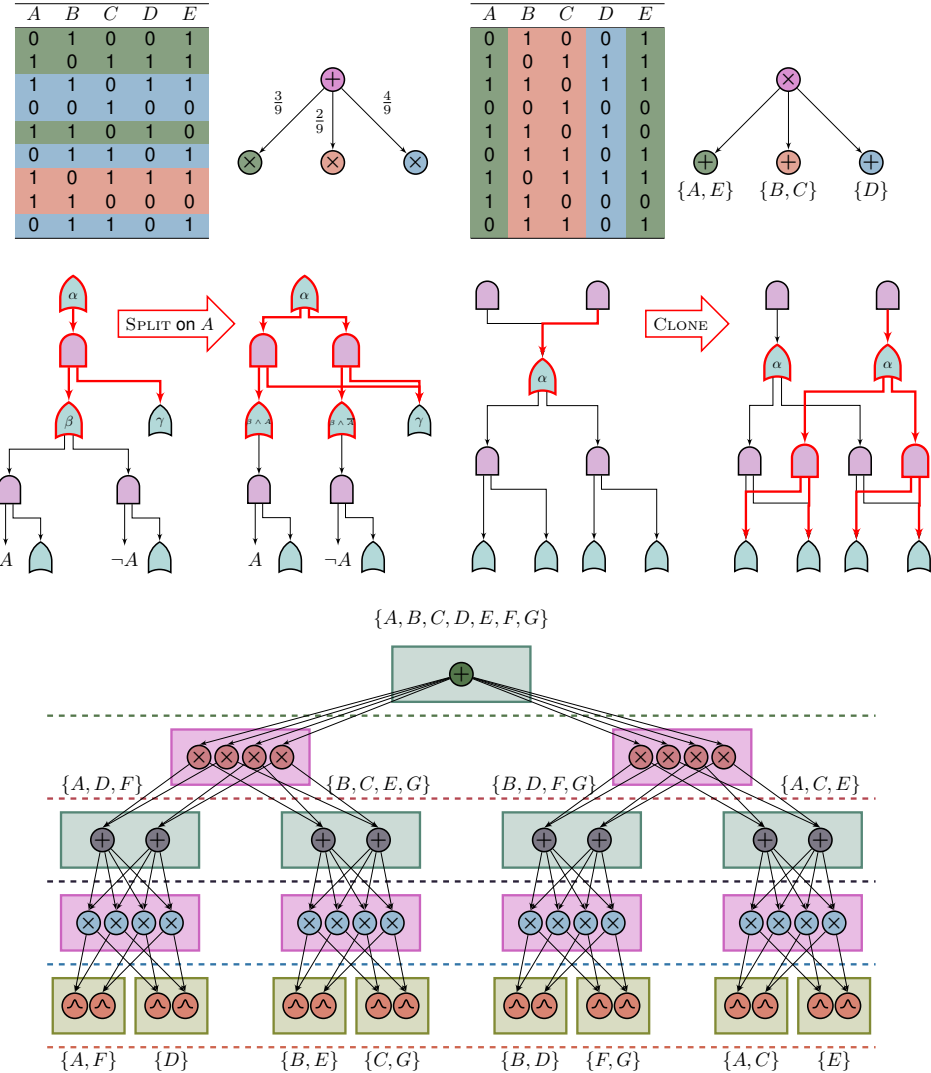
- Usually recursive;
- Splits data by similarity and stat dep;
- Stat dep usually costly;
- Usually tree-shaped.

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?

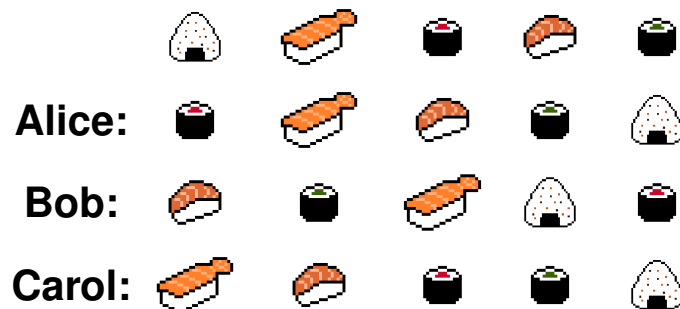
Name	Class	Time Complexity	# hyperparams	Accepts logic?	Sm?	Dec?	Det?	Str Dec?	$\{0,1\}$?	\mathbb{N} ?	\mathbb{R} ?	Reference
LEARNSPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & , \text{ if sum} \\ \mathcal{O}(nm^3) & , \text{ if product} \end{cases}$	≥ 2	✗	✓	✓	✗	✗	✓	✓	✓	Gens and Domingos [2013]
ID-SPN	DIV	$\begin{cases} \mathcal{O}(nkmc) & , \text{ if sum} \\ \mathcal{O}(nm^3) & , \text{ if product} \\ \mathcal{O}(ic(rn + m)) & , \text{ if input} \end{cases}$	$\geq 2 + 3$	✗	✓	✓	✗	✗	✓	✓	✗	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]
LEARNSDD	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]
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	\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	$\begin{cases} \mathcal{O}(ic(rn + m)) & , \text{ if input} \\ \mathcal{O}(nkmc) & , \text{ if sum} \\ \mathcal{O}(m(\log m)^2) & , \text{ if product} \end{cases}$	≥ 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]):

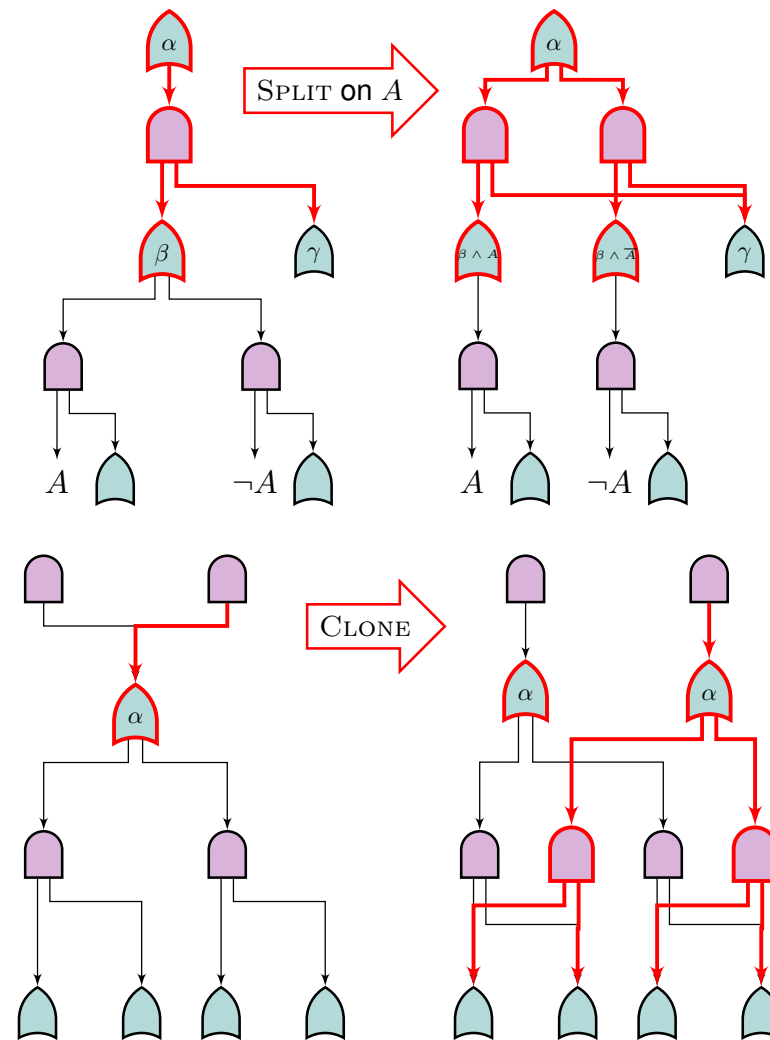
- ✗ Requires initial logic circuit encoding the support...
- ✗ Scales poorly to complex formulae and/or high dimension...
- ✗ Costly whole circuit evaluation at every iteration...
- ✓ Very good performance!

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]):

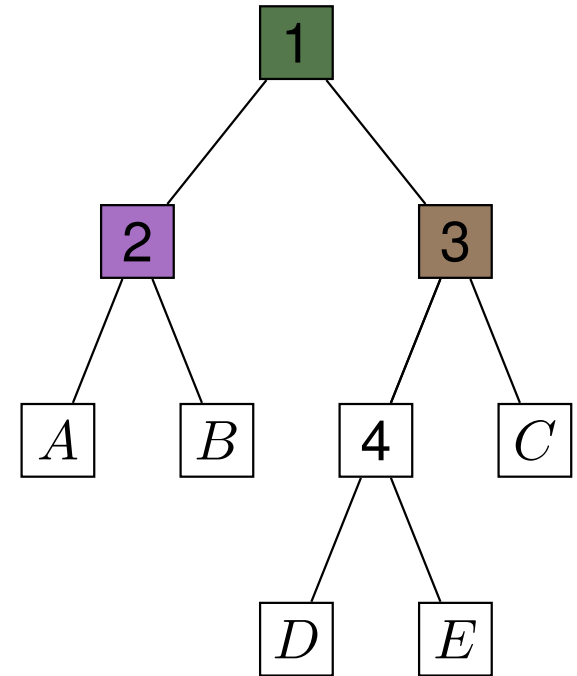
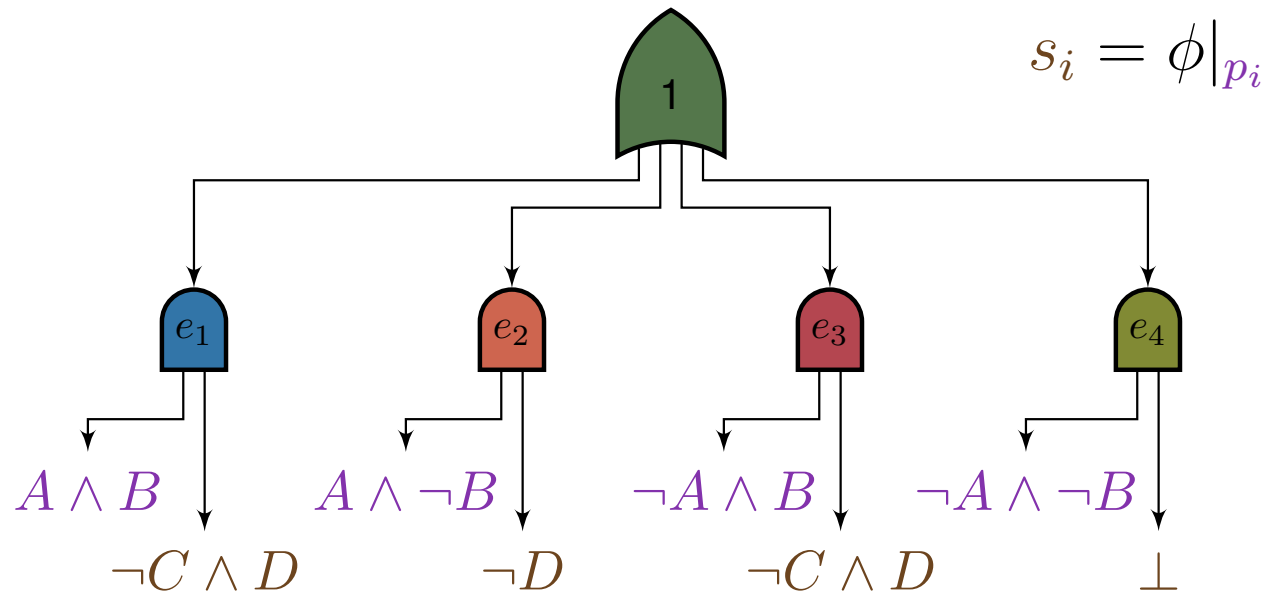
- ✓ Scales to high dimension and complex formulae!
- ✓ Constructs a structure consistent with constraints!
- ✗ But does so by relaxing the formula...
- ✗ Performance varies on set bounds and vtree structure...



SAMPLEPSDD

Common assumption: p_i are conjunctions of literals.

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

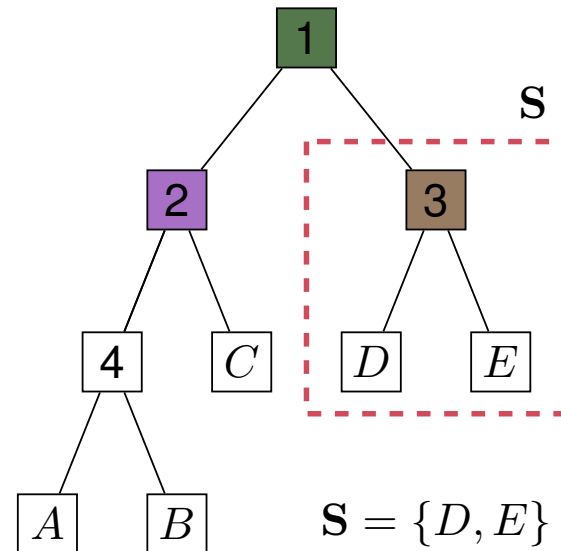
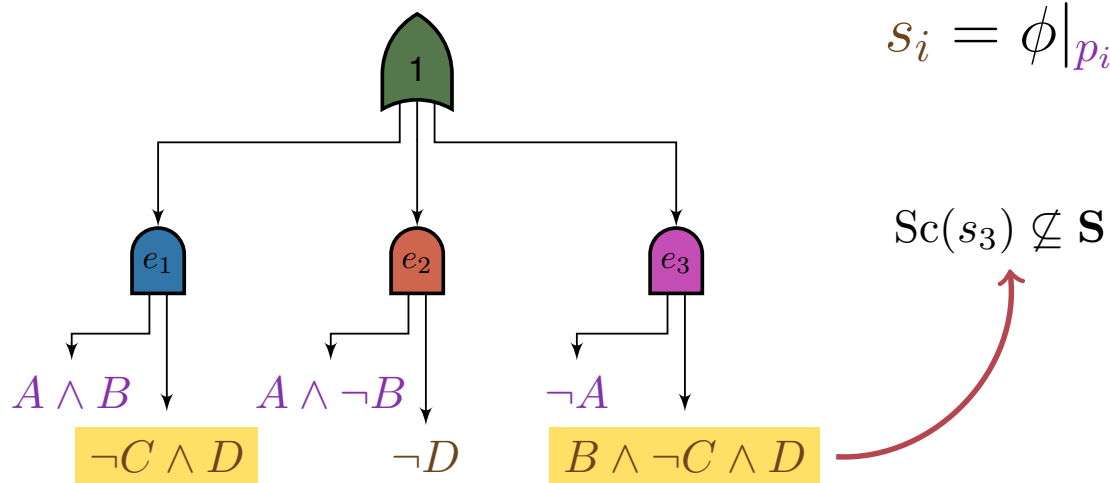


Problem: size of circuit is **exponential** in the size of p_i 's scope.

SAMPLEPSDD

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

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



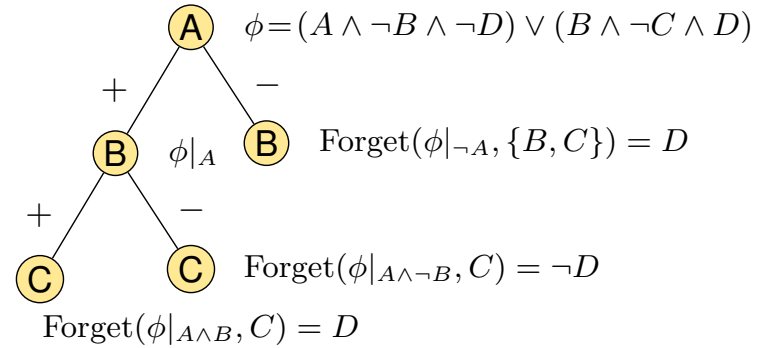
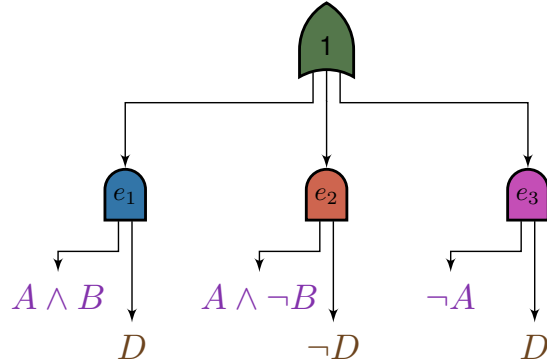
But: this violates structured decomposability:

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

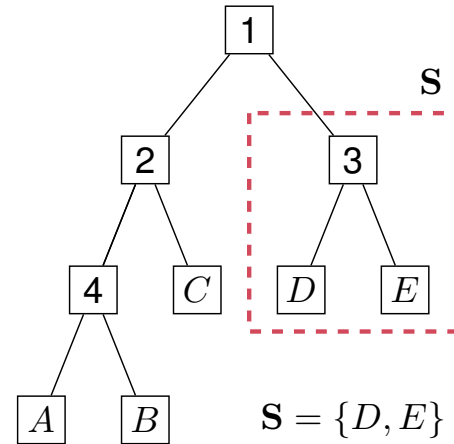
$\neg B \wedge \neg C \wedge D$ contains B and C , and $B, C \notin S$

SAMPLEPSDD

New solution: relax logical constraints ϕ

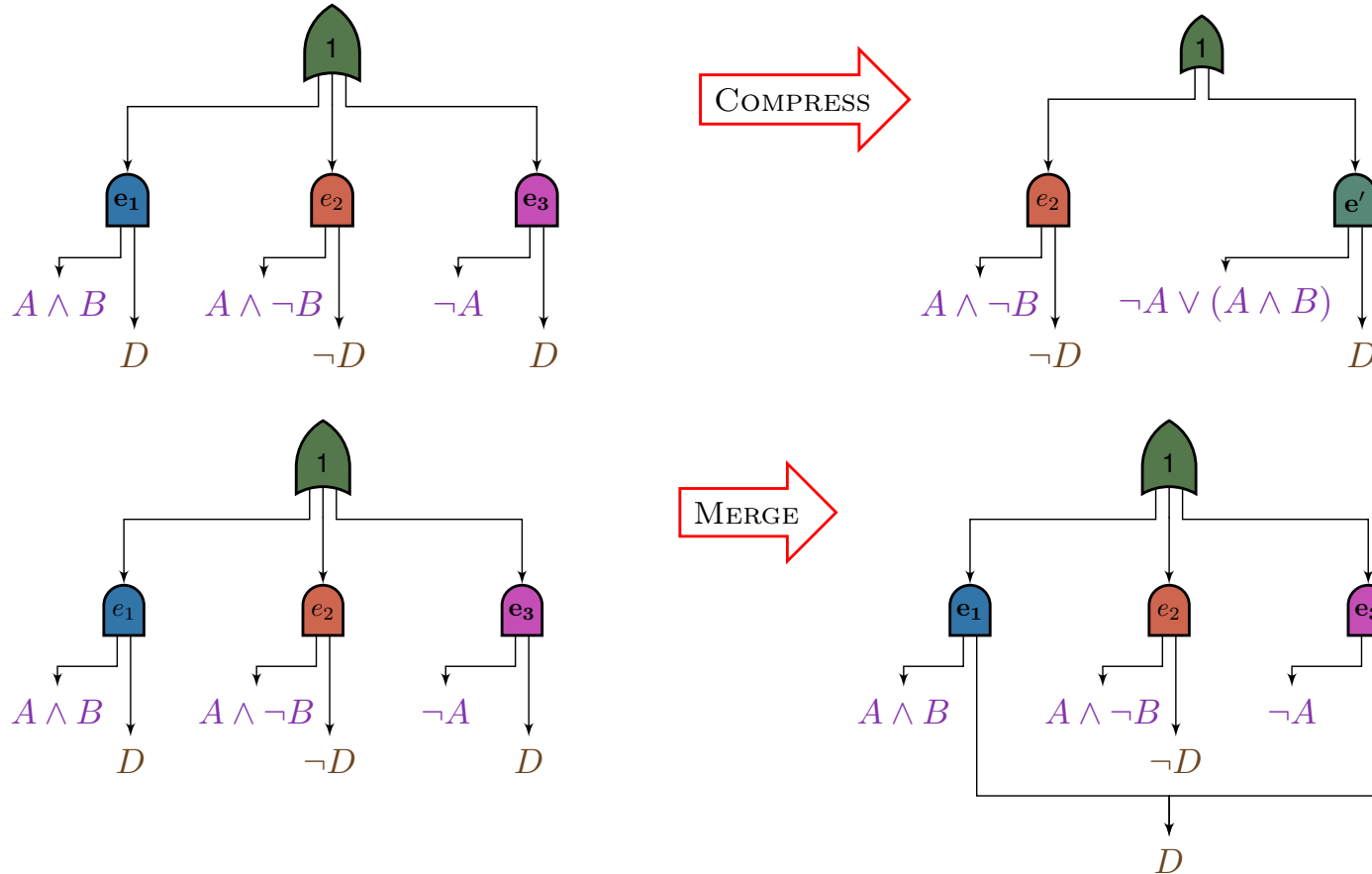


Now all s_i respect S



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);

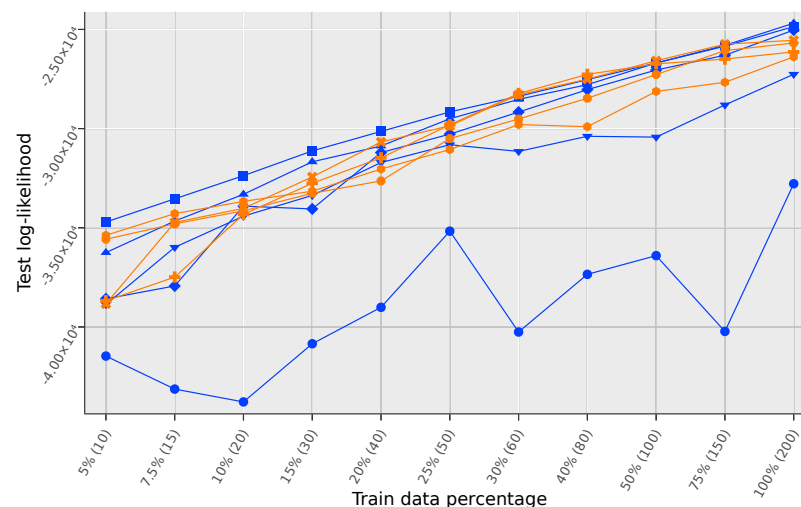
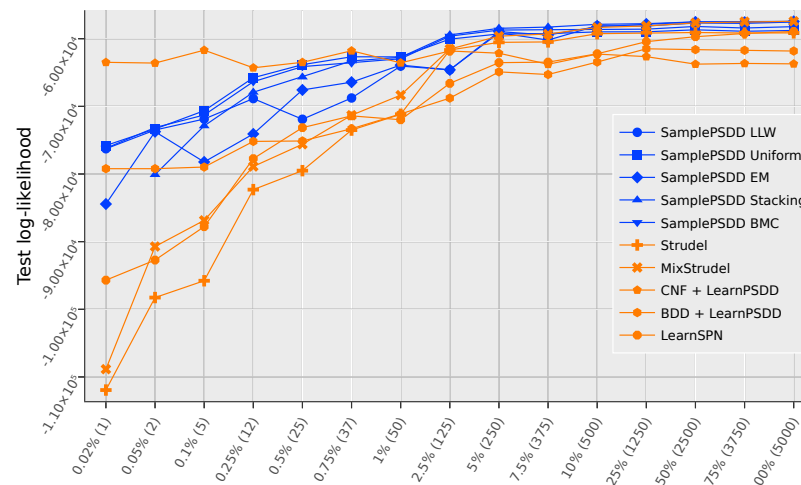
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

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

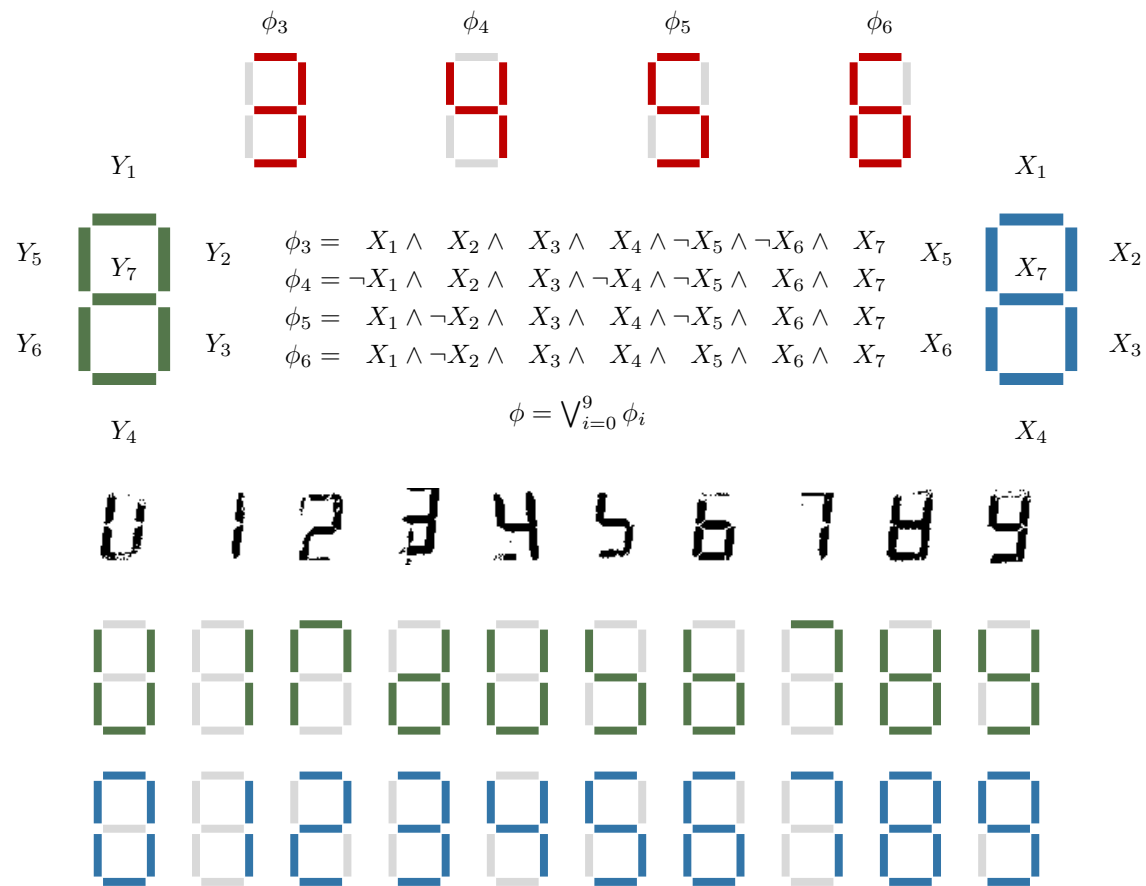
	Dataset	#vars	#train	ϕ 's size
⇒	LED	14	5000	23
⇒	LED + IMAGES	157	700	39899
	SUSHI RANKING	100	3500	17413
	SUSHI TOP 5	10	3500	37
	DOTA 2 GAMES	227	92650	1308

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);

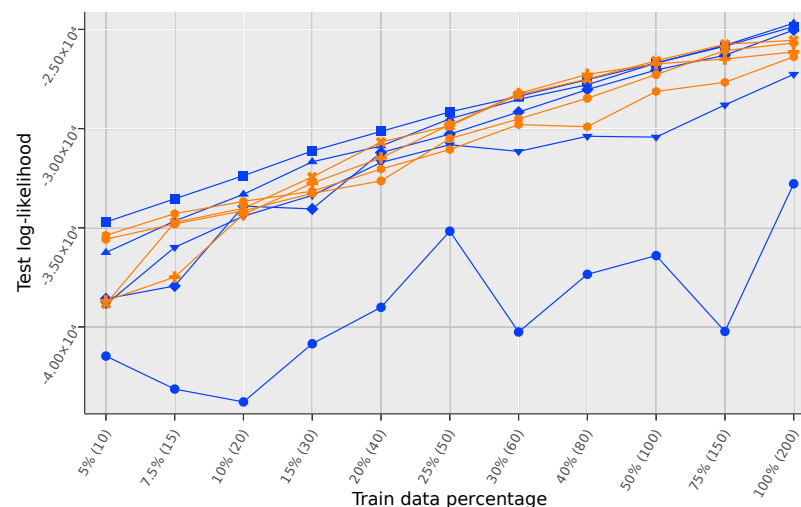
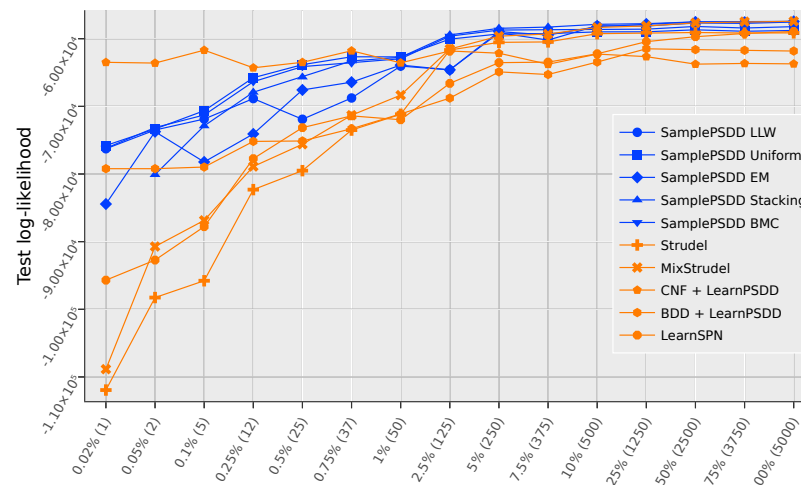
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

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

	Dataset	#vars	#train	ϕ '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);

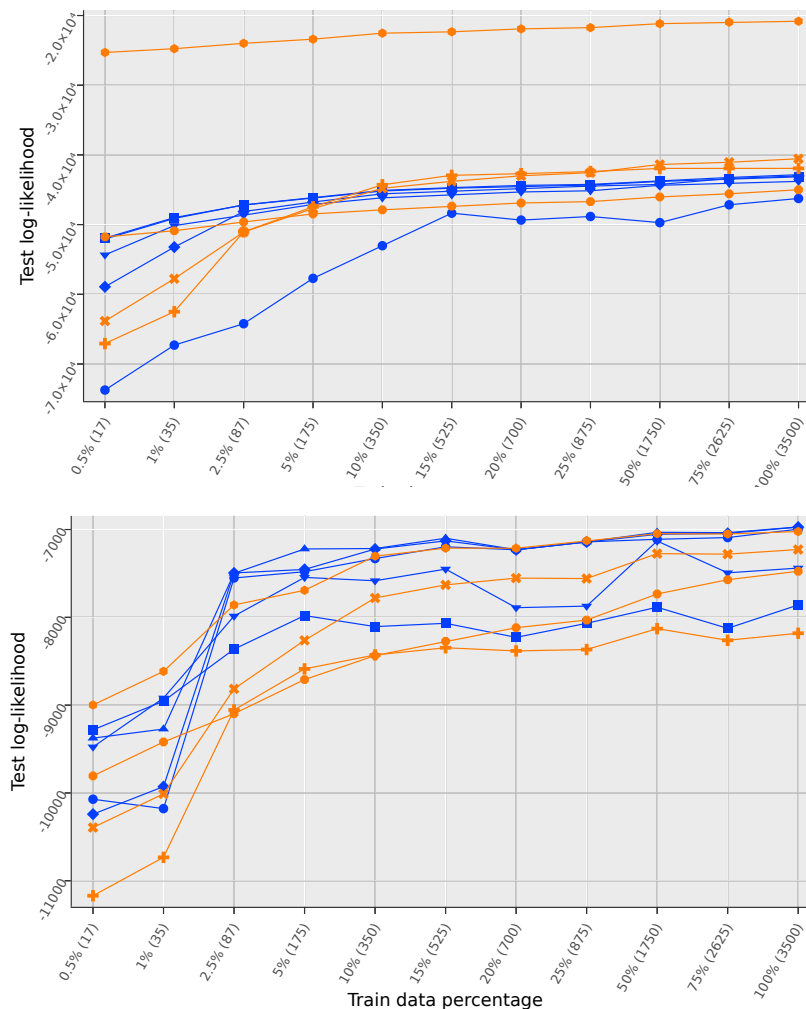
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

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

Dataset	#vars	#train	ϕ '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



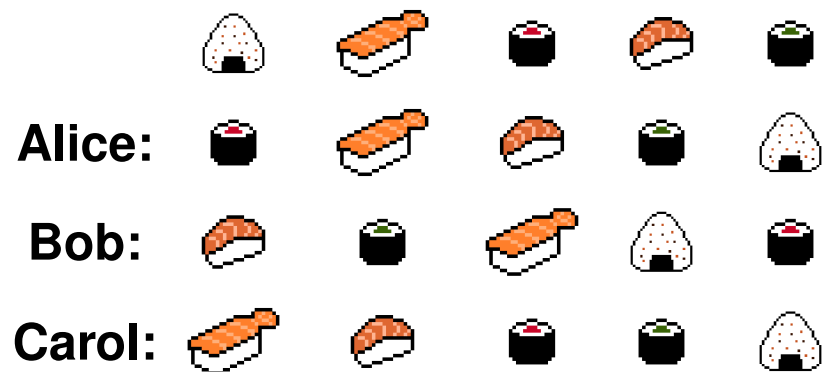
n sushi types and k rank positions

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

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

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

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);

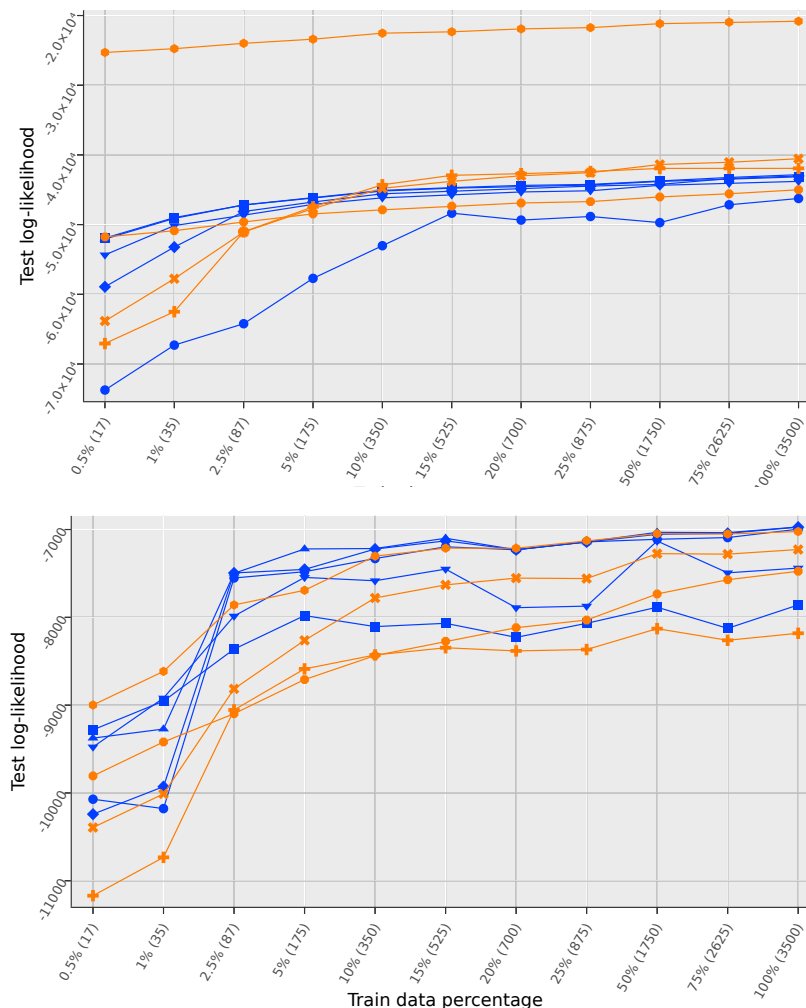
comparing against **STRUDEL**, **LEARNPSDD** and **LEARNSPN**.

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

Dataset	#vars	#train	ϕ '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);

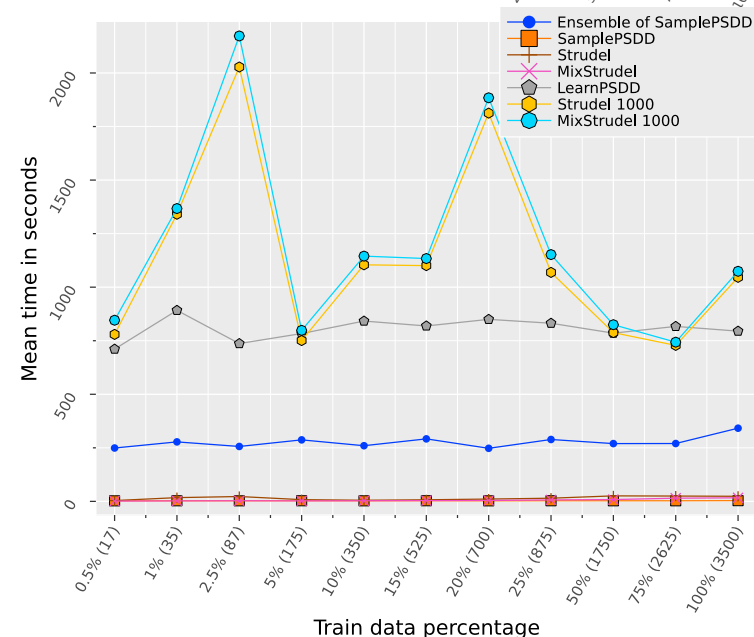
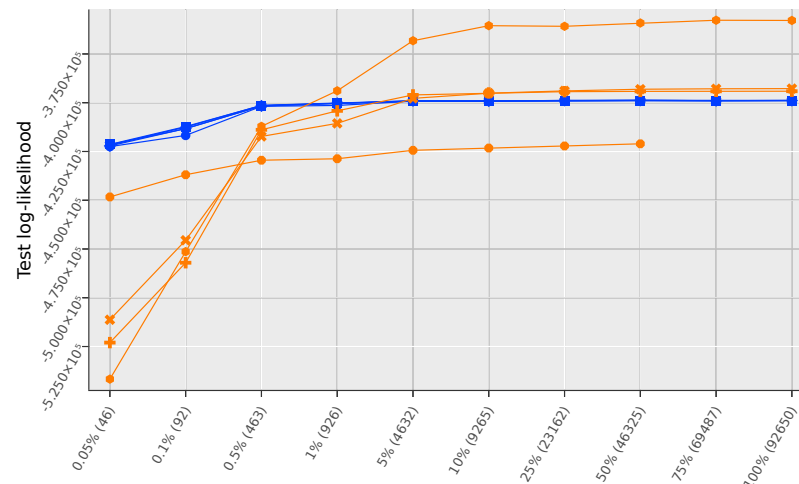
comparing against **STRUDEL**, **LEARNPSTD** and **LEARNSPN**.

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
⇒ 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);

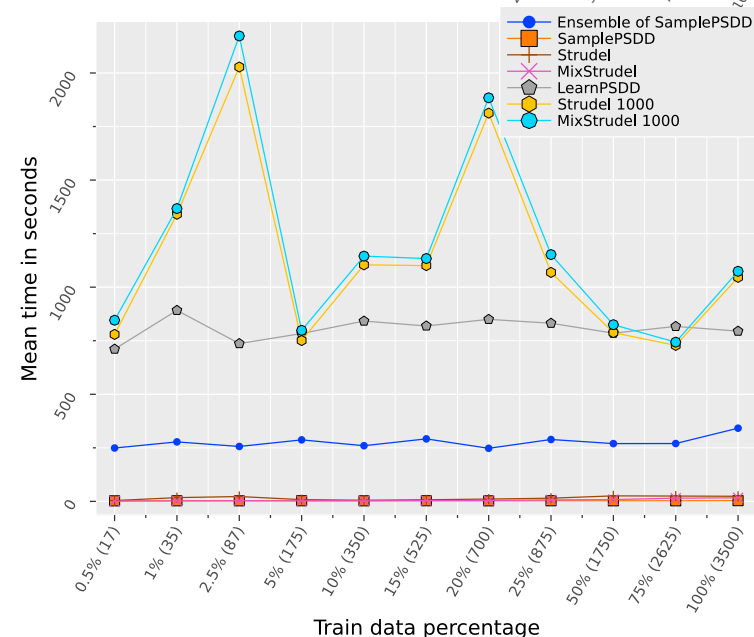
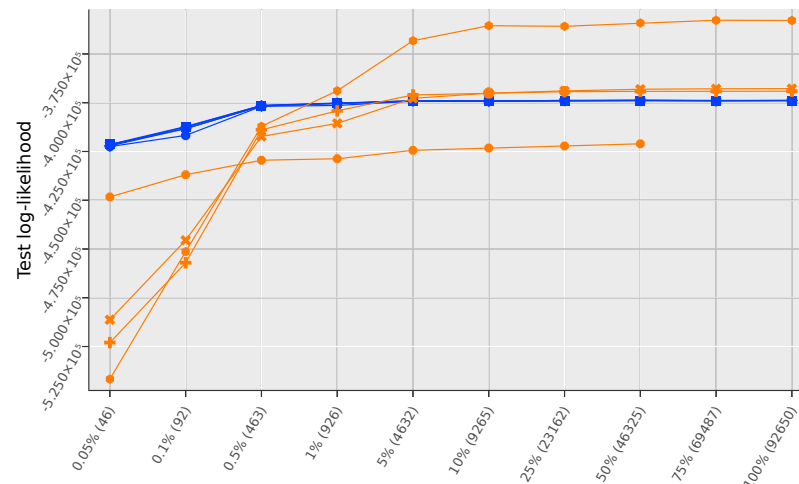
comparing against **STRUDEL**, **LEARNPSTD** and **LEARNSPN**.

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
⇒ DOTA 2 GAMES	227	92650	1308

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

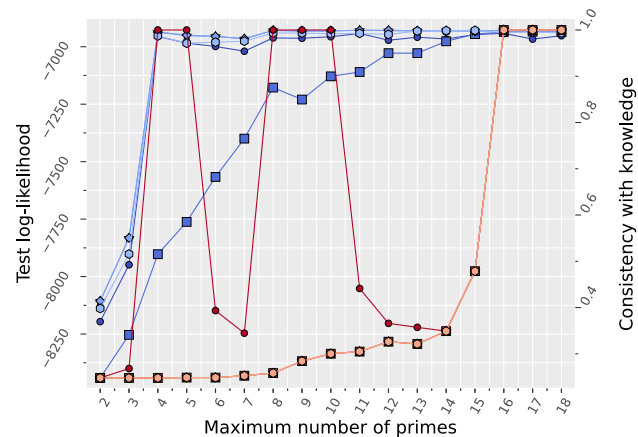
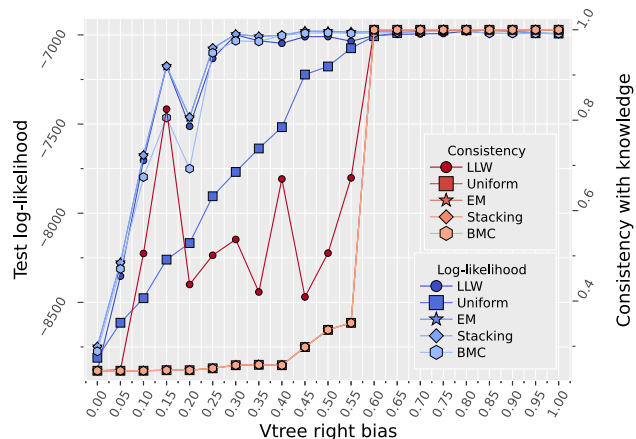
Mattei et al. [2020], Kamishima [2003], Shen et al. [2017],
Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]



SAMPLEPSDD – Experiments

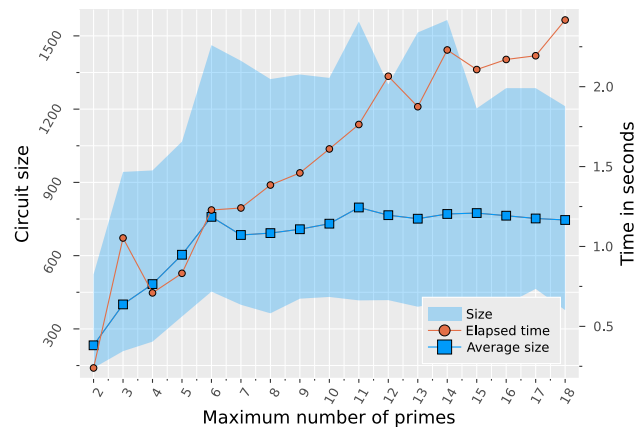
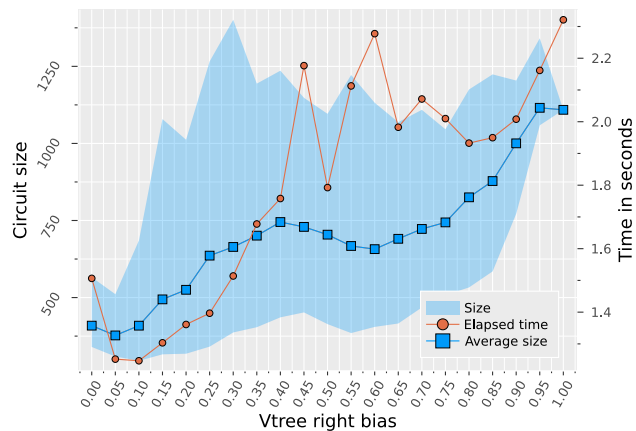
What is the impact of higher k 's and right-leaning vtrees

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]
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

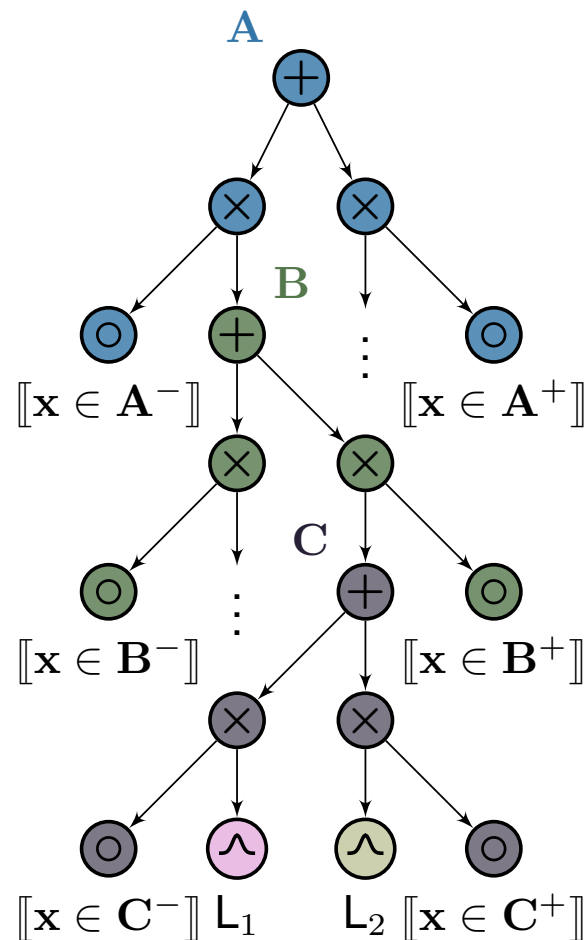
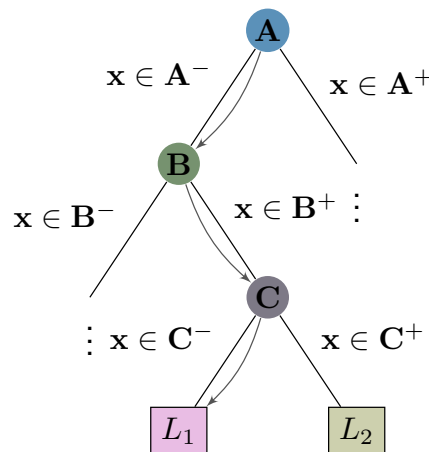
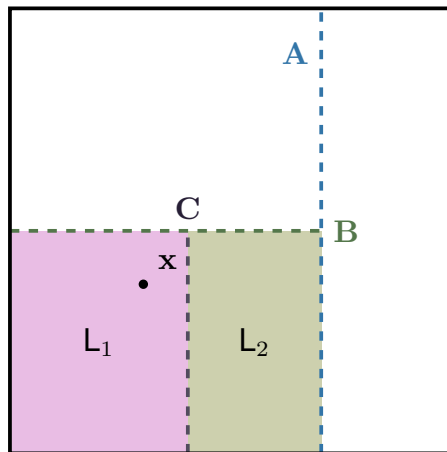
Density Estimation Trees...

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

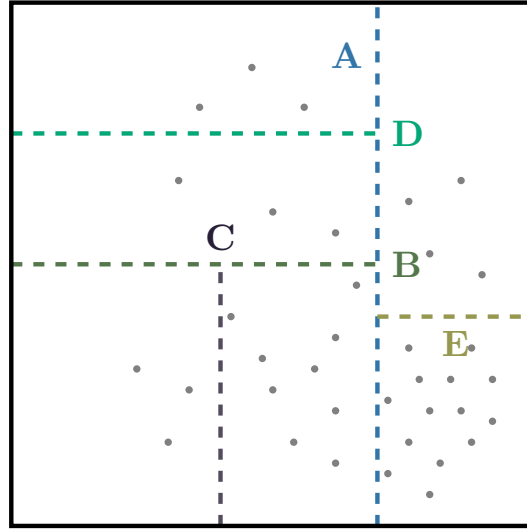
...but are subsumed by circuits!

Learn DETs \subseteq Learn PCs?

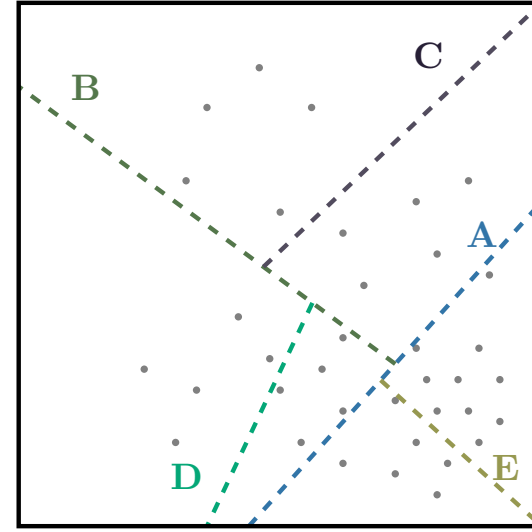
Can we take advantage of known learning procedures in DETs and transplant them to more general circuits?



Random Projections



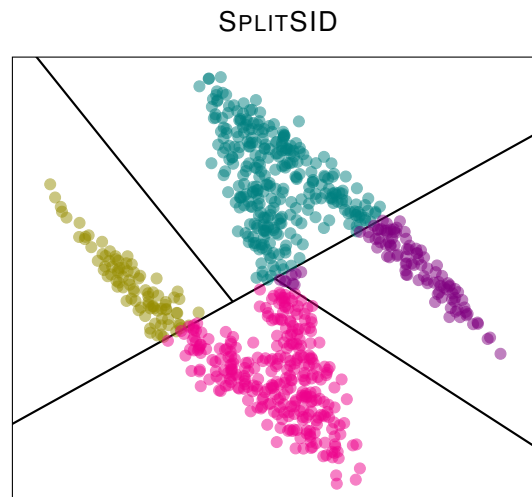
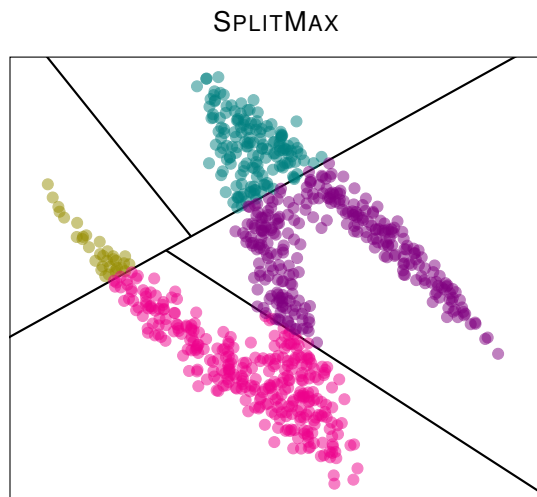
Axis-aligned projections



Random projections

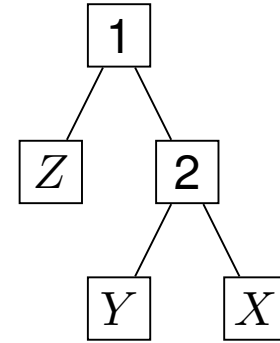
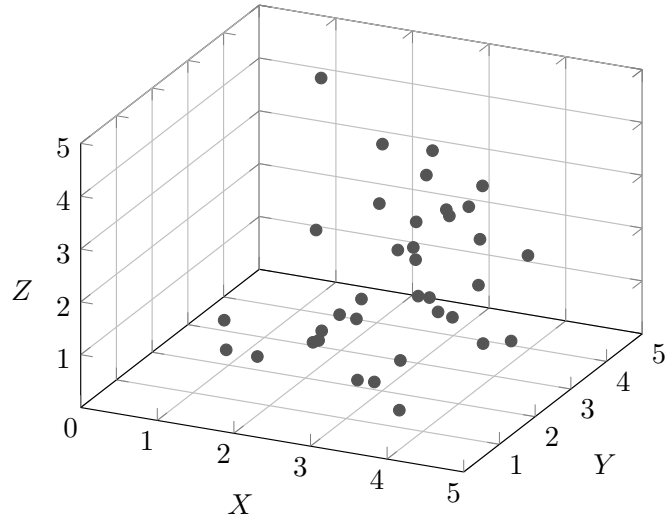
If the data has *intrinsic dimension* d , then with constant probability the part of the data at level d or higher of the tree has average diameter less than half of the data.

Random Projections

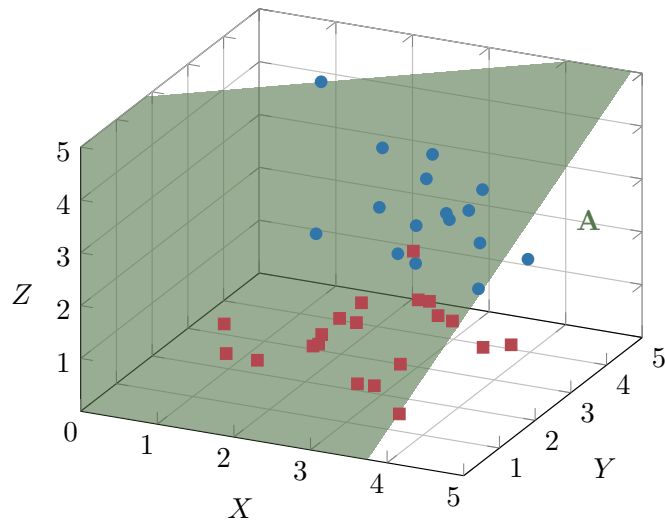


If the data has *intrinsic dimension* d , then with constant probability the part of the data at level d or higher of the tree has average diameter less than half of the data.

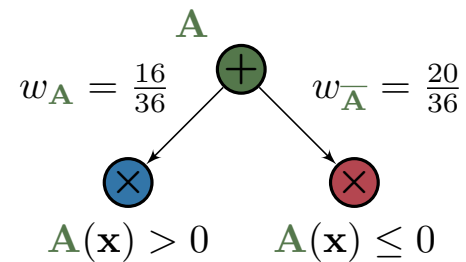
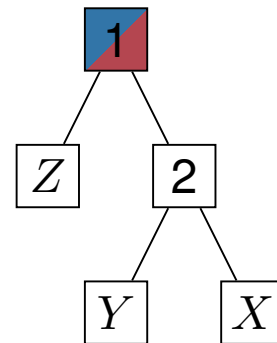
LearnRP



LearnRP

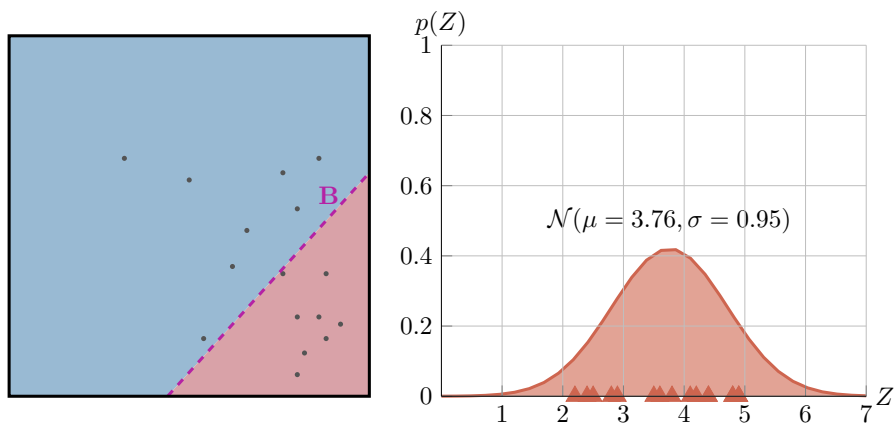
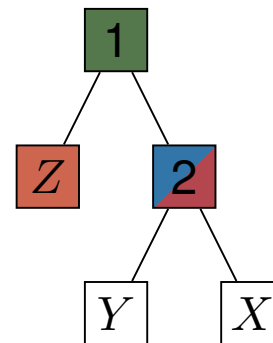
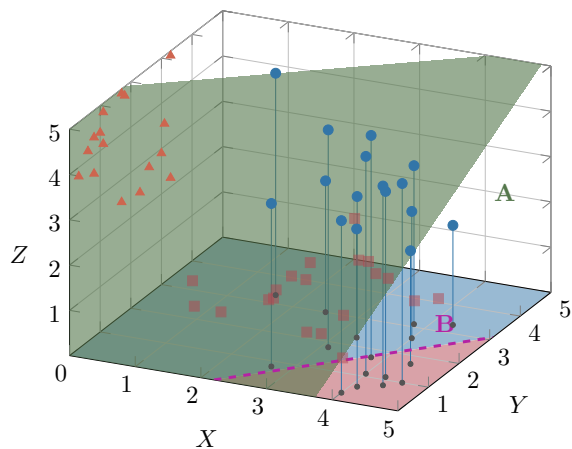


$$\mathbf{A}(x, y, z) = \underbrace{\begin{bmatrix} x & y & z \end{bmatrix} \cdot \begin{bmatrix} -0.31 \\ -0.40 \\ 0.85 \end{bmatrix}}_a + \underbrace{1}_{\theta}$$

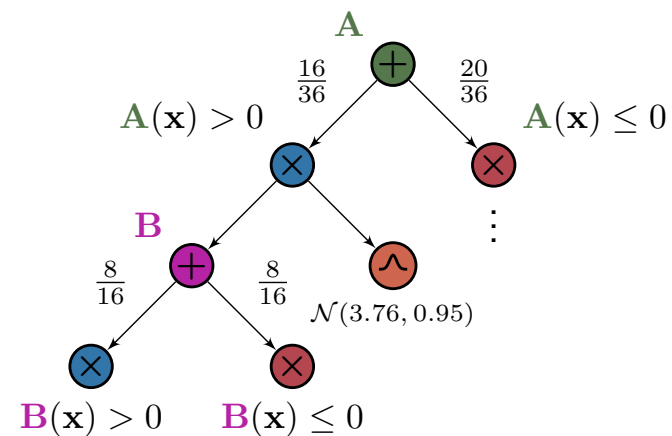


$w_{\mathbf{A}}$: probability of $\mathbf{A}(\mathbf{x}) > 0$

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})

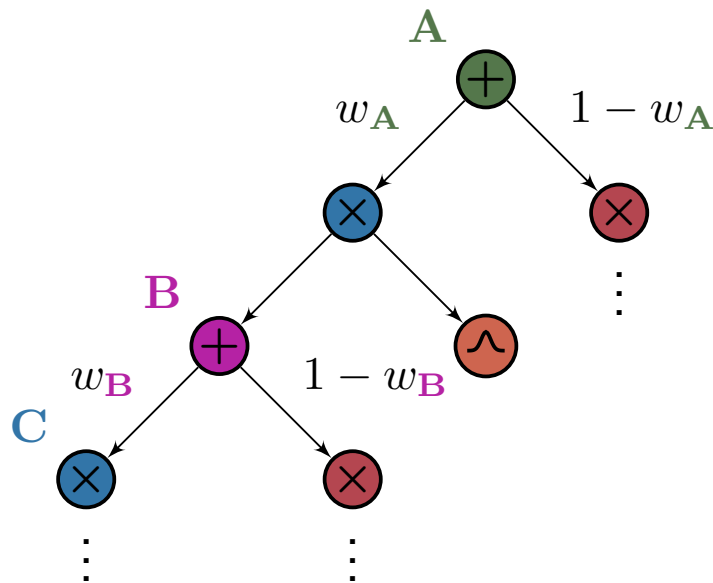
$$w_{\mathbf{B}} \propto w_{\mathbf{B}} \cdot \sum_{\mathbf{x} \in \mathbf{D}} \frac{1}{p_{\mathbf{A}}(\mathbf{x})} \cdot \frac{\partial p_{\mathbf{A}}(\mathbf{x})}{\partial p_{\mathbf{B}}(\mathbf{x})} \cdot p_{\mathbf{C}}(\mathbf{x})$$

- 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



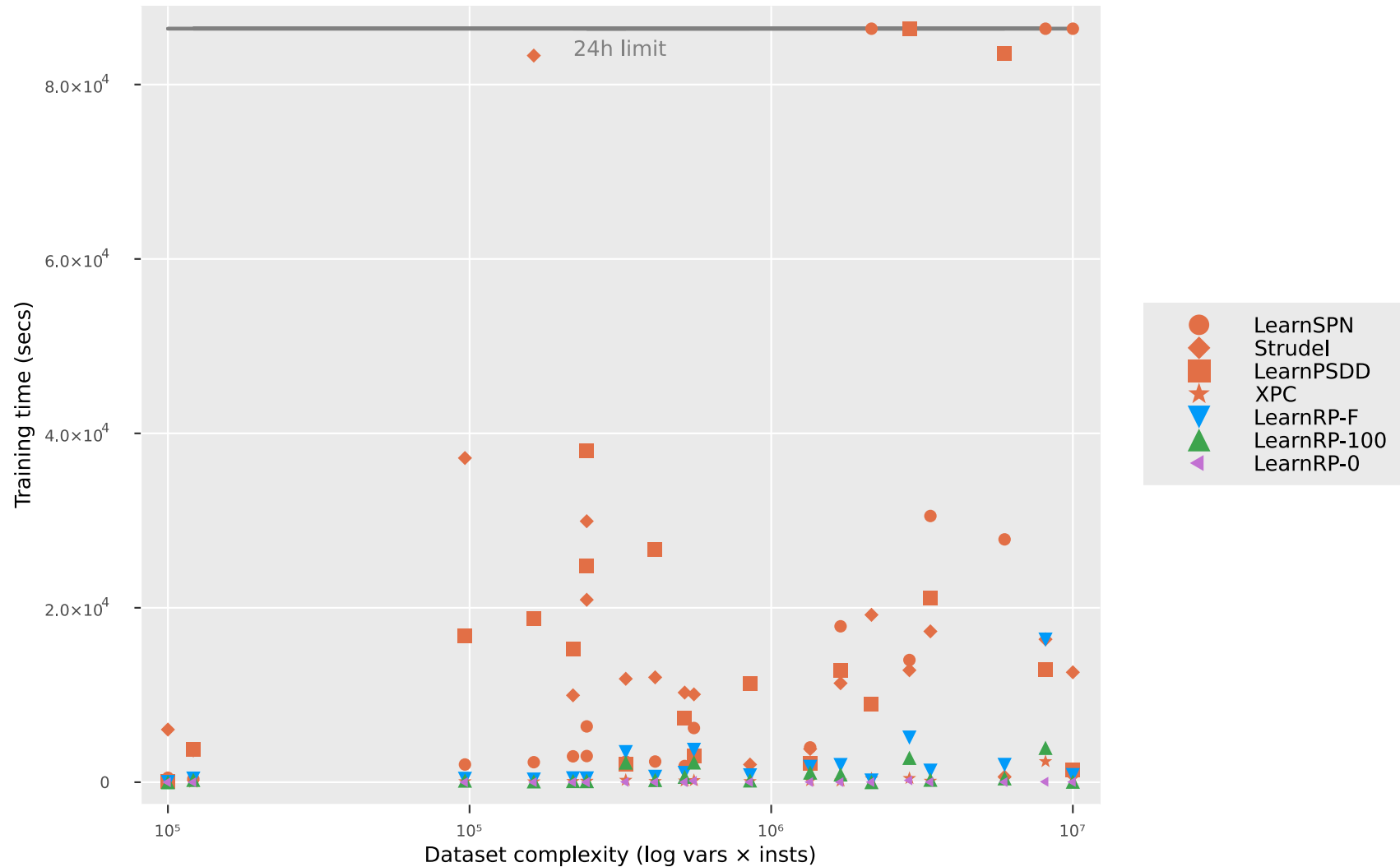
LEARNRP – Datasets

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}

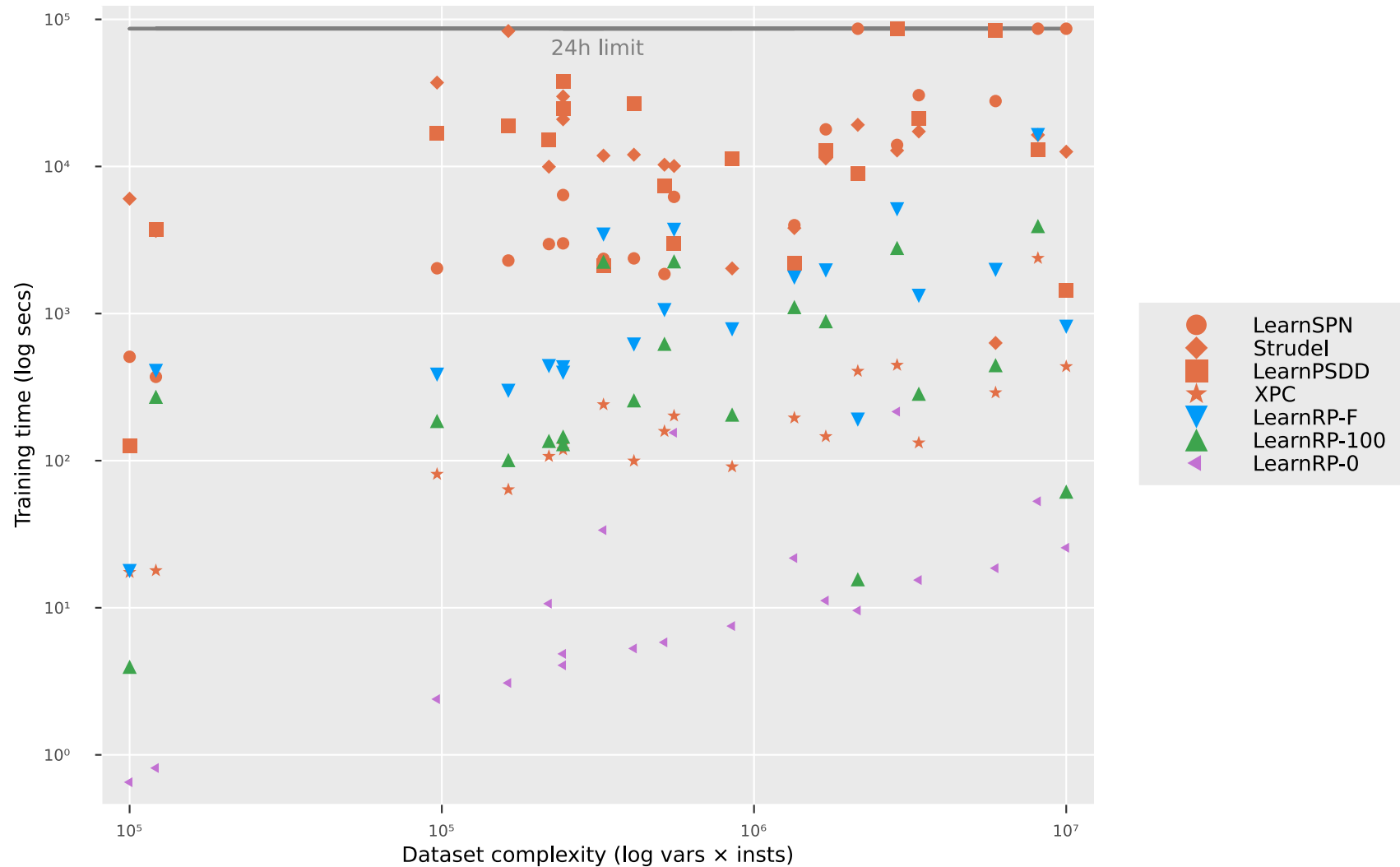
Experiments

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

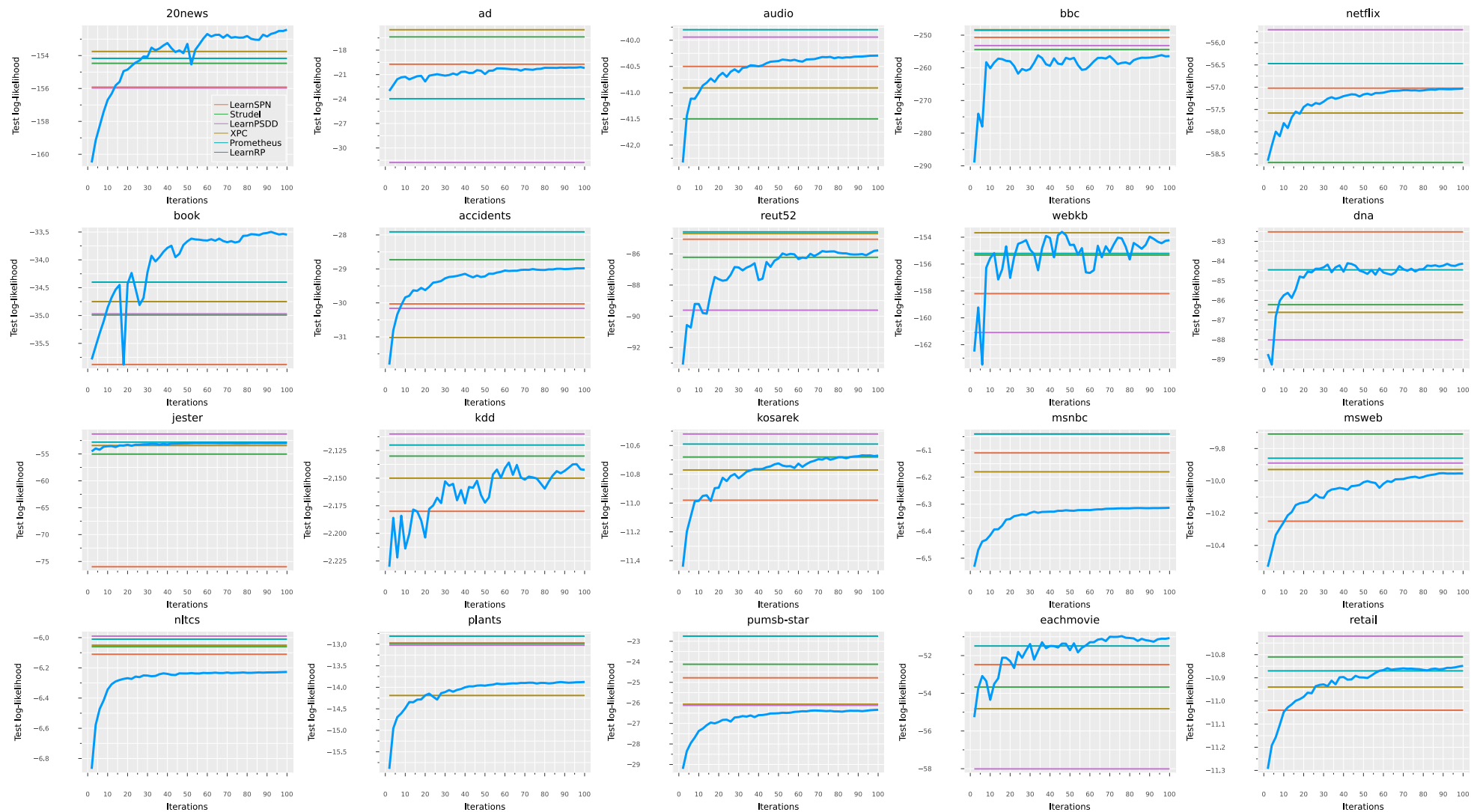
Experiments



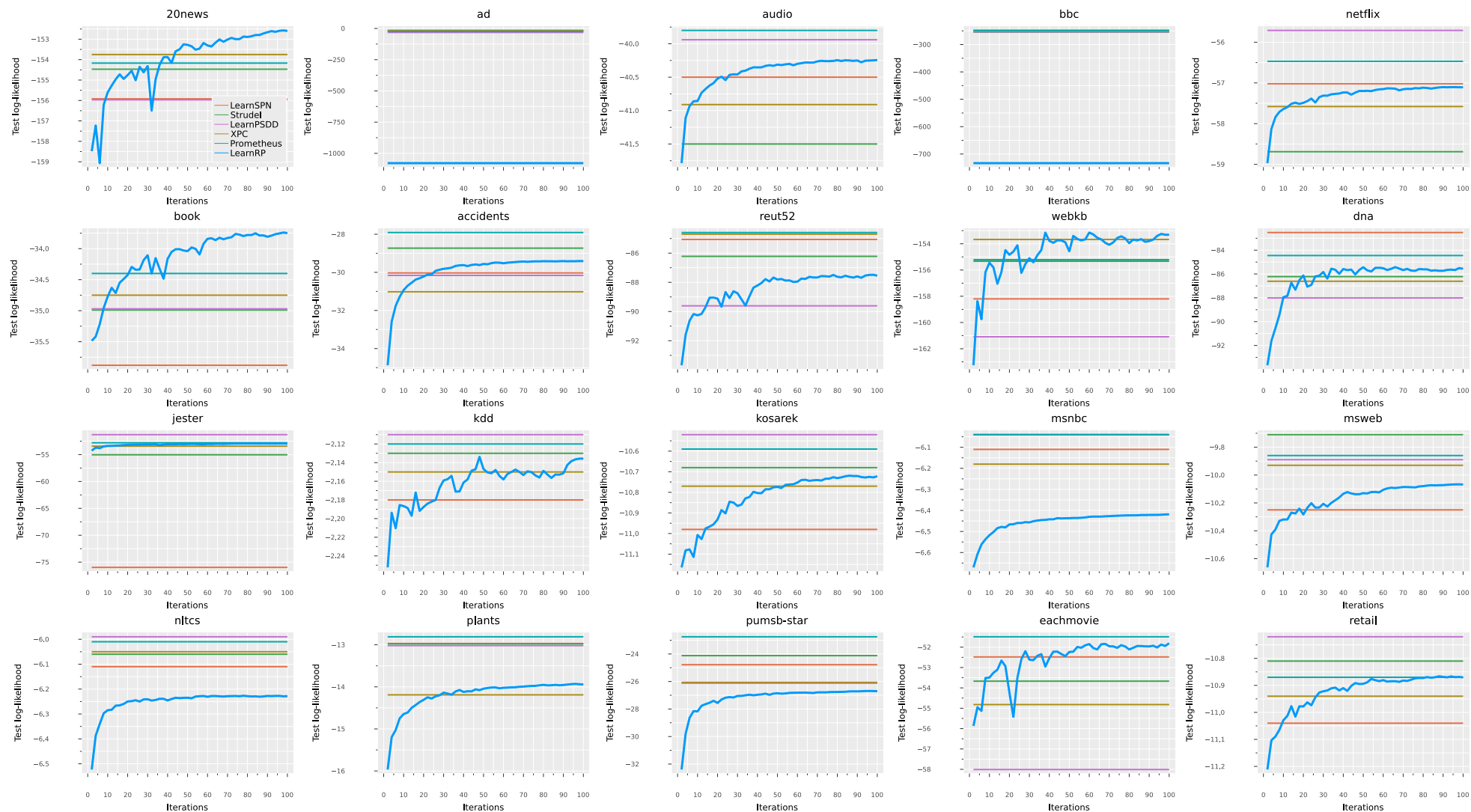
Experiments



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	—	-4.65	-0.85	—	-3.58	317
BANKNOTE	4	-2.76	-1.39	-4.64	—	-4.32	<u>-1.96</u>	—	-4.27	79
CA	22	-4.95	<u>21.19</u>	3.42	—	-7.33	27.82	—	9.48	2675
KINEMATICS	8	-5.55	-11.13	-11.20	—	-11.15	-11.12	—	<u>-10.16</u>	319
QUAKE	4	-2.38	-1.21	-3.76	—	-4.09	<u>-1.50</u>	—	-1.63	79
SENSORLESS	48	-26.91	<u>60.72</u>	8.56	—	-34.14	62.03	—	17.52	12650
CHEMDIABET	3	—	—	—	-3.02	-18.49	-2.59	<u>-2.88</u>	-19.06	47
FLOWSIZE	3	-0.79	<u>15.32</u>	5.72	—	2.27	18.03	—	2.83	49
OLDFAITH	2	—	—	—	-1.73	-4.18	-1.48	<u>-1.70</u>	-4.26	19
IRIS	4	—	—	—	-3.94	<u>-2.26</u>	-1.06	-3.74	-3.14	79

In conclusion

Contributions

Literature review

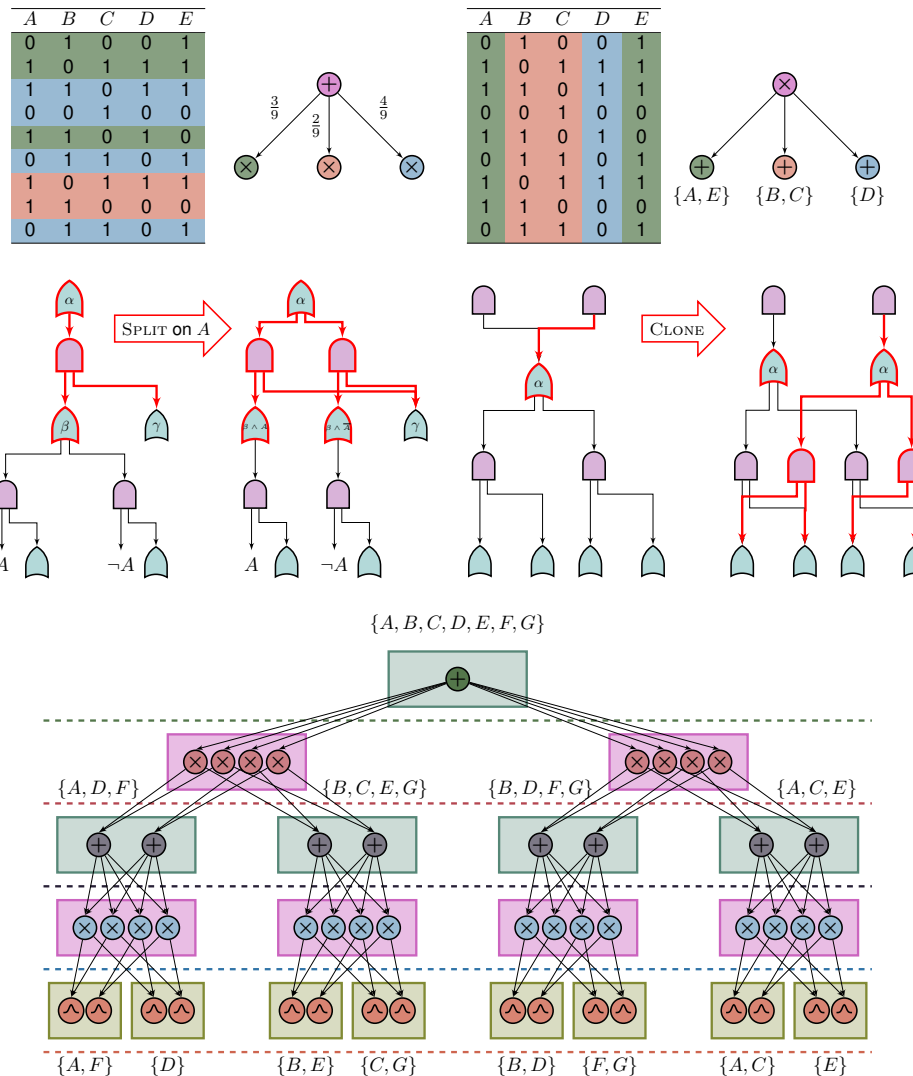
- Systematic review of literature;
- Taxonomy of popular algorithms;
- Complexity analysis;
- Pros and cons.

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

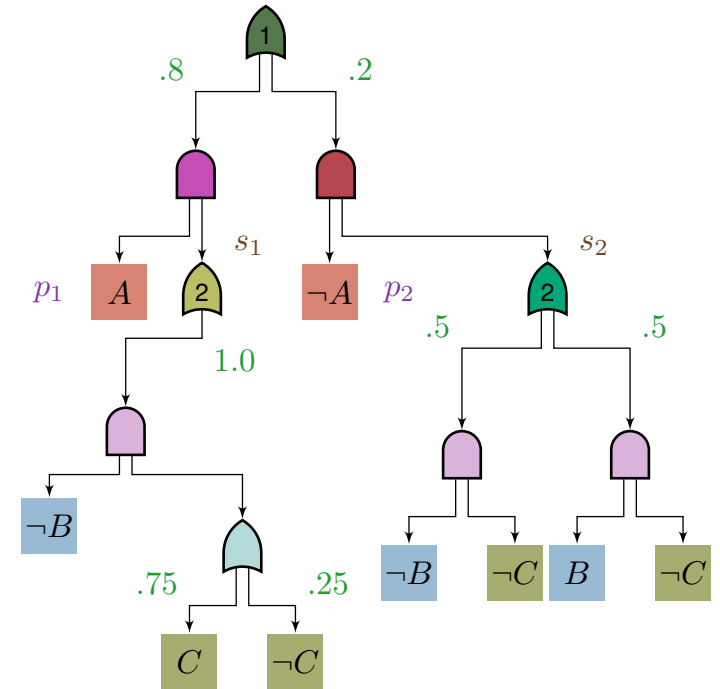
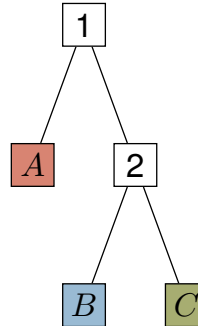
- Consistent with a relaxation of a formula;
- Relaxation as a function of vtree and sampling;
- Compromise between tractability and consistency;
- Ensembles mitigate relaxation.

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

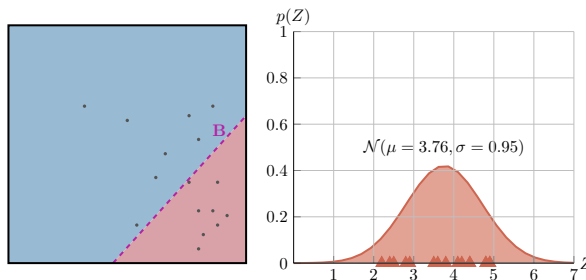
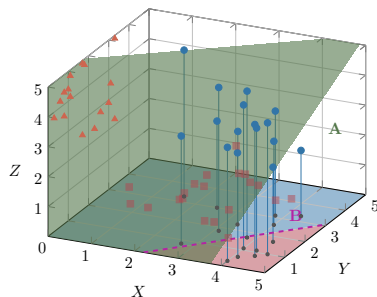
- Systematic review of literature;
- Taxonomy of popular algorithms;
- Complexity analysis;
- Pros and cons.

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.



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

