Neurosymbolic AI

INTEGRATING SYMBOLIC AND SUBSYMBOLIC KNOWLEDGE

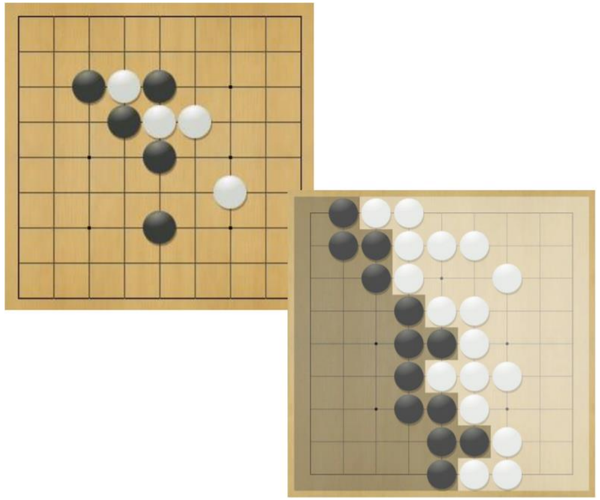
SPEAKER: VALERIO DE CARO
DIPARTIMENTO DI INFORMATICA – UNIVERSITÀ DI PISA

AN EXAMPLE ON A GRAND CHALLENGE OF AI

"Thinking fast and slow"

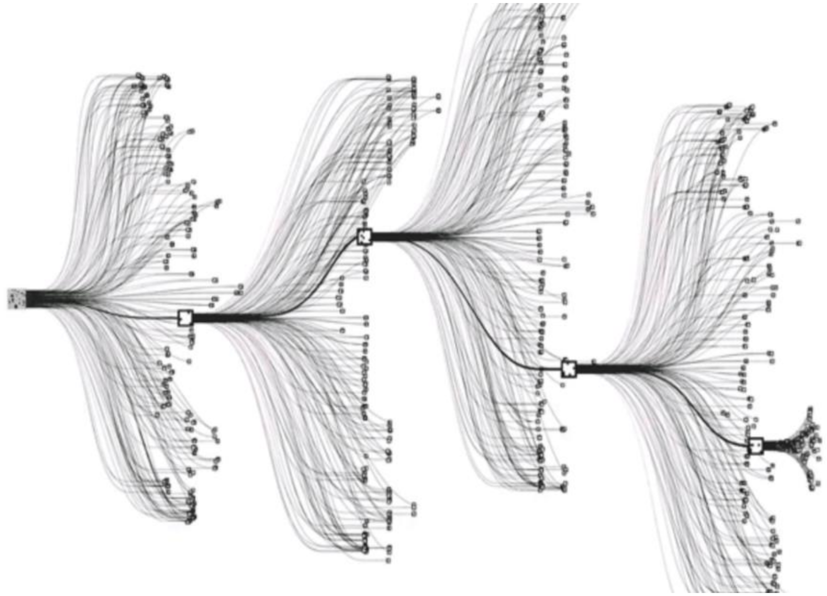
The game of Go

- Usually played in 19×19, also 13×13 and 9×9 boards;
- Black and white place down stones alternately
- Surrounded stones are captured and removed
- The player with more territory wins



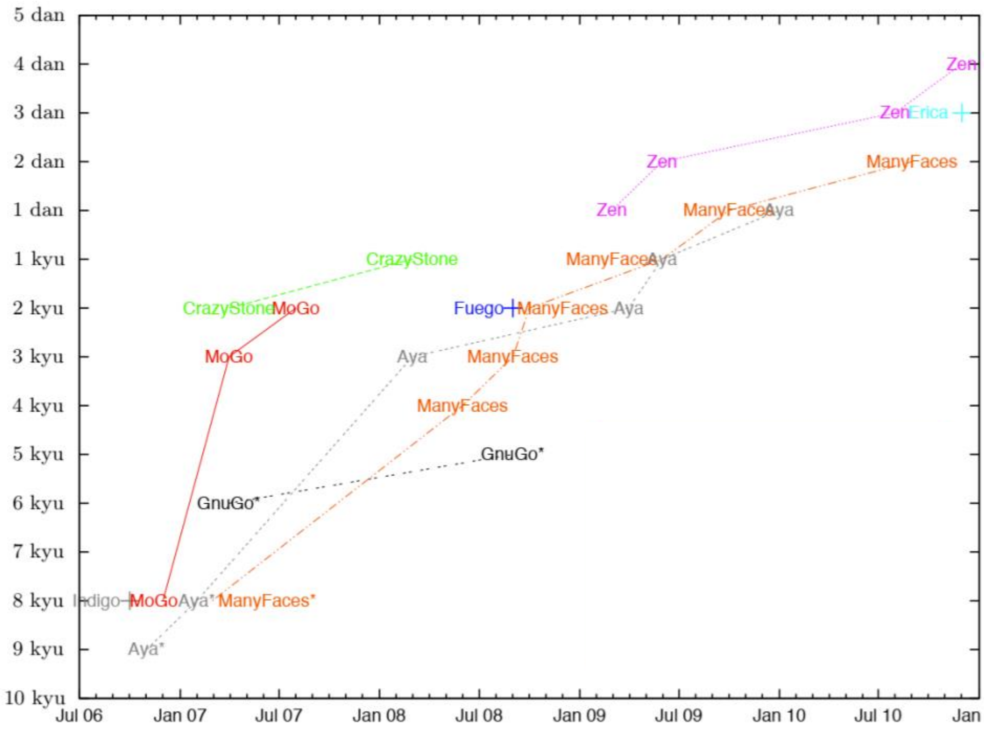
White wins.

Credits: Davide Bacciu



Making a good synthetic player is (was) hard due do the combinatorial state explosion (3^{361} states)

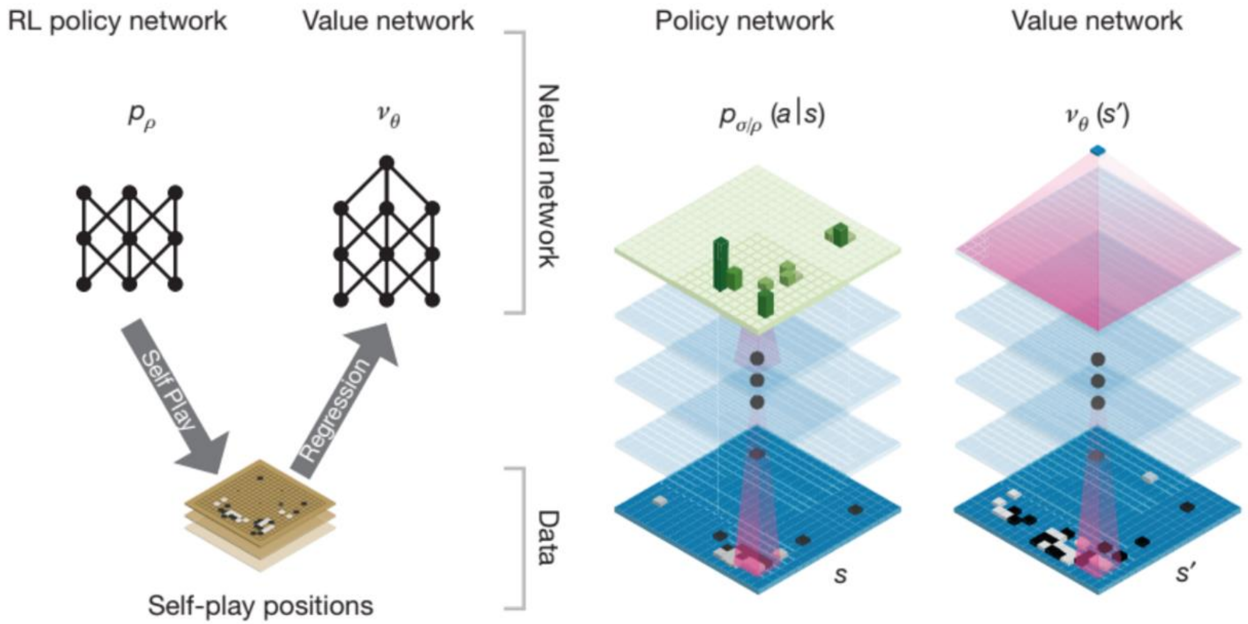
Classic AI (symbolic, a.k.a. ***thinking slow***) has a long-standing history of attempts in creating such player



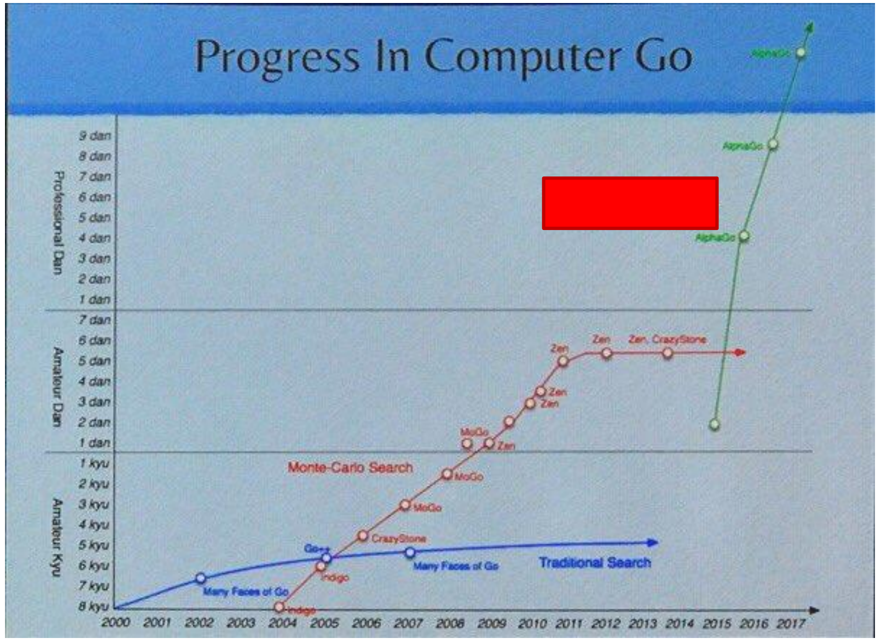
AN EXAMPLE OF GRAND FAILURES IN AI

"Thinking fast and slow"

Credits: Davide Bacciu



AlphaZero, subsymbolic, a.k.a.
thinking fast



Why should we need to "think slow"?

"THE VICTORY OF NEURAL OVER HANDCRAFTED HEURISTICS"

"Thinking fast and slow"

DIFFICULTY OF IDENTIFYING SYMBOLS OVER PATTERNS

Issues with “Thinking fast”

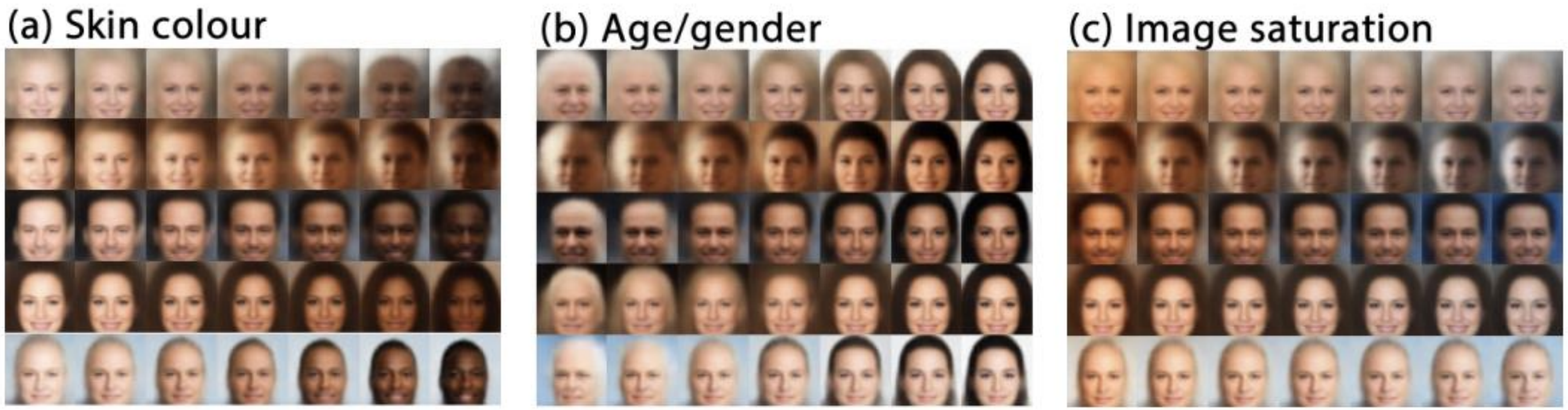


Figure 4: **Latent factors learnt by β-VAE on celebA:** traversal of individual latents demonstrates that β-VAE discovered in an unsupervised manner factors that encode skin colour, transition from an elderly male to younger female, and image saturation.

Research in disentangled representations show the need of symbols emergency from the subsymbolic space

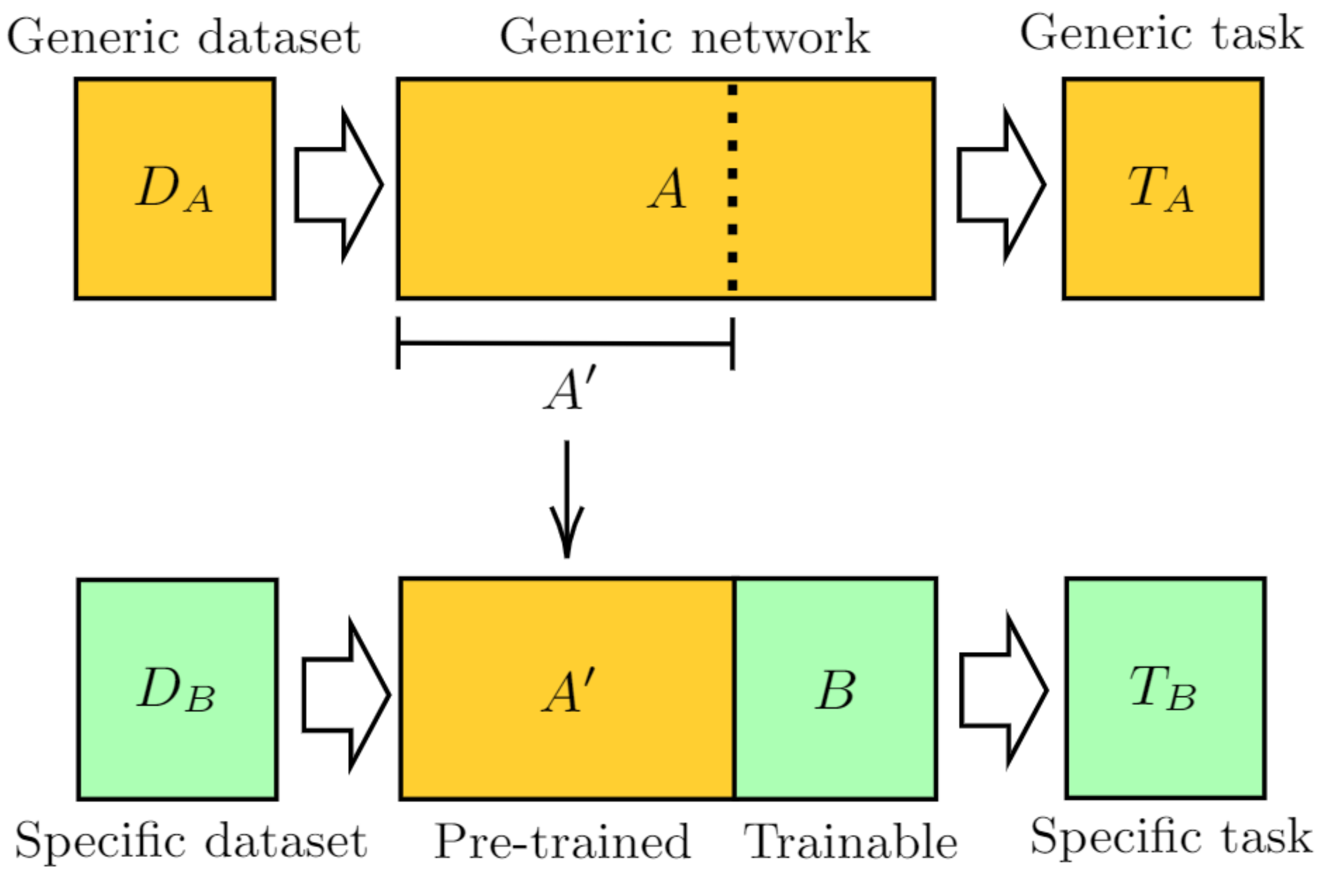
“It got dark outside. What happened as a result?”

a) Snowflakes began to fall from the sky.
b) The moon became visible in the sky.

Without any sample about such association, a ML model is not capable of answering this simple question.

COMMONSENSE REASONING

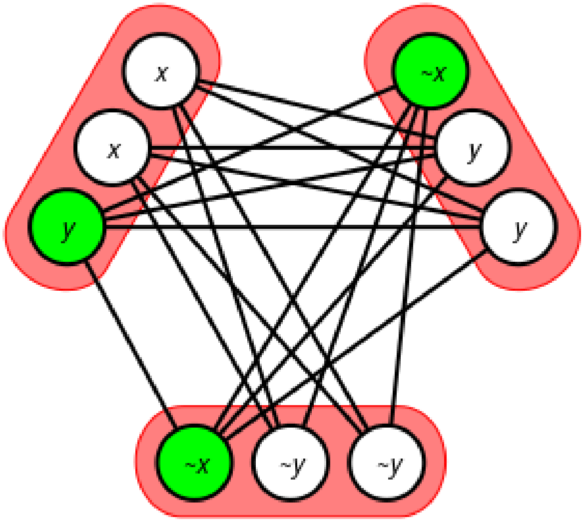
Issues with “Thinking fast”

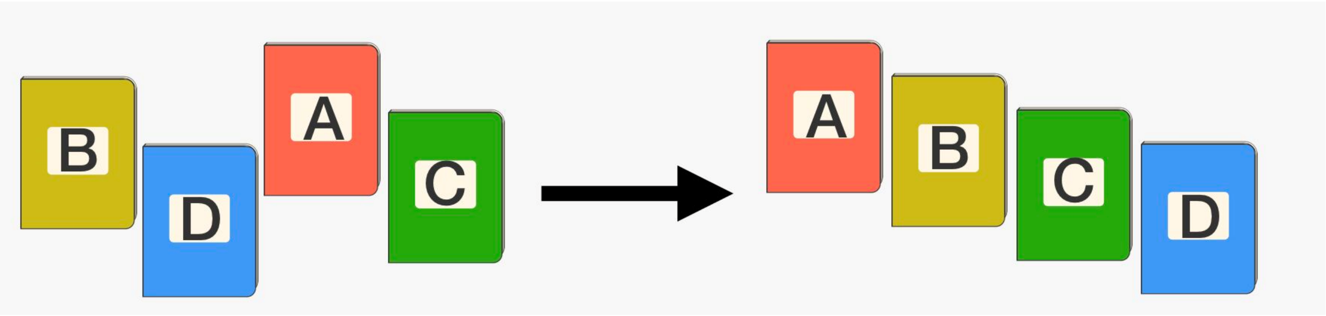


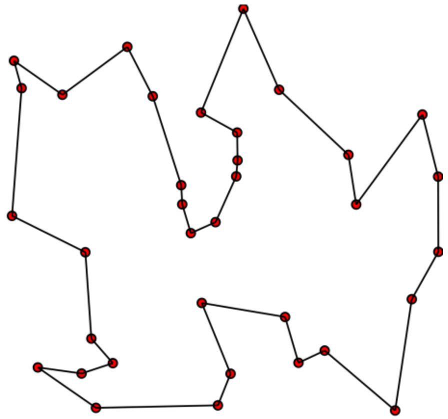
Transfer learning shows the need of compositionality in ML models

COMPOSITIONALITY

Issues with "Thinking fast"







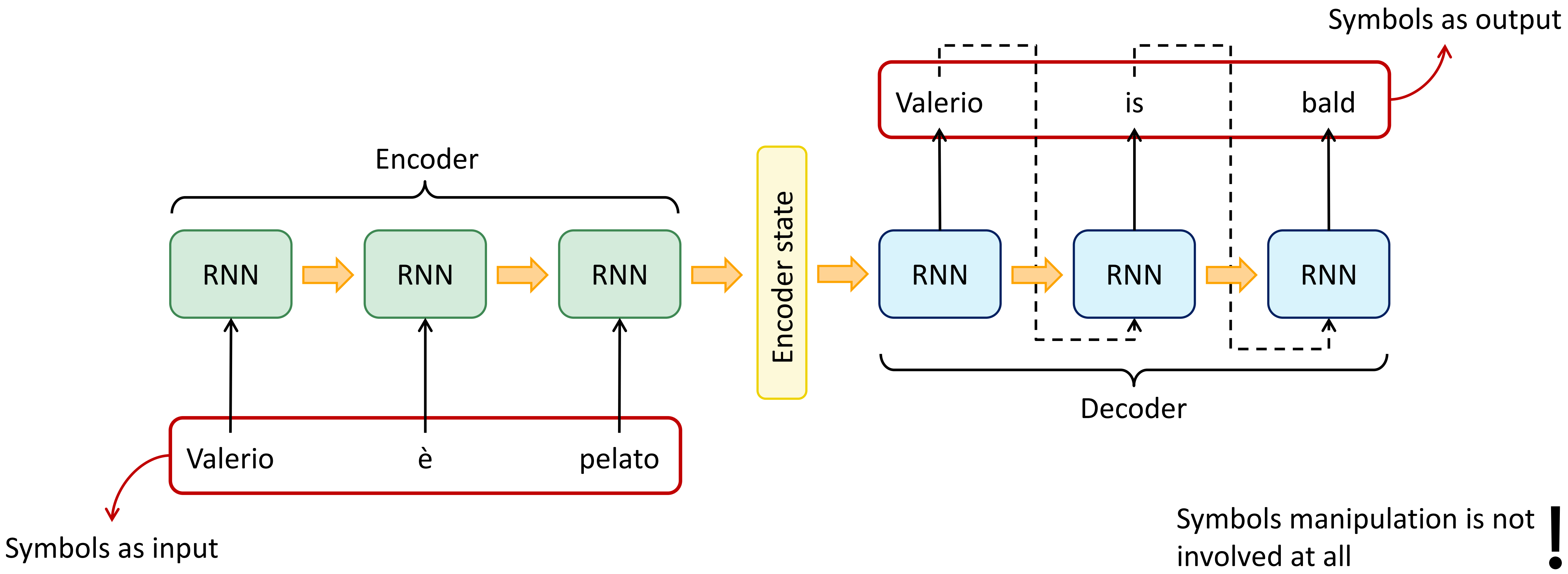
Combinatorial problems are difficult to be solved with the data-driven approach of ML.

COMBINATORIAL REASONING

Issues with "Thinking fast"

SIX DEGREES OF INTEGRATION ACCORDING TO H. KAUTZ

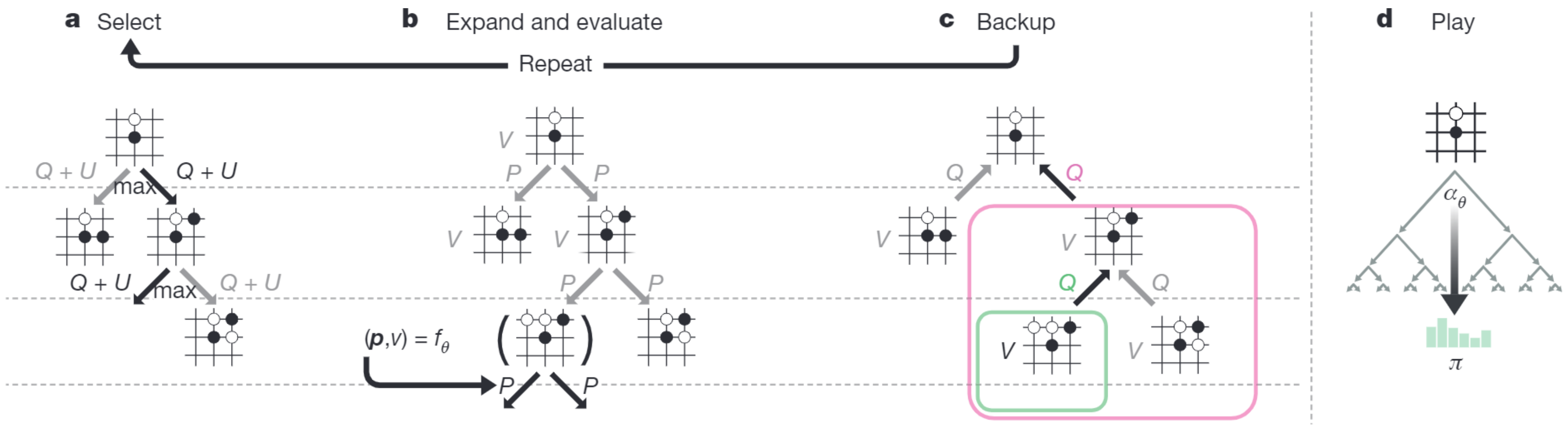
Taxonomy of neurosymbolic integration



ENCODING SYMBOLS INTO A LATENT SPACE, AND DECODING SYMBOLS FROM IT

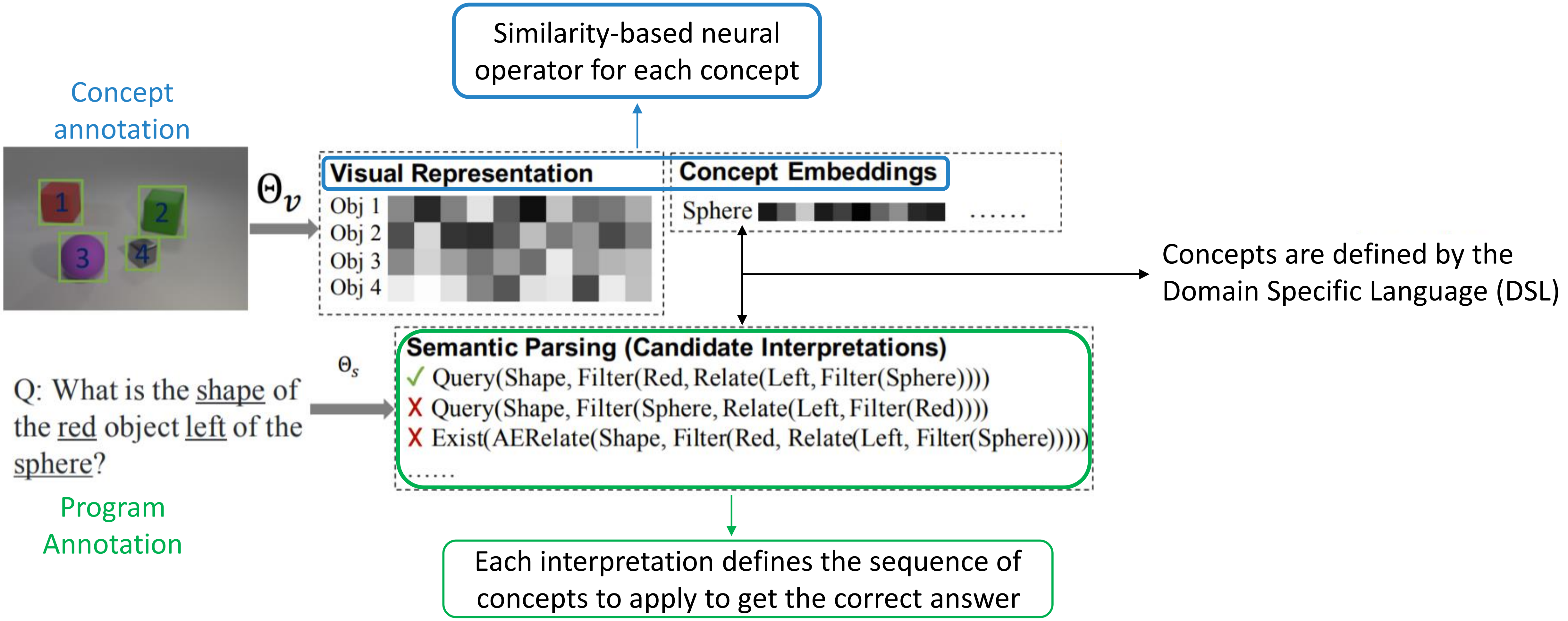
Type 1: Encoder-Decoder in NLP

Monte Carlo Tree-Search solves the subproblem of generating episodes based on the current state θ of the model



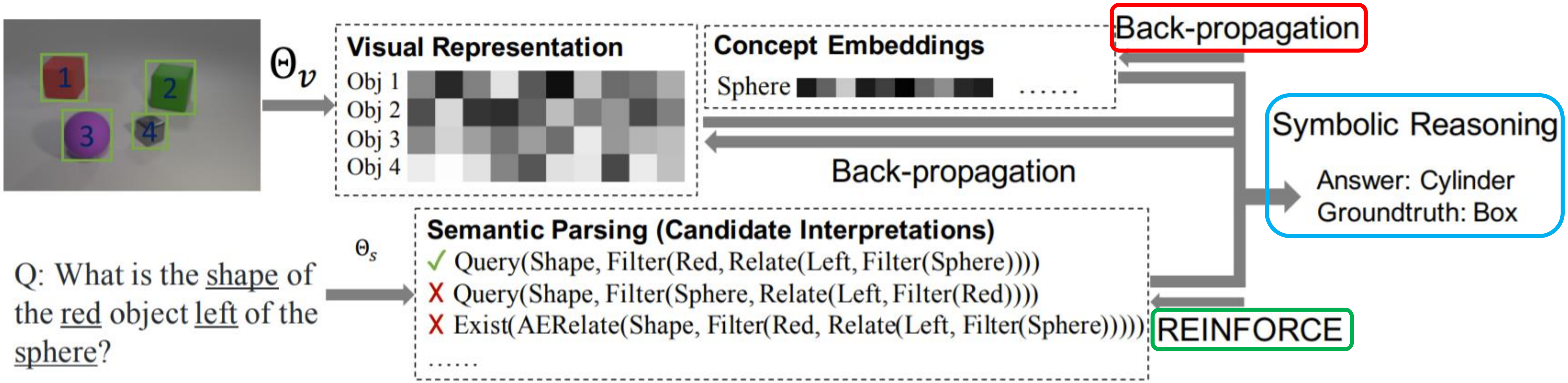
AN ACTOR-CRITIC, GAME-AGNOSTIC RL MODEL WHICH ACHIEVED STATE-OF-THE-ART PERFORMANCE IN GO

Type 2: AlphaZero



VISUAL PERCEPTION + QUASI-SYMBOLIC PROGRAM EXECUTION ON TOP OF A DSL FOR VQA

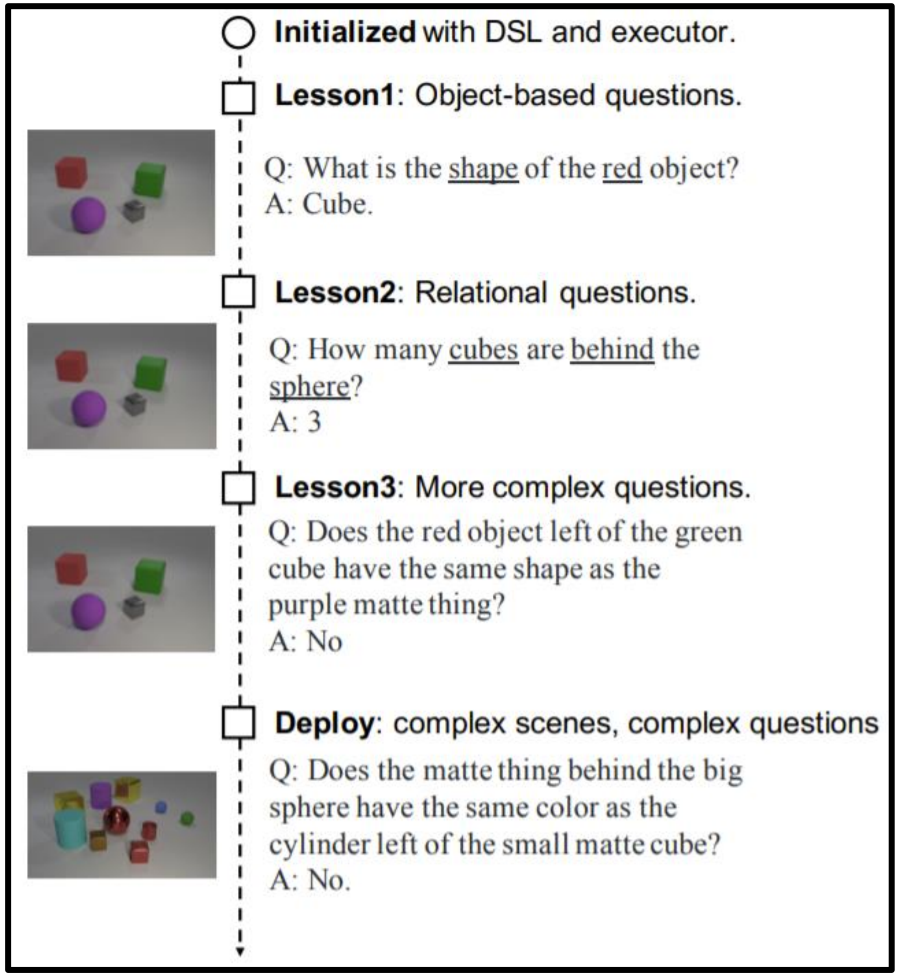
Type 3.1: Neurosymbolic Concept Learner

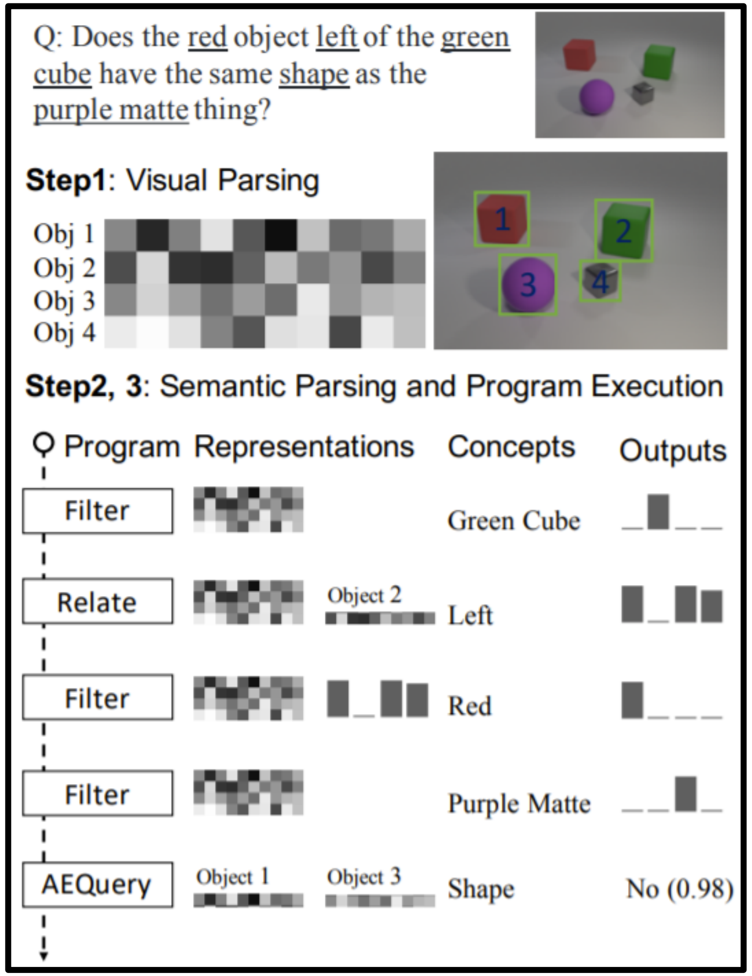


1. Gradient-based maximization → $\mathcal{L} = \langle \Pr[A = Executor(Perception(S; \Theta_v), P)] \rangle_P$
2. Gradient of Θ_v → $\nabla_{\Theta_v} \mathcal{L}$
3. Gradient of Θ_s → $\langle r \cdot \log \Pr[P = SemanticParse(Q; \Theta_p)] \rangle$ where $r = 1$ if the answer is correct, 0 otherwise

LEARNING AS A LOOP BETWEEN THE CONCEPT ANNOTATOR AND THE PROGRAM ANNOTATOR

Type 3.1: Neurosymbolic Concept Learner

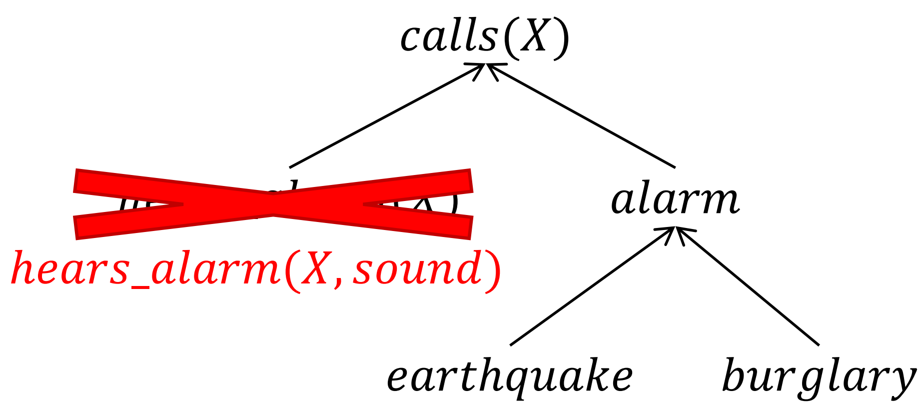




CURRICULUM CONCEPT LEARNING AND EXAMPLE OF EXECUTION

Type 3.1: Neurosymbolic Concept Learner

BAYESIAN NETWORK



$P(earthquake) = 0.2$

$P(burglary) = 0.1$

$hears\_alarm(X, sound) = allow\_sound(X, sound)$

DEEPPROBLOG PROGRAM

Facts

$0.1 :: burglary$

$0.2 :: earthquake$

$nn(device, (X, sound), [T, F]) :: allow\_sound(X, sound)$

Rules

$alarm :- earthquake$

$alarm :- burglary$

$calls(X) :- hears\_alarm(X), alarm$

Learning from entailment: $\arg\min_{\boldsymbol{x}} \frac{1}{|Q|} \sum_{(q,p)\in Q} \mathrm{L}(\mathrm{P}_{\mathcal{X}=\boldsymbol{x}}(q), p)$

AN EXTENSION OF PROBLOG TO FORMULAS WITH NEURAL OPERATORS

Type 3.2: DeepProblog

Definition (*t-norm*). A map $\mathrm{T}: [0,1]^2 \to [0,1]$ is a *triangular norm* if and only if $\forall x, y, z \in [0,1]$ the following properties are satisfied:

i. $\mathrm{T}(x, y) = \mathrm{T}(y, x)$ (symmetry);
ii. $\mathrm{T}\big(x, \mathrm{T}(y, z)\big) = \mathrm{T}(\mathrm{T}(x, y), z)$ (associativity);
iii. $(x \leq \bar{x}) \wedge (y \leq \bar{y}) \Rightarrow \mathrm{T}(x, y) \leq \mathrm{T}(\bar{x}, \bar{y})$ (monotonicity);
iv. $\mathrm{T}(x, 1) = x$ (one identity).

Łukasiewicz t-norm: $\mathrm{T}_{Ł}(x, y) = \max\{x + y - 1, 0\}$

Łukasiewicz propositional logic $[0,1]_{Ł} = \{[0,1],\ 0,\ 1,\ \neg,\ \otimes,\ \oplus,\ \Rightarrow\}$ is defined as

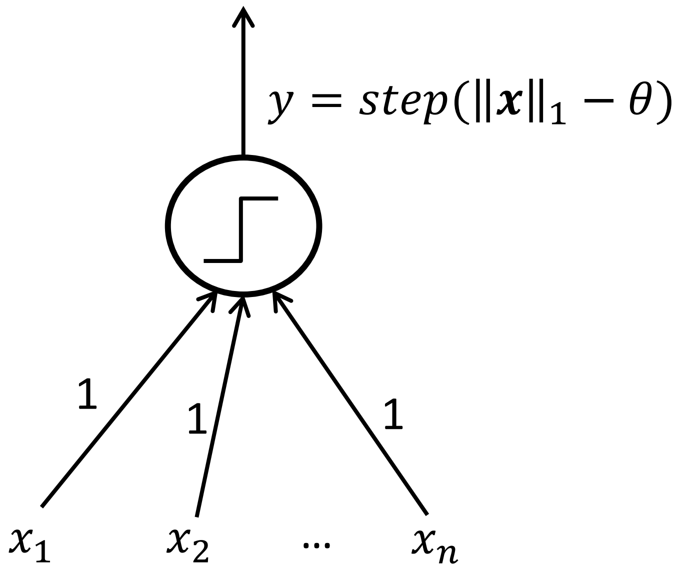
i. $\neg x = 1 - x$ (*negation*);
ii. $x \otimes y = \max\{0, x + y - 1\}$ (*strong conjunction*);
iii. $x \oplus y = \min\{1, x + y\}$ (*strong disjunction*);
iv. $x \Rightarrow y = \min\{1, 1 - x + y\}$ (*implication*).

Intro to T4 and T5: T-Norms and Real Logics

Type 4: Logical Neural Networks

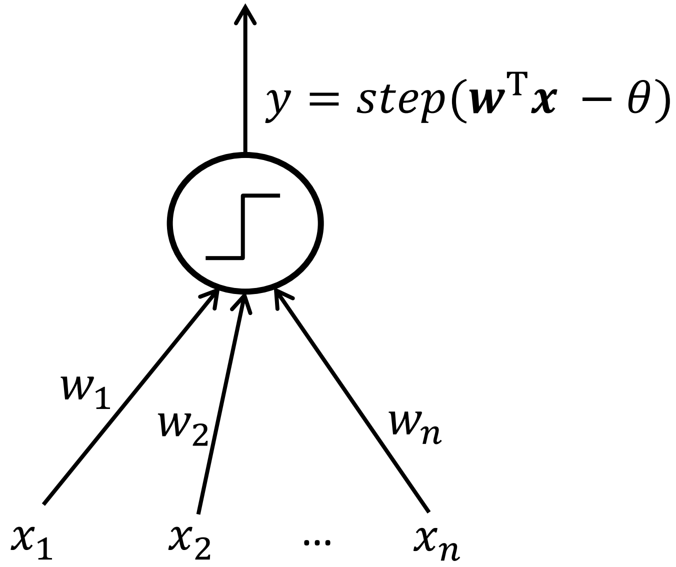
1-TO-1 CORRESPONDENCE BETWEEN NEURONS AND ELEMENTS OF LOGICAL FORMULAS (LOCALIST APPROACH)

McCulloch-Pitts neuron (1943)



Acts as a logical AND

Rosenblatt's Perceptron (1958)

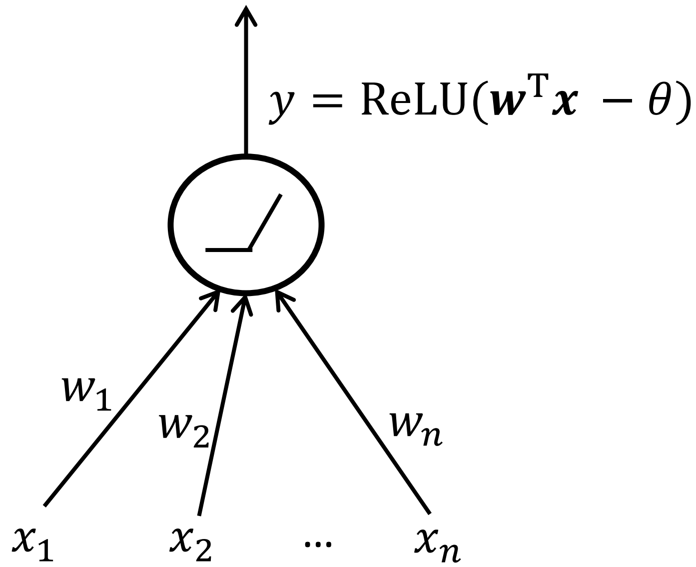


Acts as a logical AND **within the region**

$$\|\boldsymbol{w}\|_1 \geq \theta + 1$$

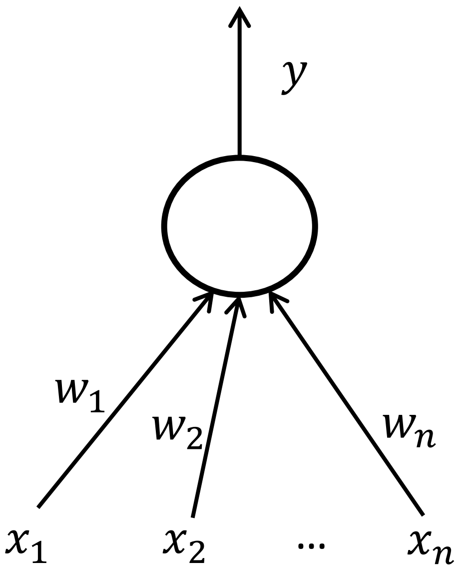
$$\forall i, \sum_j w_j - w_i \leq \theta$$

Differentiable neuron



Lost the "AND behaviour"

Constrained logical neuron



n-ary weighted Łukasiewicz nonlinear logic

$$y = \begin{cases} {}^{\beta}\left(\bigotimes_{i \in \mathcal{I}} x_i^{\otimes w_i}\right) = f\left(\beta - \sum_{i \in \mathcal{I}} w_i (1 - x_i)\right) \\ {}^{\beta}\left(\bigoplus_{i \in \mathcal{I}} x_i^{\oplus w_i}\right) = f\left(1 - \beta - \sum_{i \in \mathcal{I}} w_i x_i\right) \\ {}^{\beta}\left(x^{\otimes w_x} \Rightarrow y^{\oplus w_y}\right) = f(1 - \beta + w_x(1 - x) + w_y y) \end{cases}$$

where

$f : \mathbb{R} \to [0,1]$, e.g. $f(x) = \max\{0, \min\{1, x\}\}$
$\beta \geq 0$
$w_i \geq 0 \; \forall i$

THE CONSTRAINED LOGICAL NEURON ON TOP OF AN N-ARY WEIGHTED NONLINEAR LOGIC

Type 4: Logical Neural Network

Each neuron is going to output an **upper and a lower bound on the truth value** of their corresponding subformulae and propositions, s.t. the following states are drawn

Bounds	Unknown	True	False	Contradiction
Upper Lower	$[\alpha, 1]$ $[0, 1-\alpha]$	$[\alpha, 1]$ $[\alpha, 1]$	$[0, 1-\alpha]$ $[0, 1-\alpha]$	Lower > Upper

Choosing $0 < \alpha < 0.5$ allows us to live in the **Open-World Assumption**

Learning via "contradiction minimization", while enforcing constraints as for the Rosenblatt's perceptron (-ish)

UPPER AND LOWER BOUNDS ON CONFIDENCE OF THE INFERENTIAL PROCESS

Type 4: Logical Neural Network

We start from a first-order language (FOL) $\mathcal{L} = (\mathcal{C}, \mathcal{F}, \mathcal{P})$.

Definition (*Grounding*). A *grounding* $\mathcal{G}$ for a first order language $\mathcal{L}$ is a function from the signature of $\mathcal{L}$ to the real numbers that satisfies the following condition:

i. $\mathcal{G}(c) \in \mathbb{R}^n$ for every constant symbol $c \in \mathcal{C}$;
ii. $\mathcal{G}(f) \in \mathbb{R}^{n \cdot \alpha(f)} \rightarrow \mathbb{R}^n$ for every functional symbol $f \in \mathcal{F}$;
iii. $\mathcal{G}(P) \in \mathbb{R}^{n \cdot \alpha(P)} \rightarrow [0,1]$ for every predicate symbol $P \in \mathcal{P}$.

We inductively extend it to all the closed terms and clauses as follows

$$\mathcal{G}\big(f(t_1, \dots, t_m)\big) = \mathcal{G}(f)(\mathcal{G}(t_1), \dots, \mathcal{G}(t_m))$$
$$\mathcal{G}\big(P(t_1, \dots, t_m)\big) = \mathcal{G}(P)(\mathcal{G}(t_1), \dots, \mathcal{G}(t_m))$$
$$\mathcal{G}\big(\neg P(t_1, \dots, t_m)\big) = 1 - \mathcal{G}(P)(\mathcal{G}(t_1), \dots, \mathcal{G}(t_m))$$
$$\mathcal{G}(\phi_1 \vee \dots \vee \phi_k) = \mu(\mathcal{G}(\phi_1), \dots, \mathcal{G}(\phi_k))$$

μ is an s-norm operator (the dual of the t-norm)

REAL LOGICS BUILT ON TOP OF FIRST ORDER LOGIC (DISTRIBUTED APPROACH)

Type 5: Logic Tensor Networks

Let $\mathcal{L} = (\mathcal{C}, \mathcal{F}, \mathcal{P})$ be a language such that:

- $\mathcal{C} = \{o_1, o_2, o_3\}$ is a set of documents defined on a finite dictionary $D = \{w_1, \dots, w_n\}$ of n words,
- $\mathcal{F} = \{concat(x, y)\}$, where $concat(x, y)$ denotes the concatenation of two documents and
- $\mathcal{P} = \{sim(x, y)\}$, where $sim(x, y)$ is supposed to be $true$ when document x is deemed to be similar to document y.

The corresponding grounding $\mathcal{G}$ can be defined as:

- $\mathcal{G}(o_i) = \langle n_{w_1}^{o_i}, \dots, n_{w_n}^{o_i} \rangle$ is a bag-of-words representation of o_i;
- $\mathcal{G}(concat)(\mathbf{u}, \mathbf{v}) = \mathbf{u} + \mathbf{v}$
- $\mathcal{G}(sim)(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}$

EXAMPLE OF GROUNDING ON A FIRST ORDER LOGIC

Type 5: Logic Tensor Networks

Type 5: Logic Tensor Network

IMPLEMENTATION OF FUNCTIONS AND PREDICATES AS TENSOR TRANSFORMATIONS

FUNCTIONS

Given a function symbol f of arity m and $\mathbf{v}_1, \dots, \mathbf{v}_\mathrm{m} \in \mathbb{R}^n$ real vectors corresponding to the grounding of m terms,

$$\mathcal{G}(f)(\mathbf{v}_1, \dots, \mathbf{v}_\mathrm{m}) = \mathrm{M}_f \mathbf{v} + N_f$$

where $\mathrm{M}_f \in \mathbb{R}^{n \times mn}$ and $N_f \in \mathbb{R}^n$.

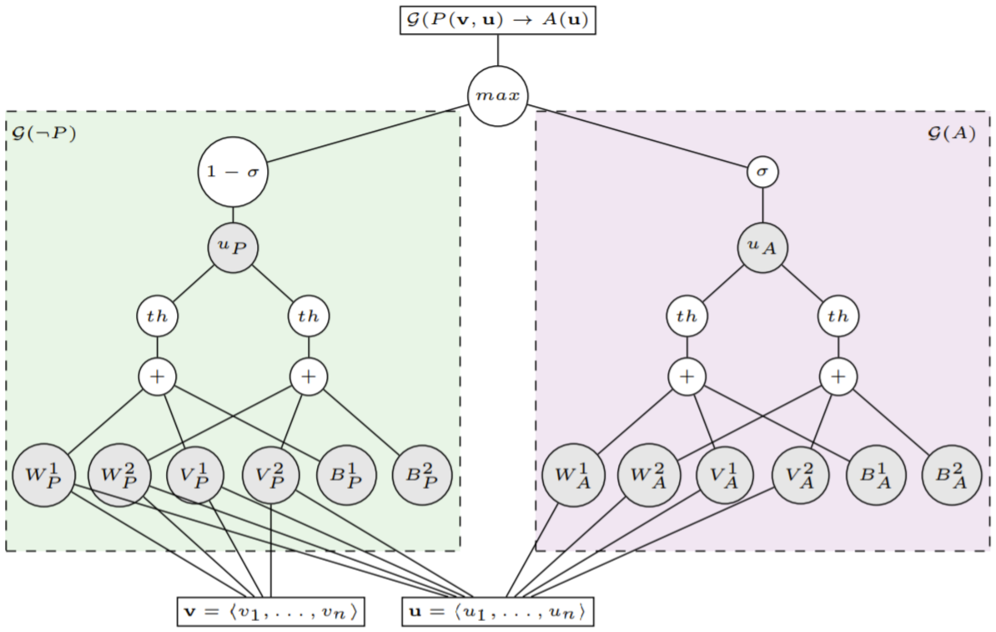
PREDICATES

Given a predicate symbol P of arity m and $\mathbf{v}_1, \dots, \mathbf{v}_\mathrm{m} \in \mathbb{R}^n$ real vectors corresponding to the grounding of m terms,

$$\mathcal{G}(P) = \sigma\left(\mu_P^\mathrm{T} \tanh\left(\mathbf{v}^\mathrm{T} \mathrm{W}_P^{[1:k]} \mathbf{v} + V_P \mathbf{v} + B_P\right)\right)$$

where $\mathrm{W}_P^{[1:k]} \in \mathbb{R}^{mn \times mn \times k}$, $V_P \in \mathbb{R}^{k \times mn}$, $B_P \in \mathbb{R}^k$ and $\mu_P^\mathrm{T} \in \mathbb{R}^k$ (*scoring function*)

EXAMPLE



Definition (*Satisfiability*). Let ϕ be a closed clause in $\mathcal{L}$, $\mathcal{G}$ a grounding, and $v \le w \in [0,1]$. We say that $\mathcal{G}$ *satisfies* ϕ in the confidence interval $[v, w]$, written $\mathcal{G} \vDash_v^w \phi$, $\boxed{\text{if } v \le \mathcal{G}(\phi) \le w.}$

So we can define the loss function as the **minimal distance** between our grounding $\mathcal{G}$ and the points in the interval, i.e.

$$\mathrm{Loss}(\mathcal{G}, \langle [v, w], \phi \rangle) = |x - \mathcal{G}(\phi)|, \qquad v \le x \le w$$

$$\mathcal{G}^* = \arg \min_{\hat{\mathcal{G}} \subseteq \mathcal{G} \in \mathbb{G}} \sum_{\langle [v,w], \phi(\boldsymbol{t}) \rangle \in \mathcal{K}_0} \mathrm{Loss}\,(\mathcal{G}, \langle [v, w], \phi(\boldsymbol{t}) \rangle)$$

Robust w.r.t. inconsistency!

Extension of the partial grounding $\hat{\mathcal{G}}$ over the family $\mathbb{G}$

Finite set of instantiations of the clauses in $\mathcal{L}$

LEARNING AS APPROXIMATE SATISFIABILITY

Type 5: Logic Tensor Network

"Type 6 system should be capable of true symbolic reasoning inside a neural engine. This is what one could refer to as a fully integrated system. [...] *Type 6 system should be capable of combinatorial reasoning, possibly by using an attention schema to achieve it effectively. Recent efforts in this direction include* [*citations*], ***although a fully-fledged Type 6 system for combinatorial reasoning does not exist yet**."*

NEUROSYMBOLIC INTEGRATION ON STEROIDS

Type 6: ...

TIGHTENING THE INTEGRATION BETWEEN SYMBOLIC AND SUBSYMBOLIC KNOWLEDGE

Summary of the taxonomy

- **Type 1**
  - Symbols as input and symbols as output, no symbols manipulation.
- **Type 2**
  - Solving of a subproblem with a symbolic approach
- **Type 3**
  - Concatenation of subsymbolic and symbolic subsystems
- **Type 4**
  - Mapping a subsymbolic system onto a symbolic one (localist)
- **Type 5**
  - Mapping a symbolic system onto a subsymbolic one (distributed)
- **Type 6**
  - Full integration between symbols and subsymbols

- We saw the difference between thinking *“fast”* and *“slow”*;

- We highlighted the main issues that we deal with with subsymbolic approach, along with some of the solutions that the ML research community is drawing out;

- We saw how the kinds neurosymbolic integration are meant to be, with an example for each level of integration.

I’M DONE.

Conclusions

Contact info

Valerio De Caro, Ph.D. Candidate
Department of Computer Science
University of Pisa
E-mail*: valerio.decaro@phd.unipi.it*

Thank you for your attention!