Scalable Learning of Probabilistic Circuits

Renato Lui Geh, Denis Deratani Mauá

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...

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

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

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

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

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

Marginals

Conditionals

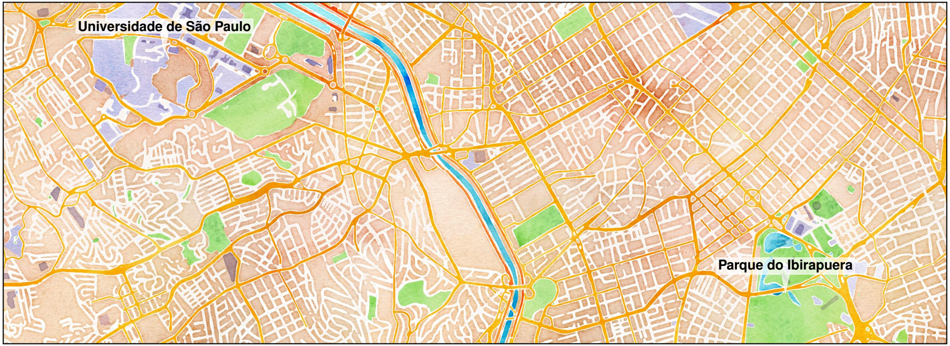
MPE

Logical events

...and extract meaningful queries from it?

Kamishima [2003]

Motivation



$\mathbf{W} \in \{\text{Sun, Mist, Light Rain}, \ldots, \text{Heavy Rain}\}$

$\mathbf{T}$: time of day

$\mathbf{D}$: day of the week

$\mathbf{S}$: streets of São Paulo

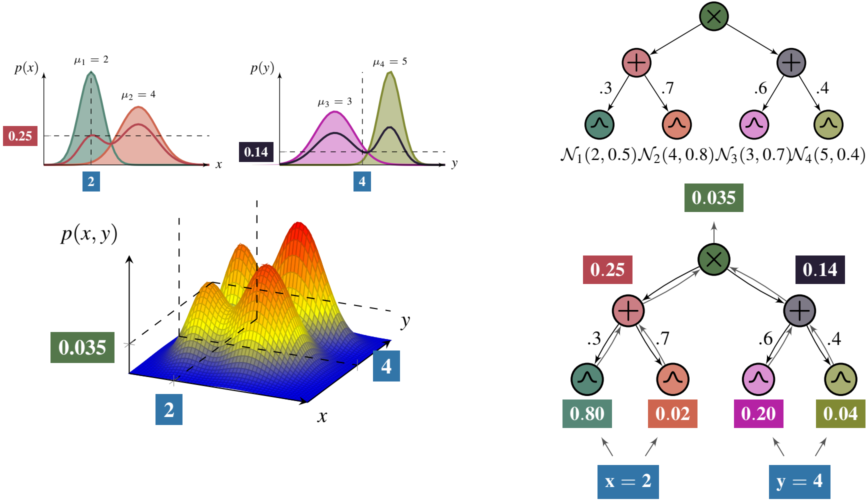
$\{J_{\mathbf{S}_i}\}_{i=1}^{|\mathbf{S}|}$: traffic jam at street $\mathbf{S}_i$

$\mathbf{H}$: holidays in São Paulo

$$\arg\max_{T \in \mathbf{T}} p(T \mid H = \text{Labor Day}, W = \text{Light Rain}, \bigvee_{S \in \text{Route}} J_S)$$

Choi et al. [2020]

Probabilistic Circuits



Querying in Probabilistic Circuits

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]

Learning Probabilistic Circuits

Divide-and-Conquer Approaches

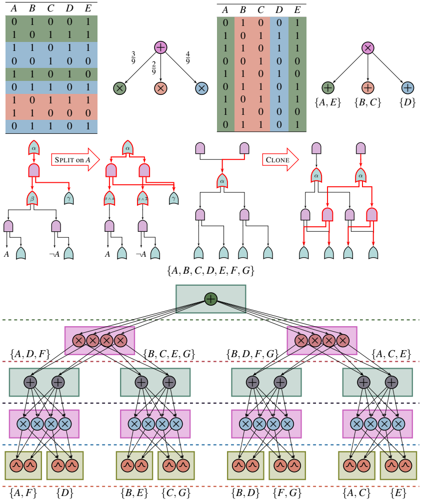
- Usually recursive;
- Splits data by similarity and stat dep;
- Stat dep usually costly;
- Usually tree-shaped.

Incremental Approaches

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

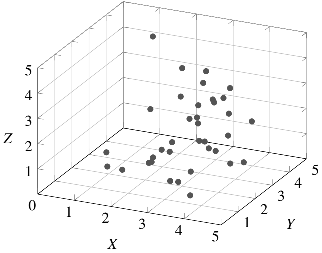
Random Approaches

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

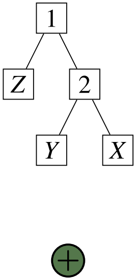


Where ~~are~~ were we ~~right now~~ then?

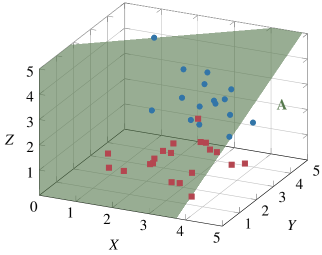
Name	Class	Time Complexity	# hyperparams	Accepts logic?	Sm?	Dec?	Det?	Str Dec?	$\{0,1\}$?	$\mathbb{N}$?	$\mathbb{R}$?	Reference
LearnSPN	DIV	$\begin{cases}\mathcal{O}(nkmc) & \text{, if sum}\\ \mathcal{O}(nm^3) & \text{, if product}\end{cases}$	≥ 2	✗	✓	✓	✗	✗	✓	✓	✓	Gens and Domingos [2013]
ID-SPN	DIV	$\begin{cases}\mathcal{O}(nkmc) & \text{, if sum}\\ \mathcal{O}(nm^3) & \text{, if product}\\ \mathcal{O}(ic(rn+m)) & \text{, if input}\end{cases}$	$\geq 2+3$	✗	✓	✓	✗	✗	✓	✓	✗	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. [2018]
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	✗	✓	✓	✗	✓	✓	✓	✓	Geh and Mauá [2021]

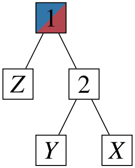


Z
X
Y
0
1
2
3
4
5

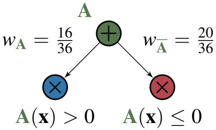


1
Z
2
Y
X



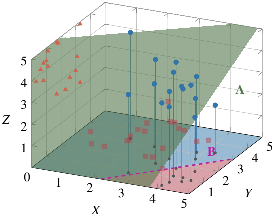


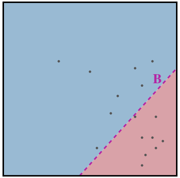
$$\mathbf{A}(x,y,z) = \begin{bmatrix} x & y & z \end{bmatrix} \cdot \underbrace{\begin{bmatrix} -0.31 \\ -0.40 \\ 0.85 \end{bmatrix}}_{a} + \underbrace{1}_{\theta}$$

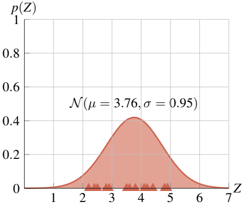


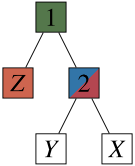
$w_{\mathbf{A}}$: probability of $\mathbf{A}(\mathbf{x}) > 0$

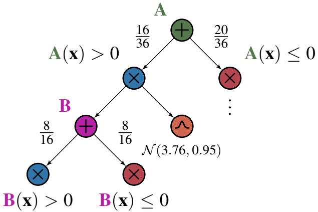
LearnRP









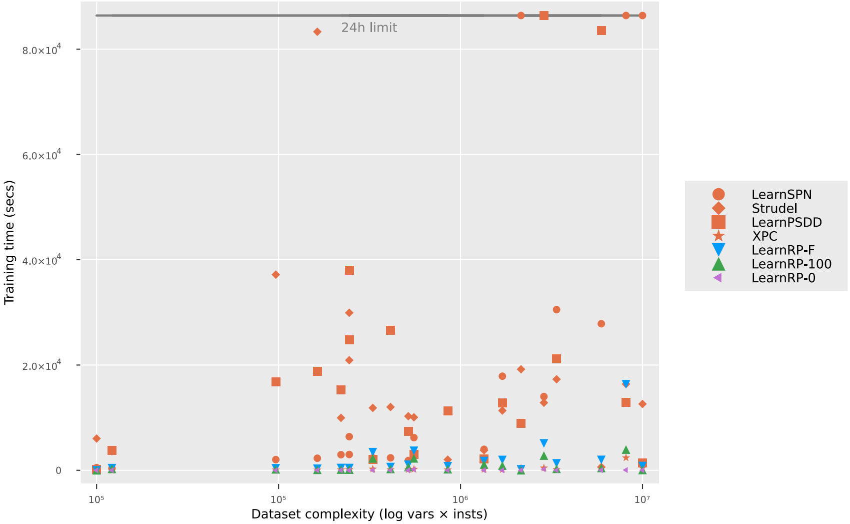


$$\mathbf{B}(x,y) = \begin{bmatrix} x & y \end{bmatrix} \cdot \underbrace{\begin{bmatrix} 1.10 \\ -1.00 \end{bmatrix}}_{b} - \underbrace{2.43}_{\gamma}$$

LearnRP — Experiments

Dataset	LearnSPN	Strudel	LearnPSDD	XPC	Prometheus	LearnRP
ACCIDENTS	-30.03	|-28.73|	-30.16	-31.02	**-27.91**	-28.66
AD	-19.73	-16.38	-31.78	**-15.50**	-23.96	|-19.26|
AUDIO	-40.50	-41.50	-39.94	-40.91	**-39.80**	|-40.27|
BBC	|-250.68|	-254.41	-253.19	**-248.34**	-248.50	-254.15
NETFLIX	|-57.02|	-58.69	**-55.71**	-57.58	-56.47	-57.02
BOOK	-35.88	-34.99	-34.97	|-34.75|	-34.40	**-33.56**
20-NEWSGRP	-155.92	-154.47	-155.97	-153.75	|-154.17|	**-152.63**
REUTERS-52	|-85.06|	-86.22	-89.61	-84.70	**-84.59**	-85.69
WEBKB	-158.20	-155.33	-161.09	-153.67	|-155.21|	**-153.52**
DNA	**-82.52**	-86.22	-88.01	-86.61	|-84.45|	-83.57
JESTER	-75.98	-55.03	**-51.29**	-53.43	-52.80	|-52.92|
KDD	-2.18	|-2.13|	**-2.11**	-2.15	-2.12	-2.14
KOSAREK	-10.98	-10.68	**-10.52**	-10.77	-10.59	|-10.62|
MSNBC	-6.11	**-6.04**	-6.04	-6.18	|-6.04|	-6.33
MSWEB	-10.25	**-9.71**	|-9.89|	-9.93	-9.86	-9.90
NLTCS	-6.11	-6.06	**-5.99**	|-6.05|	-6.01	-6.22
PLANTS	-12.97	|-12.98|	-13.02	-14.19	**-12.81**	-13.77
PUMSB-STAR	|-24.78|	-24.12	-26.12	-26.06	**-22.75**	-26.12
EACHMOVIE	|-52.48|	-53.67	-58.01	-54.82	-51.49	**-51.41**
RETAIL	-11.04	-10.81	**-10.72**	-10.94	-10.87	|-10.84|
Avg. Rank	4.28 ± 1.50	3.75 ± 1.48	|3.58 ± 2.14|	3.95 ± 1.61	$\mathbf{2.15 \pm 1.04}$	3.30 ± 1.65
Pos. (mean)	6th	4th	|3rd|	5th	**1st**	2nd

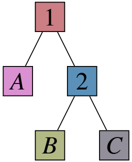
LearnRP — Experiments

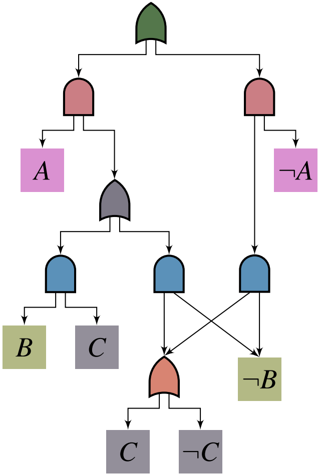


Logic Circuits $\subset$ Probabilistic Circuits

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

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



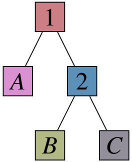


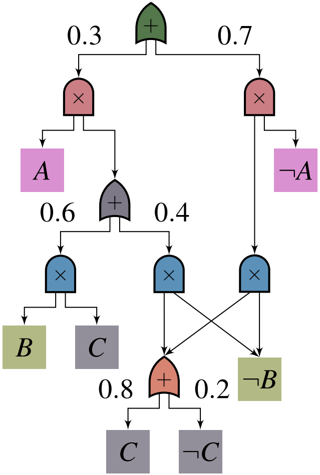
[Kisa et al., 2014]

Logic Circuits $\subset$ Probabilistic Circuits

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

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

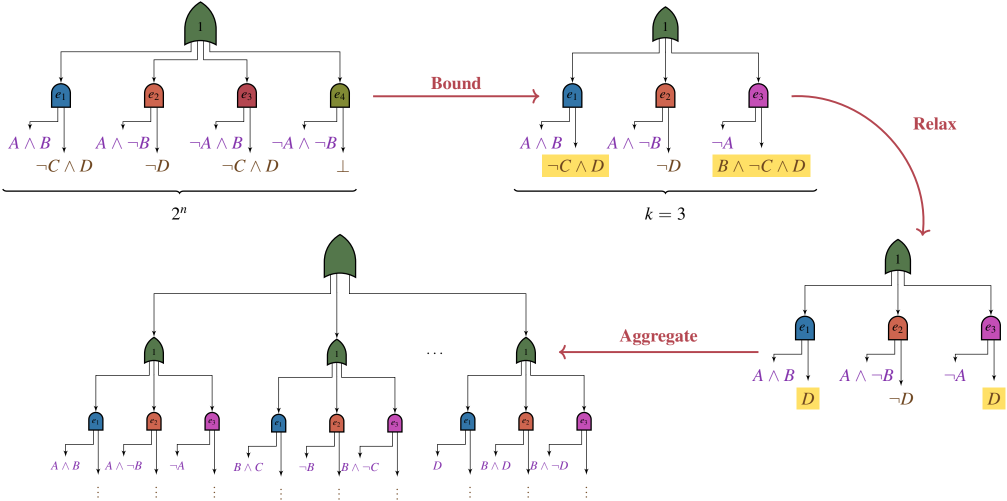




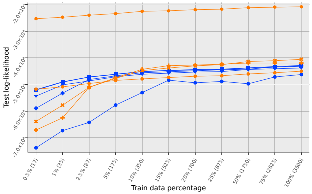
[Kisa et al., 2014]

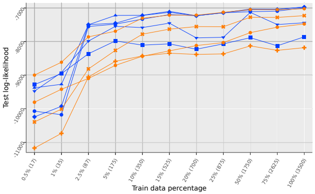
SamplePSDD

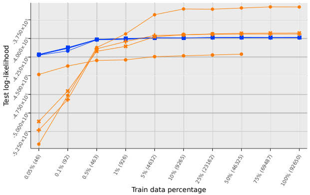
$$\phi(A,B,C,D) = (A \wedge \neg B \wedge \neg D) \vee (B \wedge \neg C \wedge D)$$

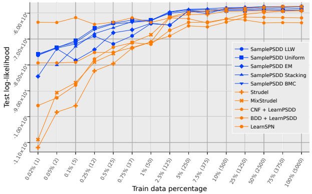


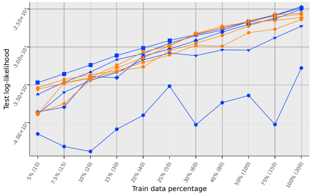
SamplePSDD — Experiments









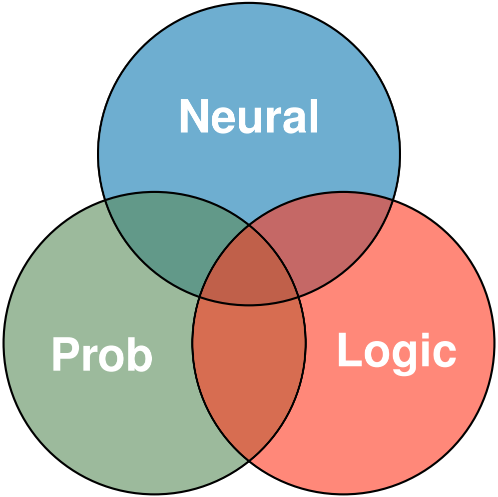


Mattei et al. [2020], Kamishima [2003], Shen et al. [2017], Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]

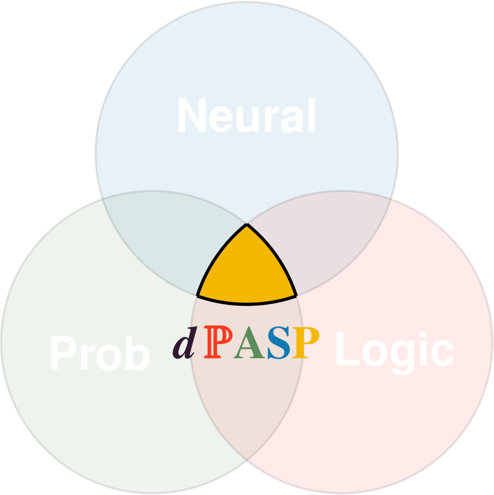
*d*PASP

***D**ifferentiable **P**robabilistic **A**nswer **S**et **P**rogramming*
For Neurosymbolic Learning and Reasoning

Renato Lui Geh, Jonas Gonçalves, Igor Cataneo Silveira,
Denis Deratani Mauá, Fabio Gagliardi Cozman, Yuka Machino


Neural
Prob
Logic

$d\mathbb{P}$ASP for Neurosymbolic Learning and Reasoning



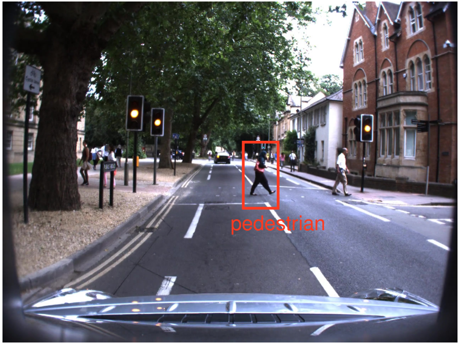
The ROAD-R Challenge

```
% Front feed.
front_img = Car.front_feed()
% Back feed.
back_img  = Car.back_feed()
...
% Neural net classifying objects ahead.
object_net = new ObjNet()
% These can be pedestrian, traffic sign, car, etc)
obj = object_net(front_img)
% If object is traffic light.
if obj == "traffic light":
    % Neural net for detecting traffic light.
    light_net = new LightNet()
    light = light_net(obj.img)
    if light == "red": Car.stop()
```

Giunchiglia et al. [2023]

The ROAD-R Challenge

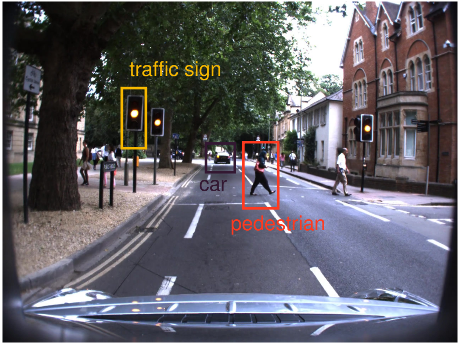
```
% Front feed.
front_img = Car.front_feed()
% Back feed.
back_img  = Car.back_feed()
...
% Neural net for detecting who's crossing.
cross_net = new CrossNet()
who = cross_net(front_img)
% If it's a pedestrian or cyclist, stop the car!
if who == "pedestrian" or who == "cyclist": Car.stop()
% Otherwise, if it's a car, slow down.
elseif who == "car": Car.slow_down()
```



Giunchiglia et al. [2023]

The ROAD-R Challenge

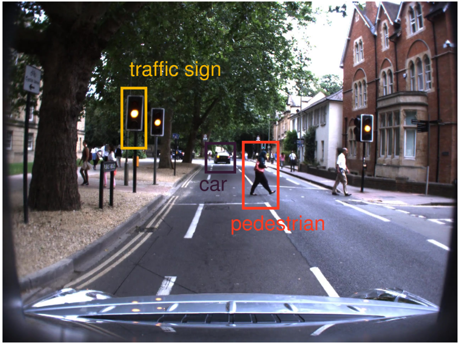
```
front_img = Car.front_feed()
back_img  = Car.back_feed()
...
object_net = new ObjNet()
obj = object_net(front_img)
if obj == "traffic light":
    light_net = new LightNet()
    light = light_net(obj.img)
    if light == "red": Car.stop()
cross_net = new CrossNet()
who = cross_net(front_img)
if who == "pedestrian" or who == "cyclist": Car.stop()
elseif who == "car": Car.slow_down()

% What's the prob we step on the gas?
???
% Prob pedestrian is crossing if slowing down?
???
% How do we learn all nets jointly?
???
```



Giunchiglia et al. [2023]

The ROAD-R Challenge

```
feed(front) ~ train(...), test(...).
feed(back) ~ train(...), test(...).
...
?::obj(Img, {ped,tsign,car,cyc}) as @net :- feed(Img).
?::tlight(Img, {r,y,g}) as @tnet :- obj(Img, tsign).
:- tlight(Img, r), tlight(Img, g).
stop :- tlight(front, r).
?::cross(Img, {ped,car,cyc}) as @cnet :- feed(Img).
stop :- cross(front, ped).
stop :- cross(front, cyc).
slow_down :- cross(front, car).
go_ahead :- not slow_down, not stop.

% What's the prob we step on the gas?
#query go_ahead.
% Prob pedestrian is crossing if slowing down?
#query cross(front, ped) | slow_down.
% How do we learn all nets jointly?
#learn @train_data, lr = 0.1, niters = 5.
```



Giunchiglia et al. [2023]

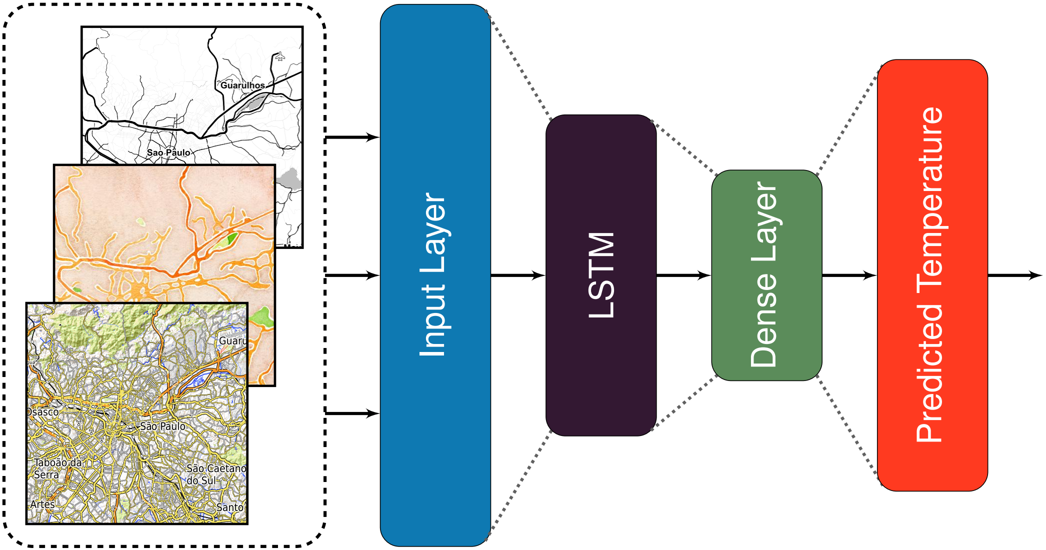
Logic Programming

```
% Fact: a true statement.
temp(sp, hot).

% Rule: if RHS, then LHS is true.
weather(sp, rain) :- temp(sp, hot).

% Rule w/ vars: if RHS for every X, then LHS is true (for every X).
hail(X) :- temp(X, hot), weather(X, rain).
temp(sp, hot). weather(sp, rain). % then hail(sp).
temp(la, hot). weather(la, rain). % then hail(la).

% Choice disjunction: one must be true.
forecast(sp, rainy); forecast(sp, cloudy); forecast(sp, sunny).
```

Probabilistic Logic Programming

```
% Probabilistic fact: true with some probability.
0.8::temp(sp, hot).

% Probabilistic rule: if RHS, then LHS is true with some probability.
1/4::weather(sp, rain) :- temp(sp, hot).

% Probabilistic rule w/ vars.
0.2::hail(X) :- temp(X, hot), weather(X, rain).
temp(sp, hot). weather(sp, rain). % then maybe hail(sp)?
temp(la, hot). weather(la, rain). % then maybe hail(la)?

% Annotated disjunction: one must be true, each with their own probability.
0.3::forecast(sp, rainy); 0.2::forecast(sp, cloudy); 0.5::forecast(sp, sunny).
```

Weather Forecasting

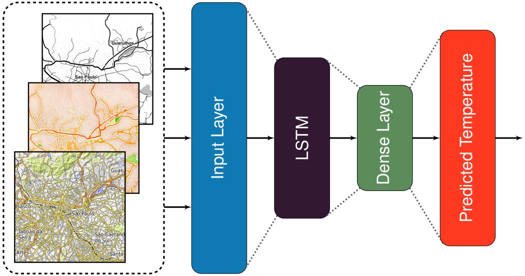


Neural Probabilistic Logic Programming

```
% Neural fact: true with some prob according to input of neural net!
?::temp(sp, hot) as @temp_net :- thermometer_data(sp).

% Neural rule: prob rule but neural net defines prob.
?::weather(sp, rain) as @weather_net :- temp(sp, hot), weather_data(sp).

% Neural rule w/ vars.
?::hail(X) as @hail_classifier :- temp(X, hot), weather(X, rain),
                                   hail_data(X).
temp(sp, hot). weather(sp, rain). % prob of hail(sp) from data.
temp(la, hot). weather(la, rain). % prob of hail(la) from data.

% Neural annotated disjunction: categorical classifiers.
?::forecast(X, {rainy,cloudy,sunny}) as @forecast_net :- forecast_data(X).
```

Weather Forecasting

```
#python
def temp_net(): ...
def temp_train(): ...
def temp_test(): ...
...
def hail_net(): ...
...
#end.
```

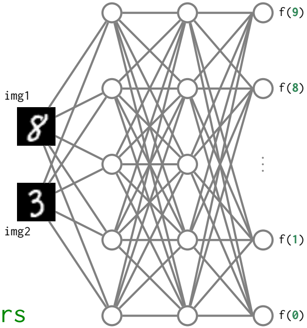



```
temp_data(sp) ~ train(@temp_train("sp")), test(@test_train("sp")).
temp_data(la) ~ train(@temp_train("la")), test(@test_train("la")).
?::temp(X, hot, Y) as @temp_net :- temp_data(X, Y).
...
?::hail(X) as @hail_net :- temp(X, hot), weather(X, rain),
                           hail_data(X).
% What is ℙ(temp(sp, hot) | not hail(sp))?
#query temp(sp, hot) | not hail(sp).
```

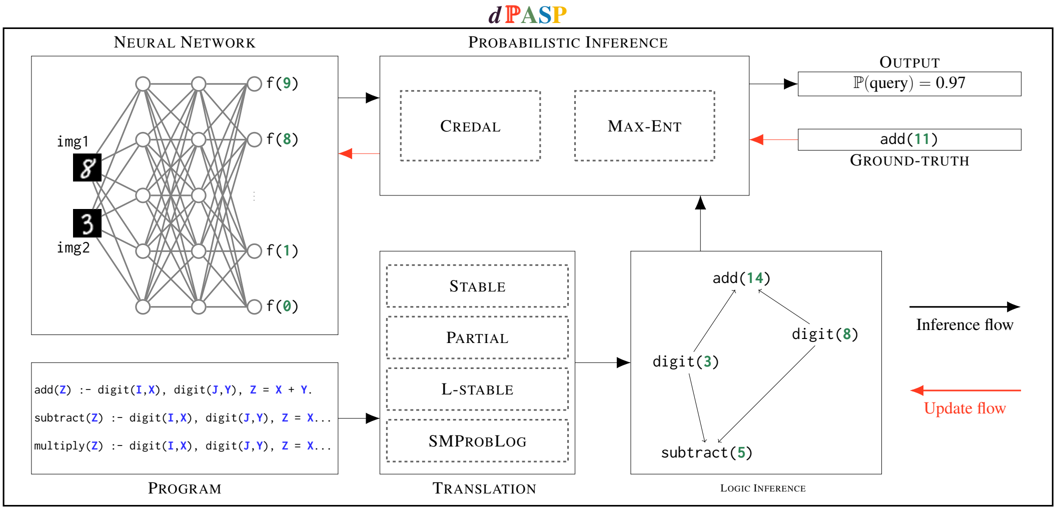
MNIST Addition

Parsing arithmetic expressions, e.g. X + Y $= f($$) + f($$) = ?$

```
% neural rule
?::digit(Image, {0..9}) :- data(Image).
% data loaders -- interact with Python code
data(img1) ~ test(@mnist_test), train(@mnist_train).
data(img2) ~ test(@mnist_test), train(@mnist_train).
% prob. answer set program
add(Z) :- digit(I, X), digit(J, Y), Z = X + Y.
subtract(Z) :- digit(I, X), digit(J, Y), Z = X - Y.
multiply(Z) :- digit(I, X), digit(J, Y), Z = X * Y.
% learn the program end-to-end and pass learning parameters
#learn @mnist_sum, lr = 1., niters = 5, ..., batch = 1000.
% inference: what is the probability of X + Y = 14 given X = 8?
#query add(11) | digit(img1, 8).
```

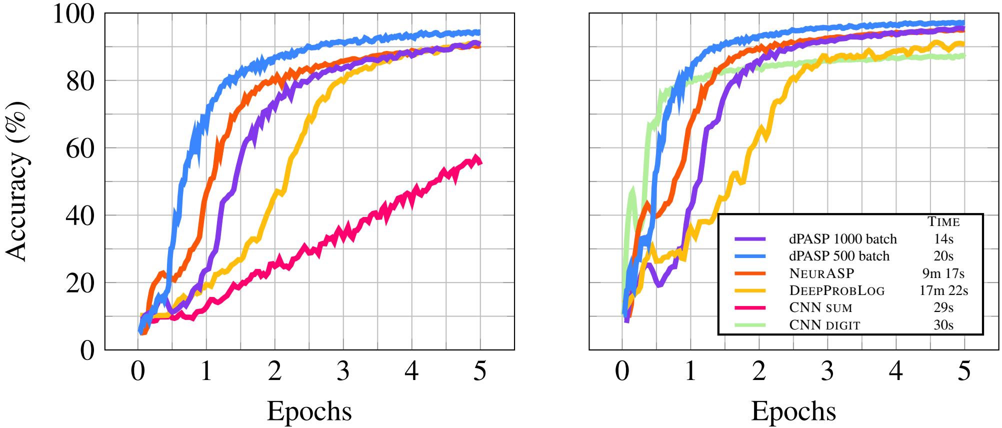


A Bird's Eye View of dℙASP

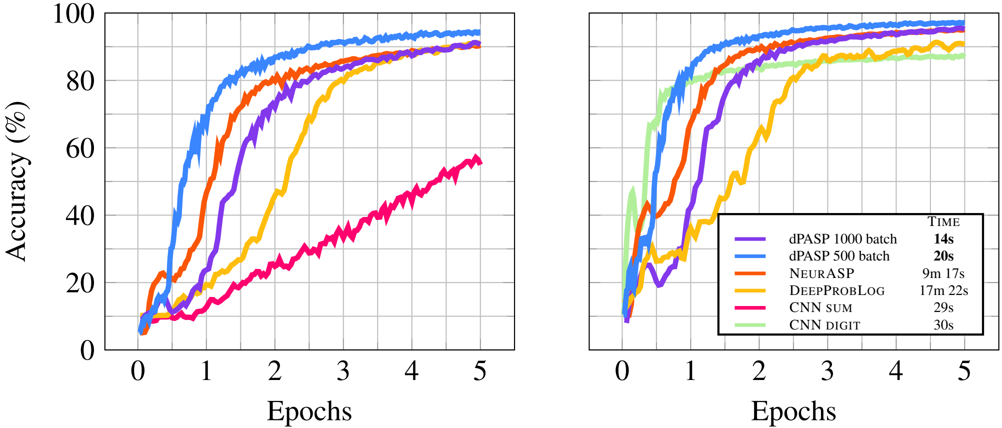


Experiments

How much **faster** is dPASP on the MNIST Add?



How much **faster** is dPASP on the MNIST Add?



Argumentation

```
% Introduction.
0.7::claim(1).
```

Ever since researchers at the Roslin Institute in Edinburgh cloned an adult sheep, there has been an ongoing debate about whether cloning technology is morally and ethically right or not. Some people argue for and others against and there is still no agreement whether cloning technology should be permitted. However, as far as I'm concerned, ***[cloning is an important technology for humankind]*** *Major Claim* *since* <u>*[it would be very useful for developing novel cures]*</u> *Claim 1*.

Argumentation

```
% Introduction.
0.7::claim(1).

% Paragraph 1.
0.9::premise(3) :- premise(4).
0.8::premise(3) :- premise(5).
premise(2) :- premise(1).
0.9::claim(2) :- premise(1).
0.7::claim(2) :- premise(3).
```

First, [cloning will be beneficial for many people who are in need of organ transplants] Claim 2*. [Cloned organs will match perfectly to the blood group and tissue of patients]* Premise 1 *since [they can be raised from cloned stem cells of the patient]* Premise 2*. In addition, [it shortens the healing process]* Premise 3*. Usually, [it is very rare to find an appropriate organ donor]* Premise 4 *and [by using cloning in order to raise required organs the waiting time can be shortened tremendously]* Premise 5*.*

Argumentation

```
% Introduction.
0.7::claim(1).
% Paragraph 1.
0.9::premise(3) :- premise(4).
0.8::premise(3) :- premise(5).
premise(2) :- premise(1).
0.9::claim(2) :- premise(1).
0.7::claim(2) :- premise(3).
...
% Paragraph 3.
0.9::premise(11) :- premise(10).
0.8::claim(5) :- not premise(11), premise(9).
```

Admittedly, [cloning could be misused for military purposes] $^{\text{Claim 5}}$*. For example, [it could be used to manipulate human genes in order to create obedient soldiers with extraordinary abilities]* $^{\text{Premise 9}}$*. However, because [moral and ethical values are internationally shared]* $^{\text{Premise 10}}$*, [it is very unlikely that cloning will be misused for militant objectives]* $^{\text{Premise 11}}$

Argumentation

```
% Introduction.
0.7::claim(1).
% Paragraph 1.
0.9::premise(3) :- premise(4).
0.8::premise(3) :- premise(5).
premise(2) :- premise(1).
0.9::claim(2) :- premise(1).
0.7::claim(2) :- premise(3).

...
% Paragraph 3.
0.9::premise(11) :- premise(10).
0.8::claim(5) :- not premise(11), premise(9).

...
% Conclusion.
major_claim) :- claim(1), claim(2), claim(3),
                claim(4), not claim (5),
                not claim(6), claim(7).
% What's the prob major claim holds given we are
% sure cloning will be misused for military purposes?
#query major_claim | premise(11).
```

To sum up, although $\underbrace{\textit{[permitting cloning might bear some risks like misuse for military purposes]}}_{\textit{Claim 6}}$*, I strongly believe that* ***[this technology is beneficial to humanity]*** $^{\textit{Major Claim}}$*. It is likely that* $\underbrace{\textit{[this technology bears some important cures which will significantly improve life conditions]}}_{\textit{Claim 7}}$*.*

dPASP

***D**ifferentiable **P**robabilistic **A**nswer **S**et **P**rogramming*
For Neurosymbolic Learning and Reasoning

Renato Lui Geh, Jonas Gonçalves, Igor Cataneo Silveira,
Denis Deratani Mauá, Fabio Gagliardi Cozman, Yuka Machino

GitHub

Learn dPASP!