From Statistical Relational to Neurosymbolic AI

Giuseppe Marra

Flanders AI Research Program

KU LEUVEN

LEUVEN.AI INSTITUTE

LEMUR
Learning with Multiple Representations

fwo

Joint work

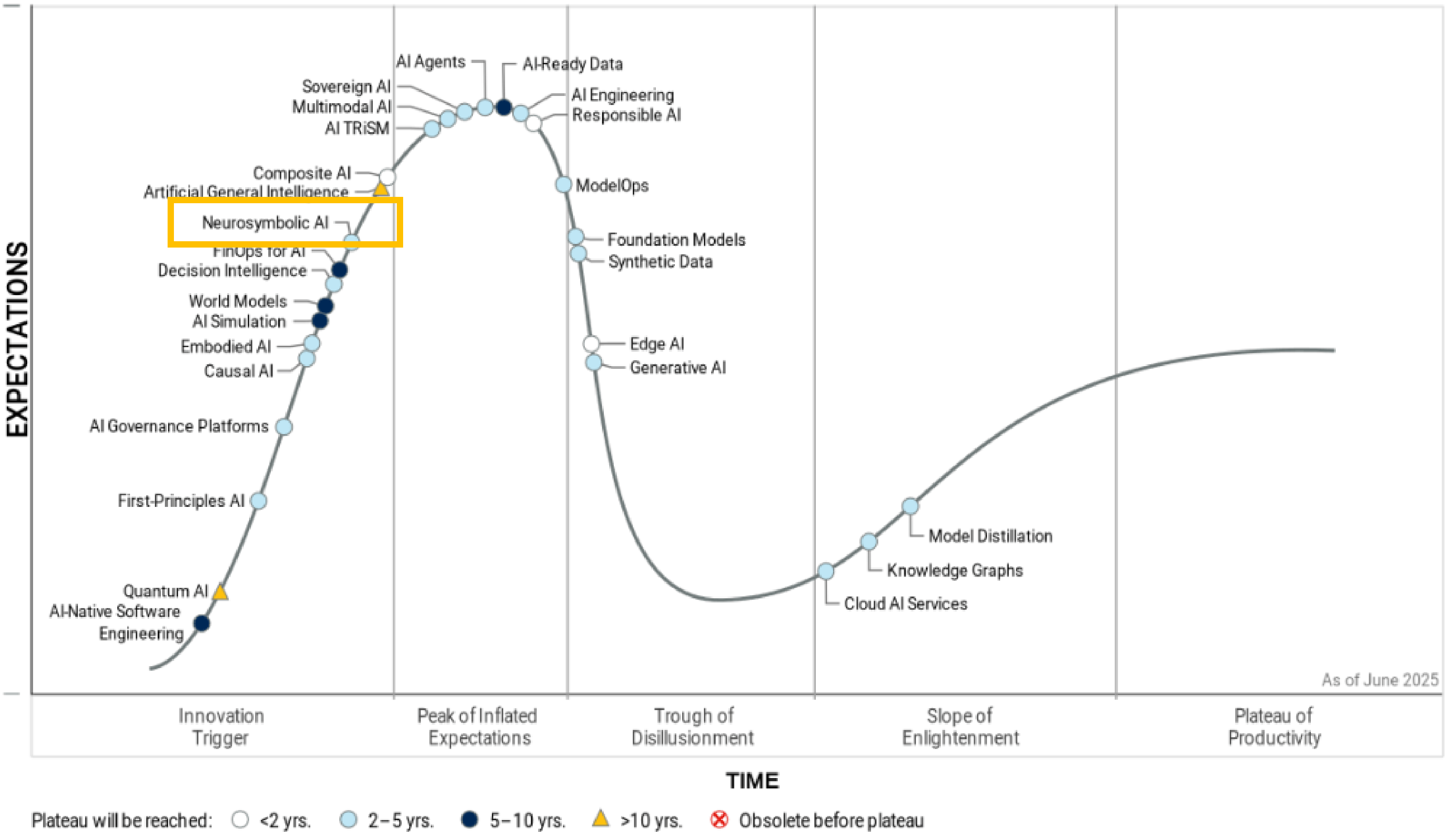
- Luc De Raedt, Robin Manhaeve, Thomas Winters, Vincent Derkinderen, Wen-Chi Yang, Lennert De Smet, Gabriele Venturato, David Debot

- Pietro Barbiero, Michelangelo Diligenti, Francesco Giannini, Marco Maggini, Marco Gori, Eleonora Misino, Emanuele Santone,

The neurosymbolic integration quest

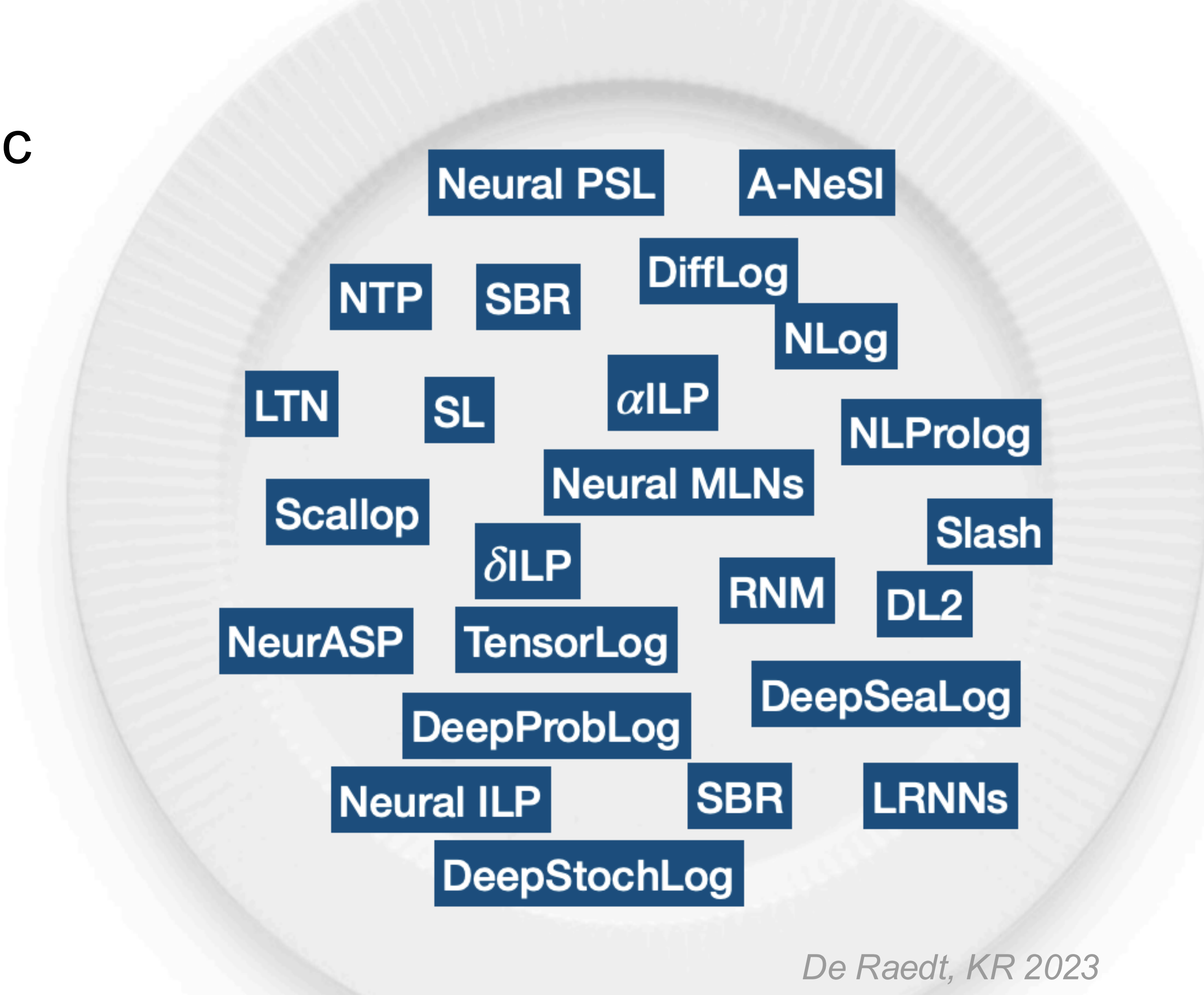
Subsymbolic Approaches	Symbolic Approaches
data	knowledge
learning	reasoning

Hype Cycle for Artificial Intelligence, 2025



Gartner

The neurosymbolic alphabet soup

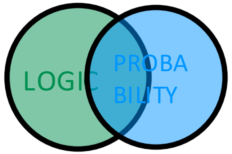


De Raedt, KR 2023

From Statistical Relational….

Another paradigm for learning and reasoning

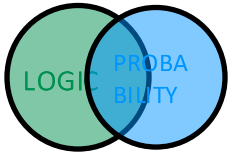
StarAI = Logic + PGMs



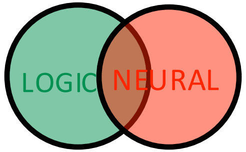
From Statistical Relational to Neurosymbolic AI

Another paradigm for learning and reasoning

StarAI = Logic + PGMs



NeSy = Logic + Neural Networks



From Statistical Relational to Neurosymbolic AI

7 dimensions:

- Proof vs Models
- Syntax
- Semantics
- Structure vs Parameter Learning
- Representations (symbolic vs subsymbolic)
- Recovery of original paradigms
- Tasks

ELSEVIER

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

From statistical relational to neurosymbolic artificial intelligence: A survey

Giuseppe Marra [a,*], Sebastijan Dumančić [c], Robin Manhaeve [a], Luc De Raedt [a,b]

[a] KU Leuven, Department of Computer Science and Leuven.AI, Belgium
[b] Örebro University, Center for Applied Autonomous Sensor Systems, Sweden
[c] Delft University of Technology, Department of Software Technology, Netherlands

Frameworks	Inference	Syntax	Semantics	Learning	Representations	Paradigms	Tasks
	(P)roof (M)odel	(P)ropositional (R)elational (FOL)	(M)inimal (S)table (C)lassical (F)uzzy (P)robability	(P)arameters (S)tructure	(S)ymbolic (Sub)symbolic	Logic (L/l) Probability (P/p) Neural(N/n)	(D)istant (S)upervision (S)emi (S)upervised (KGC)ompletion (G)enerative (K)nowledge (I)nduction
αILP [111]	P+M	FOL	S + P	P + S	S	Ln	KI
∂ILP [39]	P	R	M + F	P + S	S	Ln	DS + KI
DeepProbLog [72]	P+M	FOL	M + P	P+S	S+Sub	LpN	DS + KI
DeepStochLog [132]	P	FOL	M + P	P	S	LpN	DS + SS
DiffLog [112]	P	R	M + F	P+S	S	Ln	KI
DL2 [40]	M	P	C + F	P	S+Sub	lN	DS + SS
DLM [77]	M	FOL	C + F + P	P	S	lPN	SS + KGC
LRNN [116]	P	R	M + F	P + S	S + Sub	LN	KGC + KI
LTN [5]	M	FOL	C + F	P	S + Sub	lN	DS + SS
NeuralLP [137]	P	R	M + F	P	S	Ln	KGC + KI
NeurASP [138]	P+M	FOL	S + P	P	S	LpN	DS
NLM [35]	P	R	M + F	P + S	S	Ln	KGC + KI
NLog [121]	P	R	M + P	P	S	LpN	DS
NLProlog [131]	P	R	M + P	P + S	S + Sub	LpN	KGC + KI
NMLN [78]	M	FOL	C + P	P + S	S + Sub	lPN	KGC + G
NTP [102]	P	R	M + F	P + S	S + Sub	Ln	KGC + KI
RNM [76]	M	FOL	C + P	P	S	lPN	SS
SBR [33]	M	FOL	C + F	P	S+Sub	lN	DS + SS
Scallop [58]	P	FOL	M + P	P	S	LpN	DS
SL [133]	M	P	C + P	P	S	LpN	SS
Slash [113]	P+M	FOL	S + P	P	S	LpN	DS +SS

From Statistical Relational to Neurosymbolic AI

7 dimensions:

- **Proof vs Models**
- Syntax
- Semantics
- Structure vs Parameter Learning
- Representations (symbolic vs subsymbolic)
- Recovery of original paradigms
- Tasks

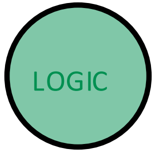
Marra et al, 2024 AIJ

The use of logic: Proof vs Model

Logic

```
Lion AND Gate -> Zoo.
Lion AND Wall -> Zoo.
```

How do we interpret these logic formulas?



The use of logic: Proof vs Model

Logic

```
Lion AND Gate -> Zoo.
Lion AND Wall -> Zoo.
```

Logic rules as computational rules (logic programs)

proof-based

Logic rules as constraints (SAT)

model-based

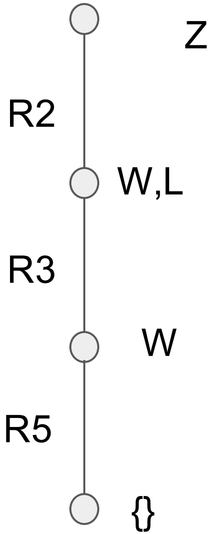
The use of logic: Proof vs Model

Logic

```
R1: Lion AND Gate -> Zoo.
R2: Lion AND Wall -> Zoo.
R3: Lion. R4: Gate. R5: Wall.
```

Logic rules as computational rules (logic programs)

proof-based



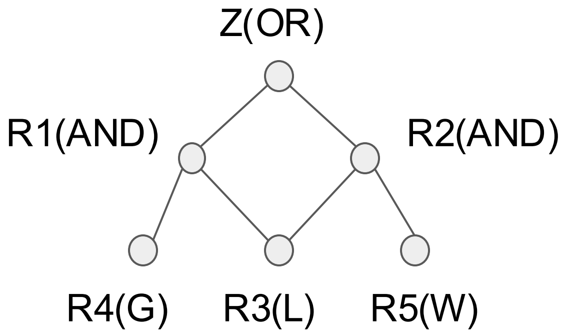
The use of logic: Proof vs Model

Logic

```
R1: Lion AND Gate -> Zoo.
R2: Lion AND Wall -> Zoo.
R3: Lion. R4: Gate. R5: Wall.
```

Logic rules as computational rules (logic programs)

proof-based



The use of logic: Proof vs Model

Logic

```
Lion AND Gate -> Zoo.
Lion AND Wall -> Zoo.
```

Logic rules as computational rules (logic programs)

Logic rules as constraints (SAT)

proof-based

model-based

The use of logic: Proof vs Model

Logic

```
Lion AND Gate -> Zoo.
```

Logic rules as constraints (SAT)

model-based

The use of logic: Proof vs Model

Logic

```
Lion AND Gate -> Zoo.
```

Logic rules as constraints (SAT)

model-based

L	G	Z	M
F	F	F	T
F	F	T	T
F	T	F	T
F	T	T	T
T	F	F	T
T	F	T	T
T	T	F	F
T	T	T	T

The use of logic: Proof vs Model

Logic

```
Lion AND Gate -> Zoo.
```

Logic rules as constraints (SAT)

model-based

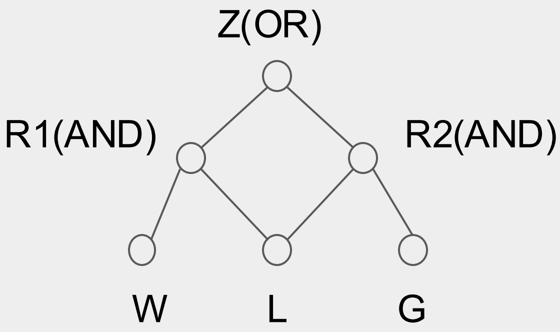
L	G	Z	M
F	F	F	T
F	F	T	T
F	T	F	T
F	T	T	T
T	F	F	T
T	F	T	T
T	T	F	F
T	T	T	T

The use of logic: Proof vs Model

Logic

```
Lion AND Gate -> Zoo.
Lion AND Wall -> Zoo.
```

Logic rules as computational rules
(logic programs)



proof-based

Logic rules as constraints
(SAT)

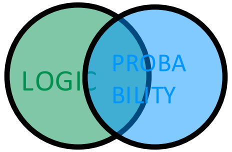
L	G	Z	M
F	F	F	T
F	F	T	T
F	T	F	T
F	T	T	T
...	...	...	...

model-based

The use of logic: Proof vs Model

Logic

Can we use the same perspective when we deal with uncertainty?



The use of logic: Proof vs Model

StarAI = Logic + Probabilities

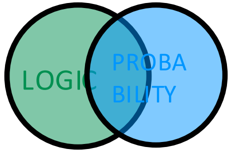
```
0.7:: Lion AND Gate -> Zoo.
0.3:: Lion AND Wall -> Zoo.
```

Stochastic Logic Programs

proof-based

Markov Logic

model-based



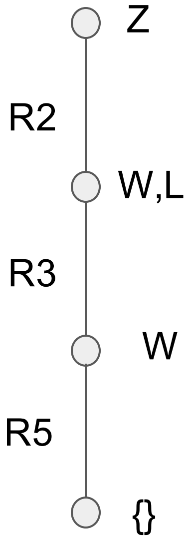
The use of logic: Proof vs Model

StarAI = Logic + Probabilities

```
0.7:: Lion AND Gate -> Zoo.
0.3:: Lion AND Wall -> Zoo.
1: Lion. 1: Gate. 1: Wall.
```

Stochastic Logic Programs

proof-based



Proof1

P(proof1) = p(R2) * p(R3) * p (R5)

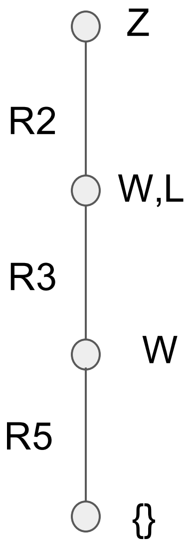
The use of logic: Proof vs Model

StarAI = Logic + Probabilities

```
0.7:: Lion AND Gate -> Zoo.
0.3:: Lion AND Wall -> Zoo.
1: Lion. 1: Gate. 1: Wall.
```

Stochastic Logic Programs

proof-based



Proof1

P(proof1) = p(R2) * p(R3) * p (R5)

Semantics:

Probability distribution over proofs

(akin to probabilistic grammars)

The use of logic: Proof vs Model

StarAI = Logic + Probabilities

```
0.7:: Lion AND Gate -> Zoo.
0.3:: Lion AND Wall -> Zoo.
```

Stochastic Logic Programs

proof-based

Markov Logic

model-based

The use of logic: Proof vs Model

StarAI = Logic + Probabilities

```
w1:: Lion AND Gate -> Zoo.
w2:: Lion AND Wall -> Zoo.
```

Markov Logic

model-based

L	W	G	Z	W
F	F	F	F	w1 + w2
F	F	F	T	…
F	F	T	F	…
F	F	T	T	…
...	...	...		...
T	T	T	F	0 + 0

The use of logic: Proof vs Model

StarAI = Logic + Probabilities

```
w1:: Lion AND Gate -> Zoo.
w2:: Lion AND Wall -> Zoo.
```

Weight of a model = sum of the weights of the formulas it makes True

Markov Logic

model-based

L	W	G	Z	W
F	F	F	F	w1 + w2
F	F	F	T	…
F	F	T	F	…
F	F	T	T	…
…	…	…		…
T	T	T	F	0 + 0

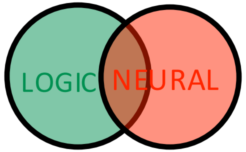
$$p(m) = \frac{e^{W}}{Z}$$

The use of logic: Proof vs Model

NeSy = Logic + Neural

```
Lion AND Gate -> Zoo.
Lion AND Wall -> Zoo.
```

Can we use the same perspective also in NeSy?



The use of logic: Proof vs Model

NeSy = Logic + Neural

```
Lion AND Gate -> Zoo.
Lion AND Wall -> Zoo.
```

Neural Program

Semantic-Based Regularizers

proof-based

model-based

Logic as a neural program

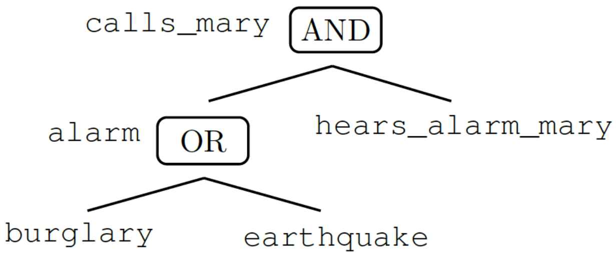
- KBANN (Towell and Shavlik AIJ 94)
- Turn a (propositional) Prolog program into a neural network and learn

Logic as a neural program

- KBANN (Towell and Shavlik AIJ 94)
- Turn a (propositional) Prolog program into a neural network and learn

Turn the program into an AND/OR tree (backward chaining)

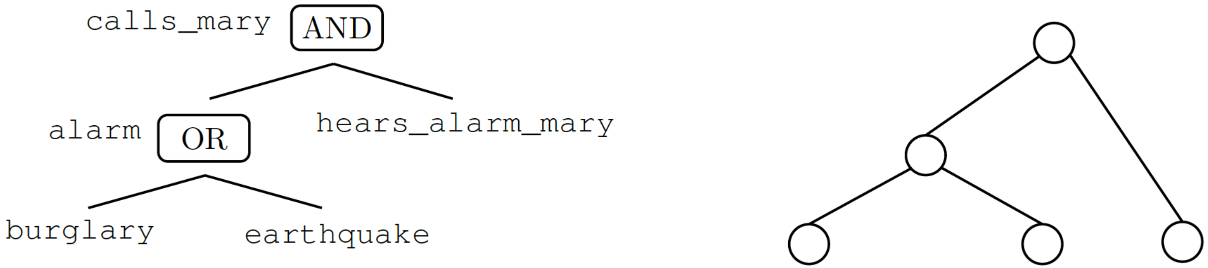
```
alarm <- earthquake.
alarm <- burglary.
calls_mary <- alarm,
              hears_alarm_mary.
```



Logic as a neural program

- KBANN (Towell and Shavlik AIJ 94)
- Turn a (propositional) Prolog program into a neural network and learn

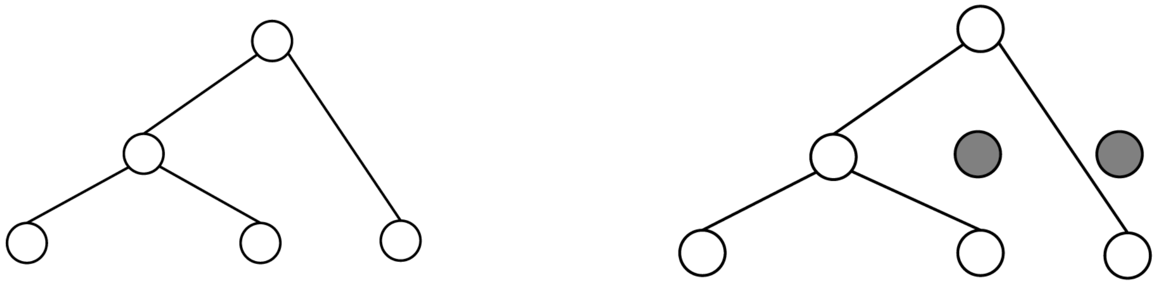
Turn AND/OR tree into a neural net (one neuron per node)



Logic as a neural program

- KBANN (Towell and Shavlik AIJ 94)
- Turn a (propositional) Prolog program into a neural network and learn

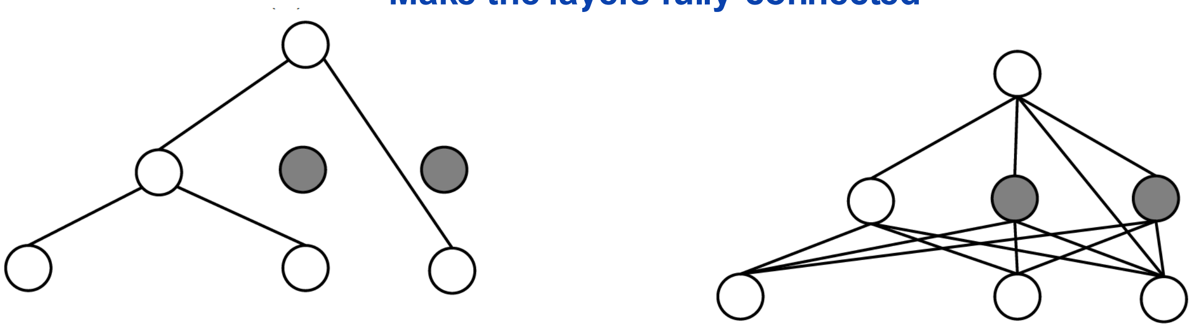
Add Hidden Nodes in a layered structure



Logic as a neural program

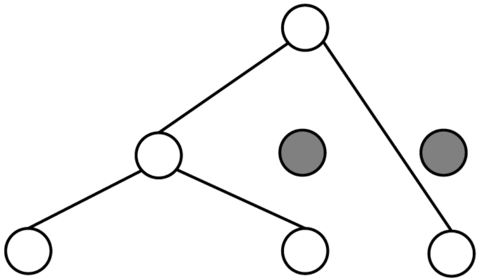
- KBANN (Towell and Shavlik AIJ 94)
- Turn a (propositional) Prolog program into a neural network and learn

Make the layers fully-connected

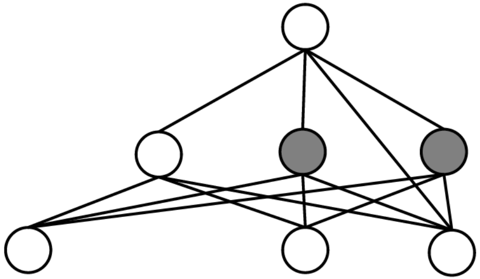


Logic as a neural program

- KBANN (Towell and Shavlik AIJ 94)
- Turn a (propositional) Prolog program into a neural network and learn

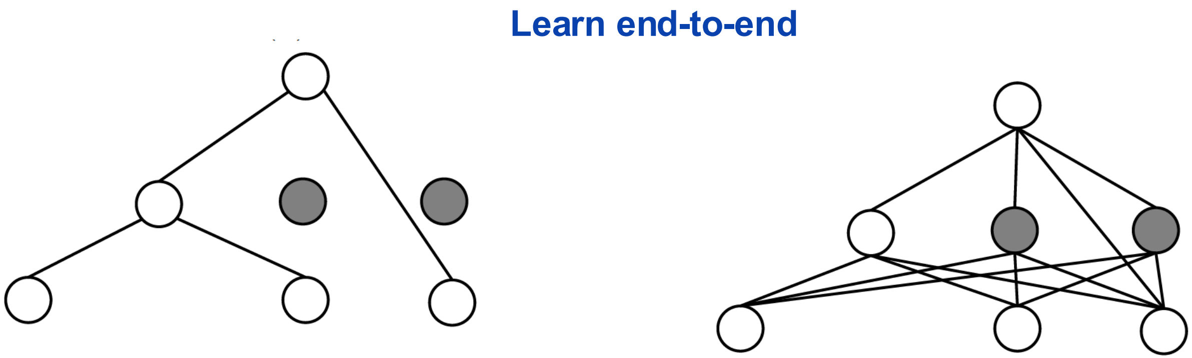


Initialise weights "coherent to logic"



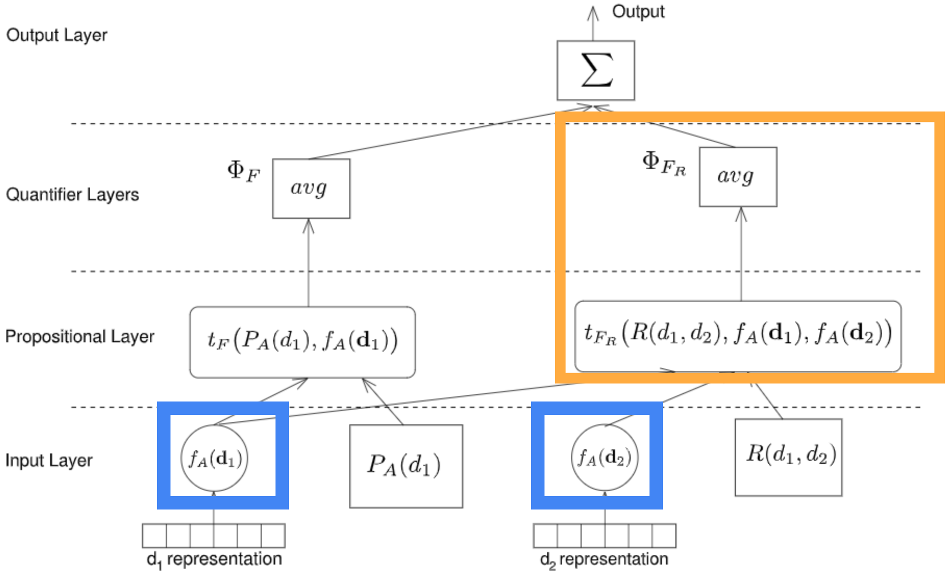
Logic as a neural program

- KBANN (Towell and Shavlik AIJ 94)
- Turn a (propositional) Prolog program into a neural network and learn



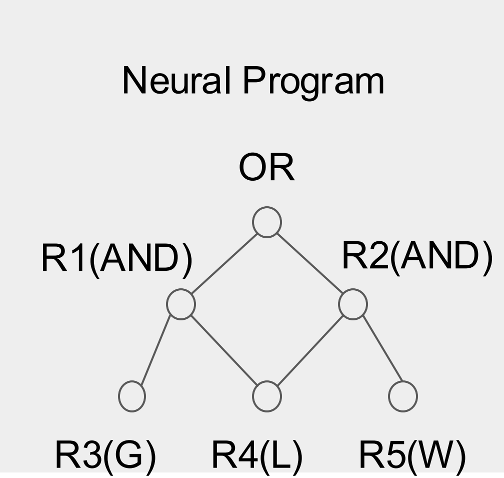
Logic as a regularizer

$$\forall d1, d2 : R(d1, d2) \rightarrow f_A(d1) \leftrightarrow f_A(d2)$$

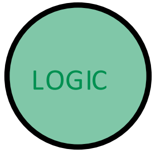


L	W	G	Z	W
F	F	F	F	w1 + w2
F	F	F	T	…
F	F	T	F	…
F	F	T	T	…
...	...	...		...
T	T	T	F	0 + 0

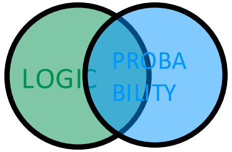
The use of logic: Proof



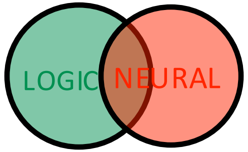
proof-based



Proof



Probabilistic Grammars



Neural Networks

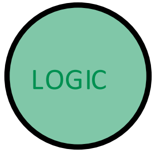
Inference structures = graphical models

Logic is a template for the architecture

Inference = Traversal

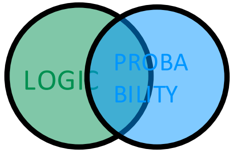
The use of logic: Model

Logic is used to:



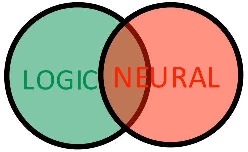
Check

Models



Weight

Models



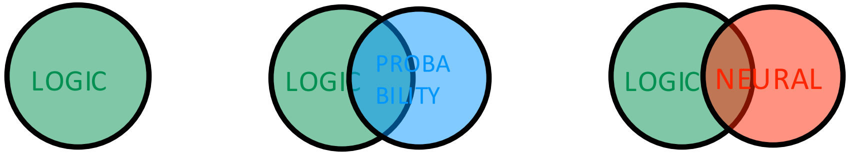
Weight

Networks' Outputs

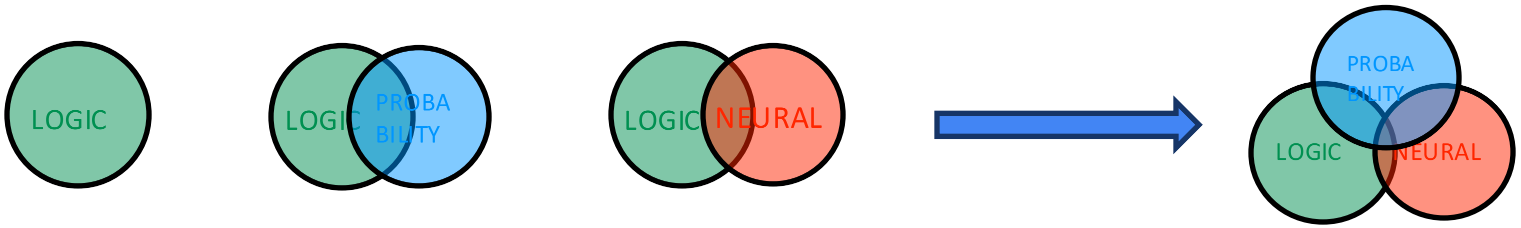
L	W	G	Z	W
F	F	F	F	w1 + w2
F	F	F	T	…
F	F	T	F	…
F	F	T	T	…
...	...	...		...
T	T	T	F	0 + 0

Models = variables of interest = neural network

Logic = constraint = expected behaviour = loss function


LOGIC
LOGIC
PROBA
BILITY
LOGIC
NEURAL

StarAI as a recipe for NeSy



The use of logic: Proof
Stochastic Definite Clause Grammars

Parse Sequences: [“0”, “+”, “9”, “+”, “1”].

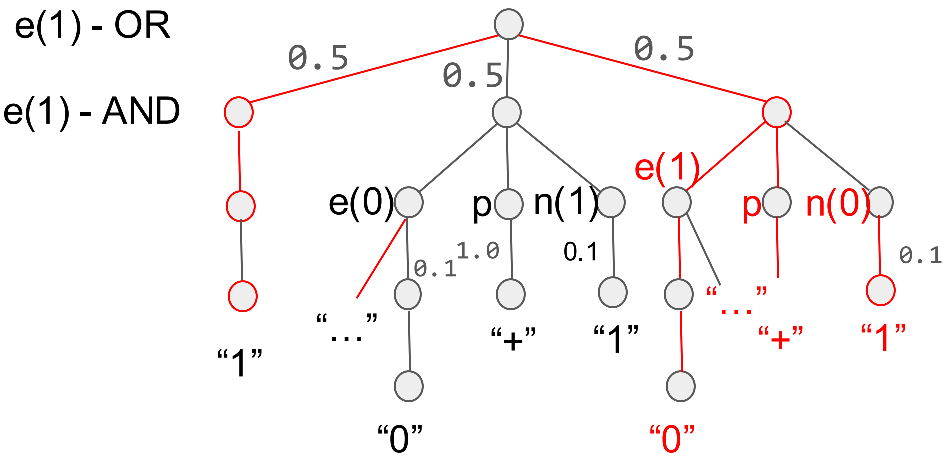
```
0.5 :: e(N) --> n(N).
0.5 :: e(N) --> e(N1), p, n(N2),
                {N is N1 + N2}.
1.0 :: p    --> [“+”].

0.1 :: n(0) --> [“0”].
0.1 :: n(1) --> [“1”].
        …
0.1 :: n(9) --> [“9”].
```

The use of logic: Proof

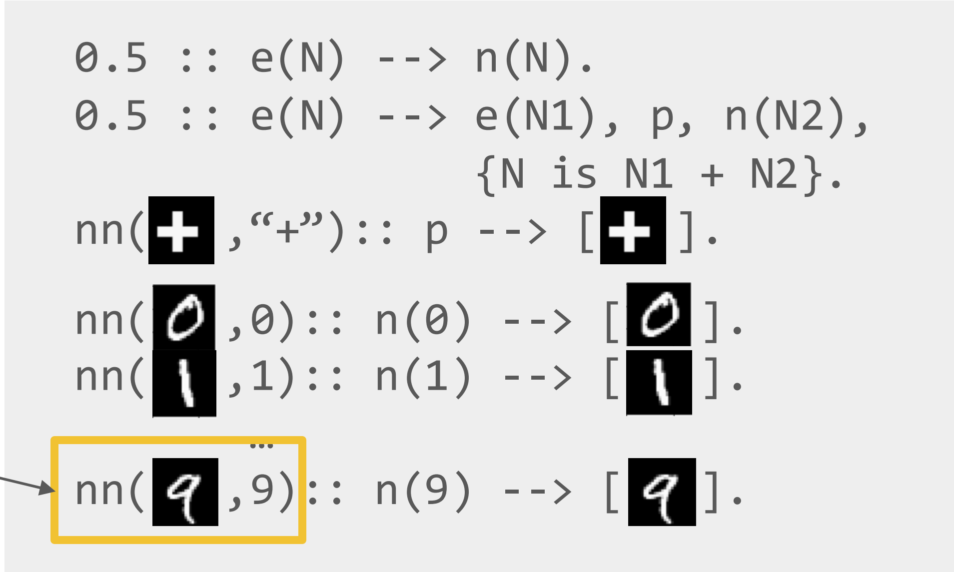
Stochastic Definite Clause Grammars

```
Parse ["0", "+", "1"]
Query e(1)
```



The use of logic: Proof

DeepStochLog

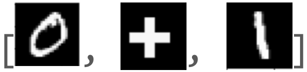


neural
rule

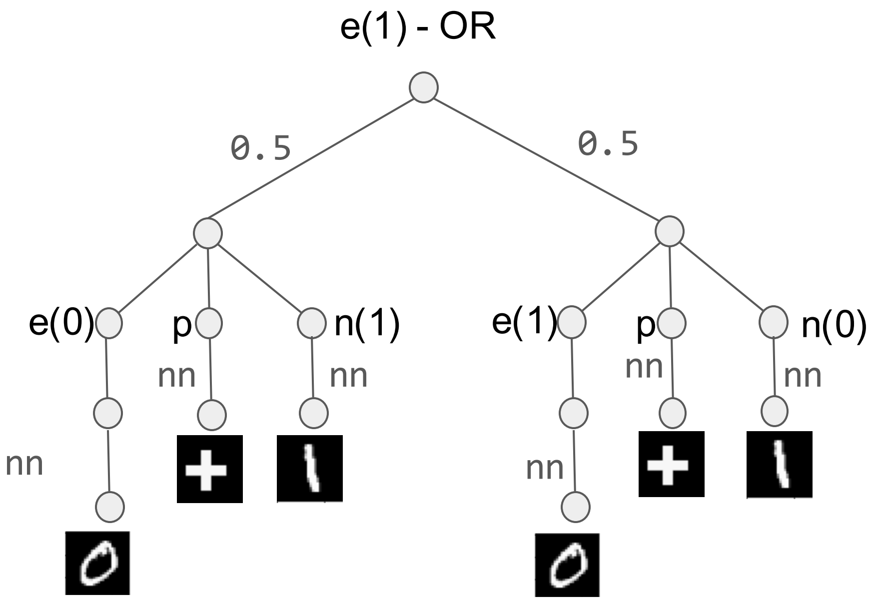
Winters, AAAI 2022

The use of logic: Proof

Stochastic Definite Clause Grammars



```
Parse [ , , ]
Query e(1)
```



The use of logic: Model

Markov Logic Networks

$$p(G,L,S,Z,W) = \frac{1}{Z}\exp(\beta_1\phi_1(G,L,S,Z,W) + \beta_2\phi_2(G,L,S,Z,W) + \ldots)$$

Probability of a model

weighted satisfaction of logical rules

(e.g. `2.75:: L, G -> Z`)

The use of logic: Model

Relational Neural Machines

Add neural-unary factors to MLN

conditioning on subsymbols

$p(G, L, S, Z, W \mid$  $) = \frac{1}{Z} \exp(\beta_1 \phi_1(G, L, S, Z, W) + \beta_2 \phi_2(G, L, S, Z, W) + \ldots$

$+ NN_G($  $) + NN_L($  $) + \ldots)$

neural unary factors

Marra et al, ECML 2019
Marra et al., ECAI 2020

The use of logic: Model

Relational Neural Machines

```
0.5 :: e(N) --> n(N).
0.5 :: e(N) --> e(N1), p, n(N2),
                {N is N1 + N2}.
nn( ,"+"):: p --> [ ].

nn( ,0):: n(0) --> [ ].
nn( ,1):: n(1) --> [ ].
...
nn( ,9):: n(9) --> [ ].
```

conditioning on subsymbols

$$p(G,L,S,Z,W \mid$$  $$) = \frac{1}{Z}\exp(\beta_1\phi_1(G,L,S,Z,W) + \beta_2\phi_2(G,L,S,Z,W) + \dots$$

$$+ NN_G($$  $$) + NN_L($$  $$) + \dots)$$

neural unary factors

Marra et al, ECML 2019
Marra et al., ECAI 2020

StarAI as a recipe for NeSy
The StarAI reparameterization viewpoint

StarAI

```
p(symbols; params)
```

NeSy

```
p(symbols; neural nets)
```

StarAI as a recipe for NeSy
StarAI as a layer

StarAI

`params` → Probabilistic Formal Model → `symbols`

NeSy

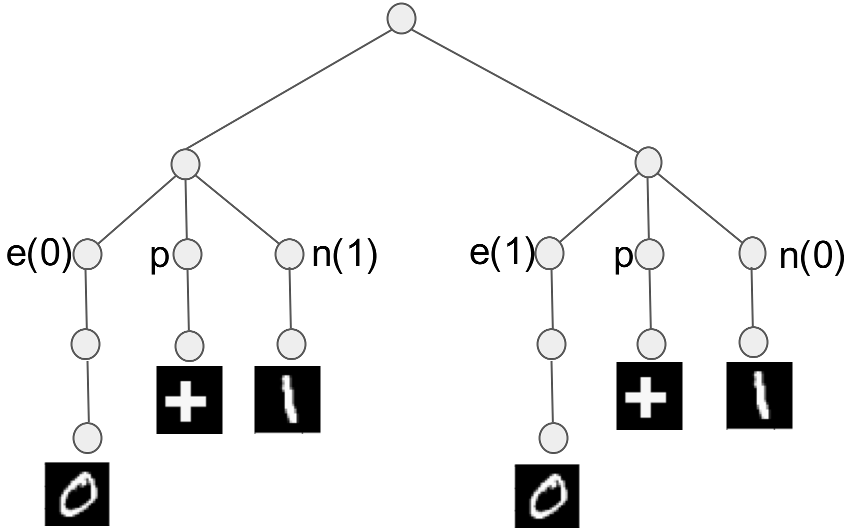
`Neural Nets` → Probabilistic Formal Model → `symbols`

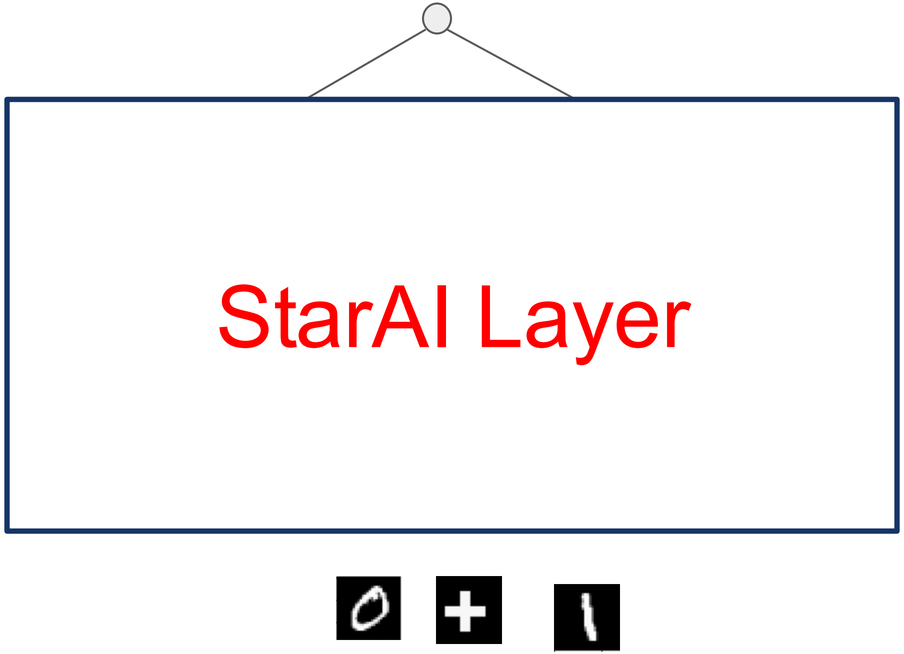
StarAI Layer

Ahmed et al, 2022

Logic as a layer

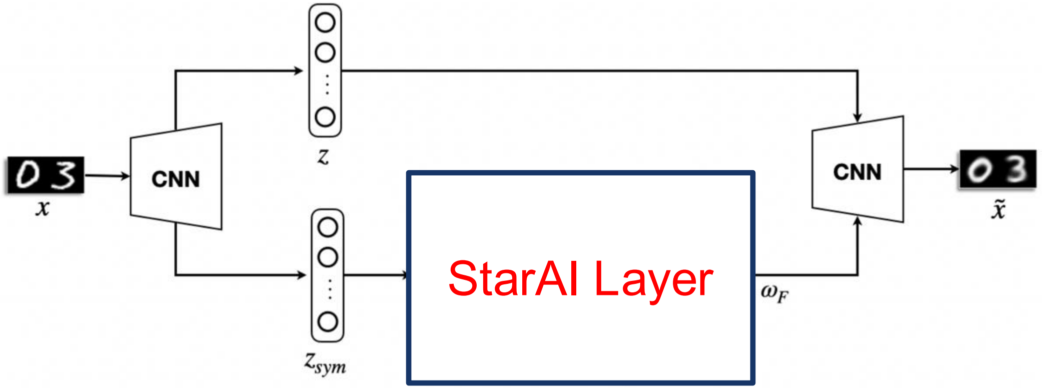
StarAI as a layer





DeepProbLog, Manhaeve et al, NeurIPS 2018
DeepStochLog, Winters et al, AAAI 2022
DeepSeaProbLog, De Smet et al, UAI 2023

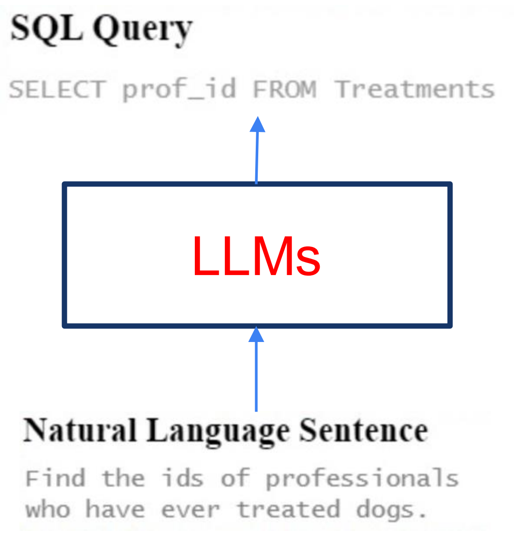
Logic as a layer
StarAI layers in conditional VAE

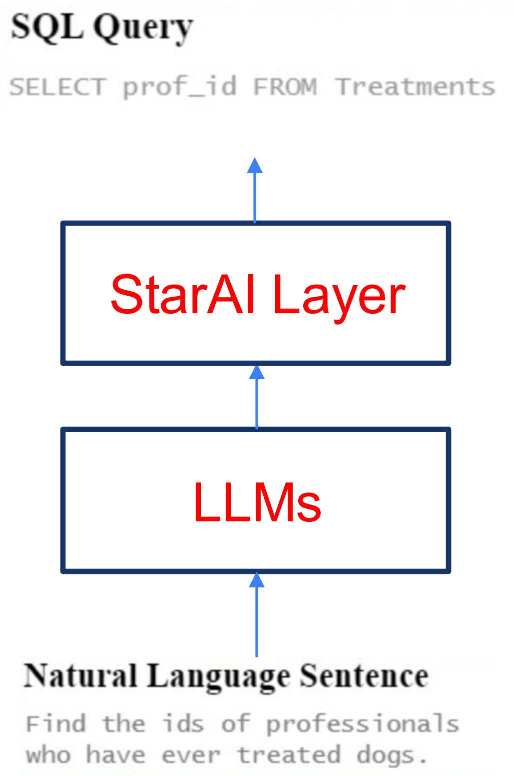


Zero shot generaliation
by programming VAEs

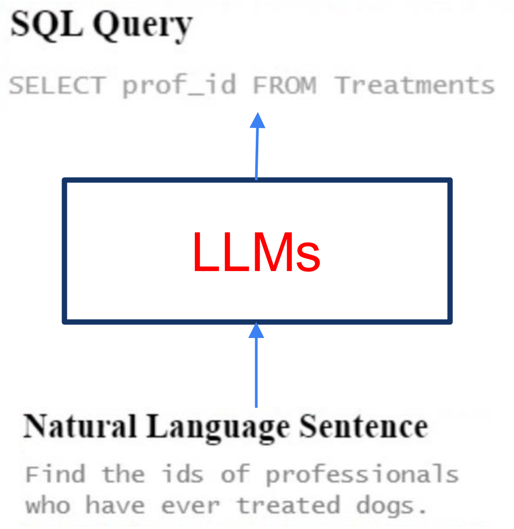
Misino et al, NeurIPS 2022

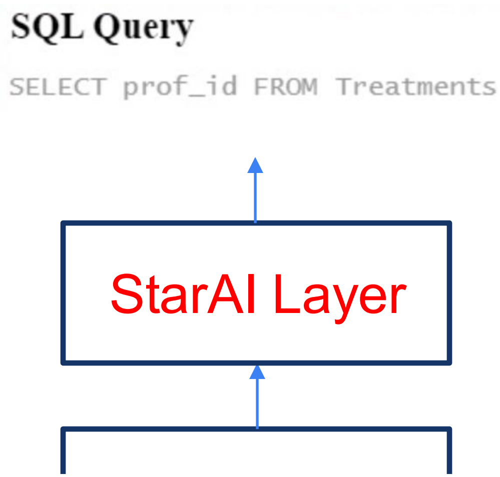
Logic as a layer





Logic as a layer
StarAI layers in conditional VAE

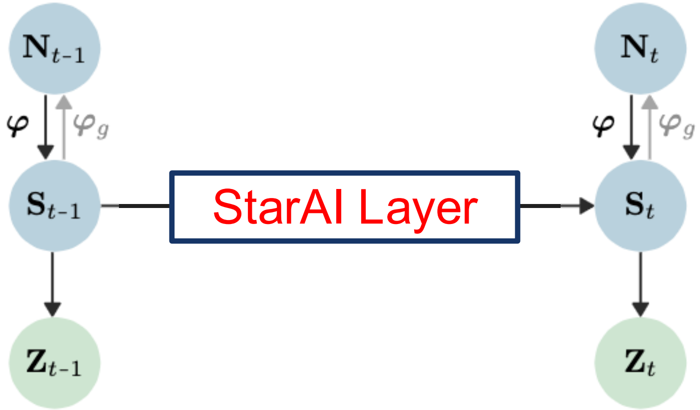




		Validity%	Exact Matching %
Smaller Models (Millions Params.)	T5-small	53.9	41.1
	T5-small+CFGs	88.8	67.1
	Ours (T5-small+DCGs)	**100.0**	75.6
Larger Models (B/Trillions Params.)	DAIL-SQL (GPT-4)	99.2	88.8
	DIN-SQL (GPT-4)	99.2	78.7
	Graphix-T5 (T5-3B+PICARD)	99.6	**91.9**

Logic as a layer

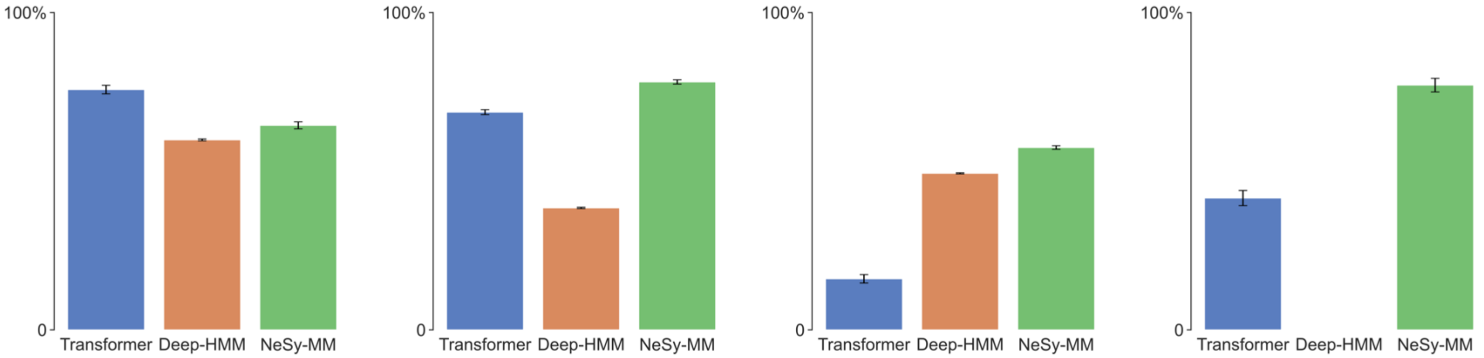
NeSyMMs

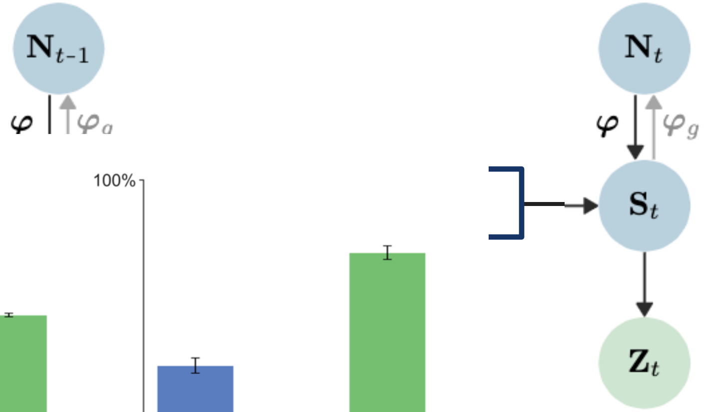


De Smet et al, AAAI 2025

Logic as a layer

NeSyMMs





10x10 → 15x15

10 steps → 20 steps → 20 steps

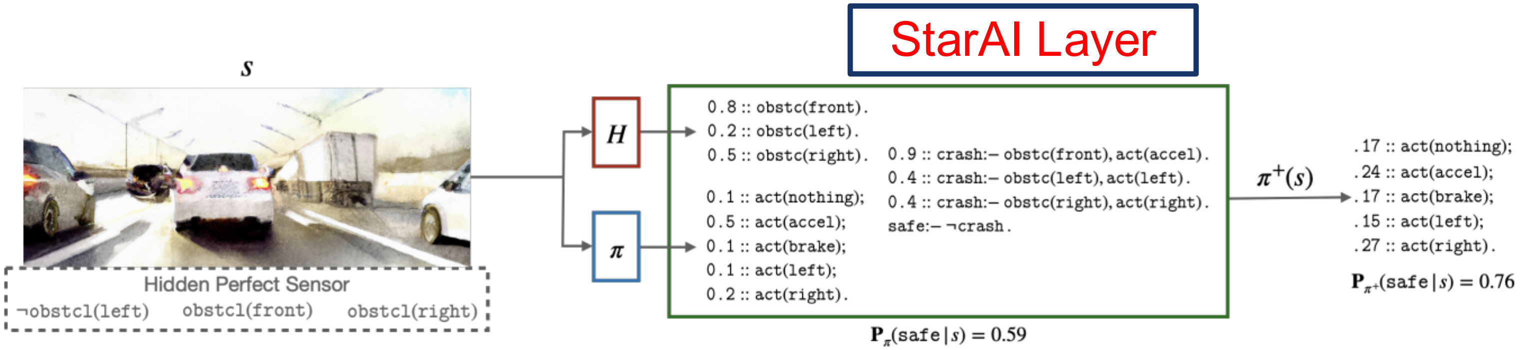
1 enemy → 2 enemies → 2 enemies → 2 enemies

Out of distribution

De Smet et al, AAAI 2025

Logic as a layer

Logic layers in policy gradient

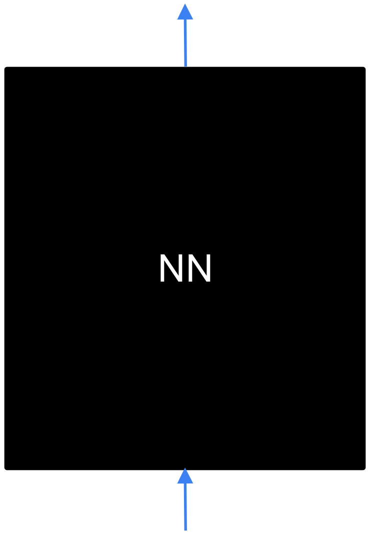


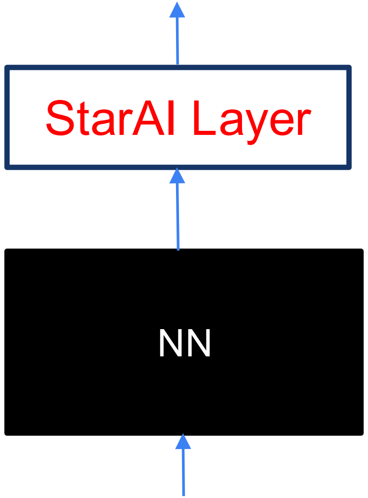
Probabilistic Logic Shields for Safe Reinforcement Learning

Yang et al, IJCAI 2023
Debot et al, AAAI 2025

Logic as a layer

StarAI layer as an interpretable layer





Concept-Based Interpretable Model

Barbiero et al, ICML 2023
Debot et al, NeurIPS 2024
Dominici et al, ICLR 2025

StarAI as a recipe for NeSy

- StarAI has already studied sound semantics for learning and reasoning
- StarAI can be used as a starting point for NeSy
- StarAI layers can give a neurosymbolic flavour to existing neural approaches

What should NeSy be about?

Enthusiast: *Integrating knowledge into neural networks (best of both worlds)*

Critic: *Knowledge bottleneck: obtaining, formalizing and maintaining symbolic human knowledge is hard.*

Three aspects of knowledge in NeSy

Source

Format

Function

The **source** of knowledge: **human**

Enthusiast: NeSy can integrate human knowledge

Critic: You are constraining what the machine can learn by the human

- Integrating human knowledge is (to very different degrees) part of all AI
- Human knowledge can be:
    - Logic rules, Inference Rules -> as in Symbolic AI
    - Supervision / Data preprocessing / Inductive Biases / Loss functions -> as in supervised learning
    - Content -> as in self-supervised
    - Rewards/Environments -> as in reinforcement learning

- Integrating knowledge is everywhere in AI
    - Not a prerogative of NeSy
    - But in other ”terms” very well accepted in ML/AI

The **format** of knowledge: **symbolic**

Critic: Formalizing human knowledge in symbolic way is hard and error-prone

Enthusiast: Yet, you can provide guarantees and, therefore, trust the AI model

Different issue:

- Formalization is hard (no matter the source)
- But the same effort is repaid in trust
- The real question is: how much knowledge should I really encode in a formal way?

The **function** of knowledge: **prescriptive**

Enthusiast: If I know how to do addition, you should not learn it from data

Critic: Maybe there is a better way to do addition

Knowledge: prescriptive (**how**)

- addition(X,Y,Z) :- digit(X,N1), digit(Y,N2), Z is N1 + N2
- More general: without the knowledge you can't solve the task at all
    - Complete
    - Consistent

The **function** of knowledge: <u>towards **descriptive**</u>

Knowledge should be used for expressing what we care about (**what**)

E.g.:

- **Constraints** that MUST be satisfied;
  - Aka: I want addition to be done only in that way
- Values:
  - Learn whatever you want as far as this property is guaranteed

- ***Knowledge*** is limited to what really matters to the human;
- ***Knowledge*** is part of the definition of the task itself; the user should still specify

What should NeSy be about?

Most of AI use (forms of) human knowledge;

Symbolic format is hard but allows to get guarantees

We should move (in all AI) to **descriptive** knowledge (the **what** we want, constraints) as much as possible

In **NeSy**, the role of knowledge should be:

- **not** to **replace learning**,
- to **shape the landscape** in which learning occurs

Challenges

- The role of human specification is under-looked
    - Alignment in semantics human-machine
    - E.g. reasoning shortcuts / identifiability issues
- Formalization of ”what” is not necessarily easier
    - Hard “what”: fairness, privacy, ethical behaviour
- Not a new way of doing NeSy, but a reframe of its scope

Thank you!