Neural Probabilistic Logic Programming in Discrete-Continuous Domains

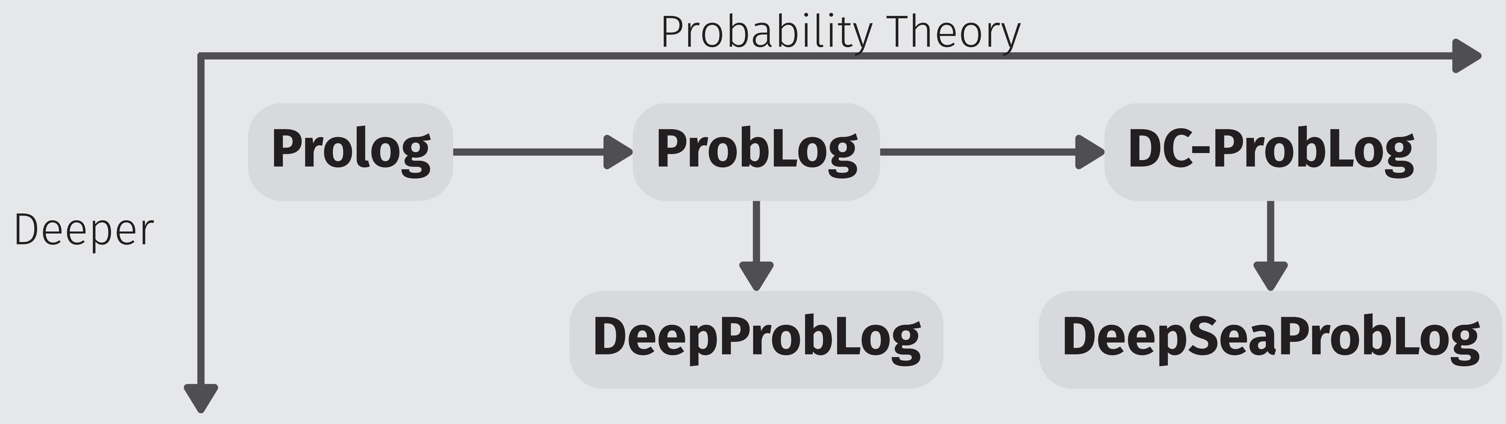
Lennert De Smet[1], Pedro Zuidberg dos Martires[2], Robin Manhaeve[1], Giuseppe Marra[1], Angelika Kimmig[1] and Luc De Raedt[1,2]
[1] Department of Computer Science, KU Leuven, Belgium, [2] Center for Applied Autonomous Systems, Örebro University, Sweden
✉ lennert.desmet@kuleuven.be

Neural-Symbolic AI (NeSy)

Neural Networks	Discrete-*Continuous*	Symbolic Methods
Backpropagation	Probability theory	Logic

Probabilistic NeSy

Declarative Family



Building Blocks

```
% Declaring Neural Distributional Facts (NDFs)
humid ~ bernoulli(0.5).
cloudy(🌎) ~ categorical(cloudy_net(🌎)).
temp(🌎) ~ normal(temp_net(🌎)).

% Reasoning with Probabilistic Comparison Formulas (PCFs)
rainy(🌎) :- humid, cloudy(🌎) =\= 0.

good_weather(🌎) :- temp(🌎) < 0, rainy(🌎).
good_weather(🌎) :- temp(🌎) > 20, not rainy(🌎).

% Querying probabilities of logical statements
query(good_weather(image_020823))
```

Inference

Logic represented by tractable **probabilistic circuit**

Neural distributions as weighing functions

$$P(\mathrm{q}) = \int \sum \prod \mathbb{1}(c(\mathbf{x}))\ p_\Lambda(\mathbf{x})\ \mathrm{d}\mathbf{x}$$

$$P(\texttt{temp(🌎) < 0}) = \int \mathbb{1}(x<0) \frac{\exp\left(-\frac{(x-\mu(🌎))^2}{2\sigma^2(🌎)}\right)}{\sqrt{2\pi}\sigma(🌎)}\,\mathrm{d}x$$

Learning

Obstacles

- Indicator functions are not differentiable

$$\mathbb{1}(c(\mathbf{x}))\ p_\Lambda(\mathbf{x}) \xrightarrow{\text{relaxation}} \sigma(\tau \cdot c(\mathbf{x}))\ p_\Lambda(\mathbf{x})$$

- Sampling approximation blocks gradient

$$\sigma(\tau \cdot c(\mathbf{x}))\ p_\Lambda(\mathbf{x}) \xrightarrow{\text{reparametrisation}} \sigma(\tau \cdot c(r_\Lambda(\mathbf{u})))\ p(\mathbf{u})$$

Result

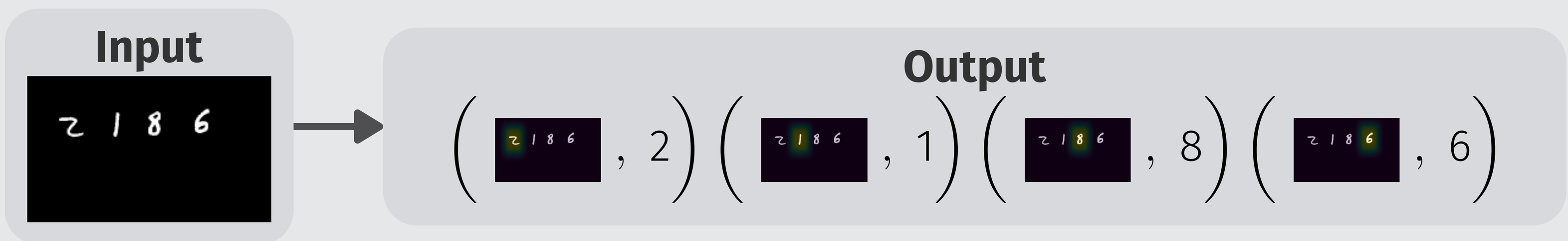
Convergence to **unbiased gradients** for $\tau \to +\infty$.

Efficient backpropagation through the deep probabilistic-logical model

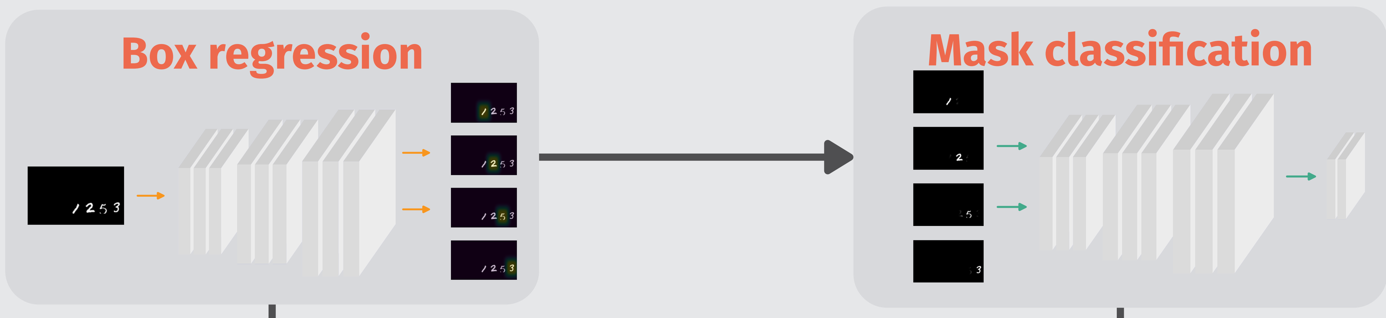
$$\partial_\lambda P(\mathrm{q}) \approx \int \partial_\lambda \sum \prod \sigma(\tau \cdot c(r_\Lambda(\mathbf{u})))\ p(\mathbf{u})\ \mathrm{d}\mathbf{u}$$

NeSy Attention

Predict + localise digits without bounding box supervision



Neural-Symbolic Architecture




```
box(X) ~
    generalisednormal(X).

left_right(■, ■, ■, ■) :-
    box(■) < box(■),
    box(■) < box(■),
    box(■) < box(■).
```

Box ordering

```
class(Mask) ~
    categorical(class_net(Mask)).

supervise(■, ■, ■, ■) :-
    class(■) =:= 1, class(■) =:= 2,
    class(■) =:= 5, class(■) =:= 3.
```

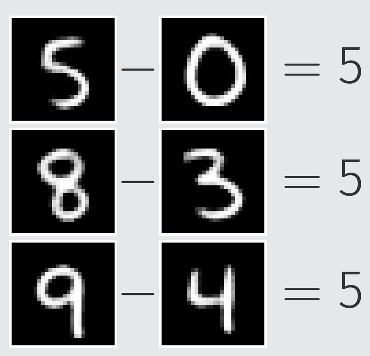
Supervision

Method	Results	
	acc.	IoU
DeepSeaProbLog	93.77 ± 0.57	17.69 ± 0.23
LTN	76.50 ± 12.10	10.73 ± 1.69
Neural Baseline	54.71 ± 14.33	6.26 ± 1.77

NeSy Generation

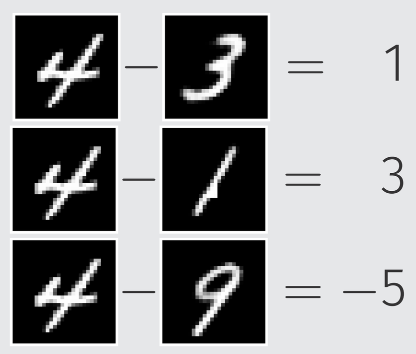
Task 1: Distant Learning

Learn to fill in ? − ? = 5



Task 2: Conditional Generation

Zero-shot completion of 4 − ? = 5 in **same style**.



Links

Personal | Twitter | Code

KU Leuven
Department of Computer Science
Celestijnenlaan 200A box 2402
3001 Leuven, Belgium

DTAI | ÖREBRO UNIVERSITET | WASP | fwo | TAILOR