Learning Where and When to Reason in Neuro-Symbolic Inference

Cristina Cornelio, Jan Stühmer, Shell Xu Hu, Timothy Hospedales ~ Samsung AI Center-Cambridge

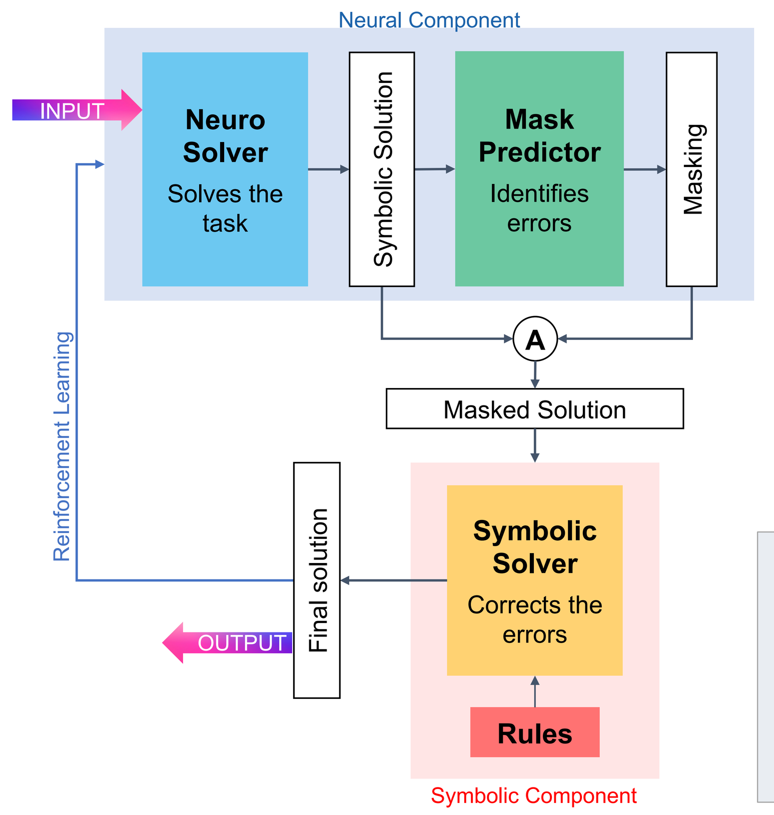
ICLR | May 1-5, 2023

Overview

Motivation

- **SOTA:** "Weak"-constraints = enforced only at training time
- **Goal:** Imposition of hard constrains at testing to ensure that the domain-specific knowledge is respected by the predictions
- **Idea:** Neuro-Symbolic pipeline **NASR** (**N**eural **A**ttention for **S**ymbolic **R**easoning)

Architecture



Given: a task to solve **&** a set of rules $\mathcal{R}$

1. **Neuro-Solver:** outputs an approximate solution
2. **Mask-Predictor:** identifies the components of the symbolic-solution that do not satisfy the rules R
3. **Adapter function:** combines the symbolic-solution and the masking to form the masked solution (matching the Symbolic-Solver format)
4. **Symbolic-Solver:** uses the rules R to correct the masked components of the symbolic solution

- **Symbolic-Solver corrects the Neuro-Solver prediction errors identified by the Mask-Predictor**
- Symbolic reasoning is **not feasible** in many scenarios
- **Mask predictor:** makes the reasoning more efficient, directing the reasoning focus

NASR without RL

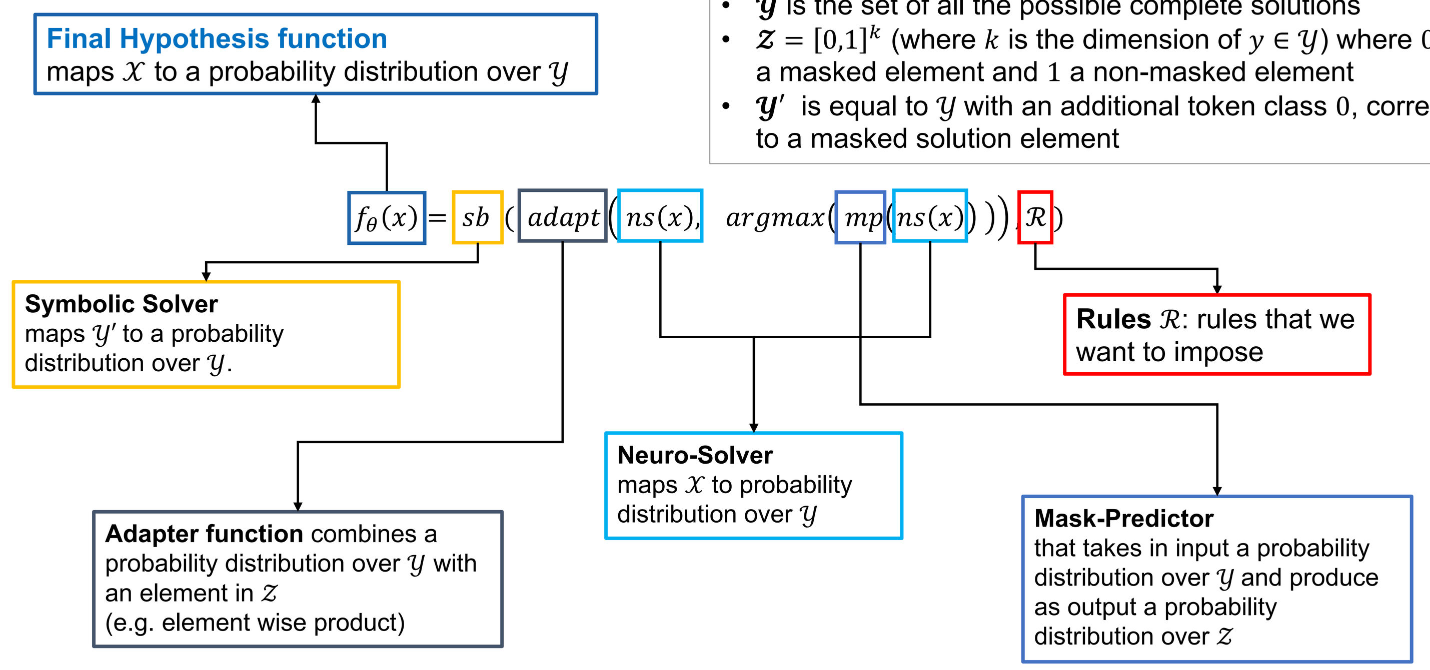
Neuro-Solver and **Mask-Predictor** are trained individually (**supervised learning**) and then integrated together

NASR with RL

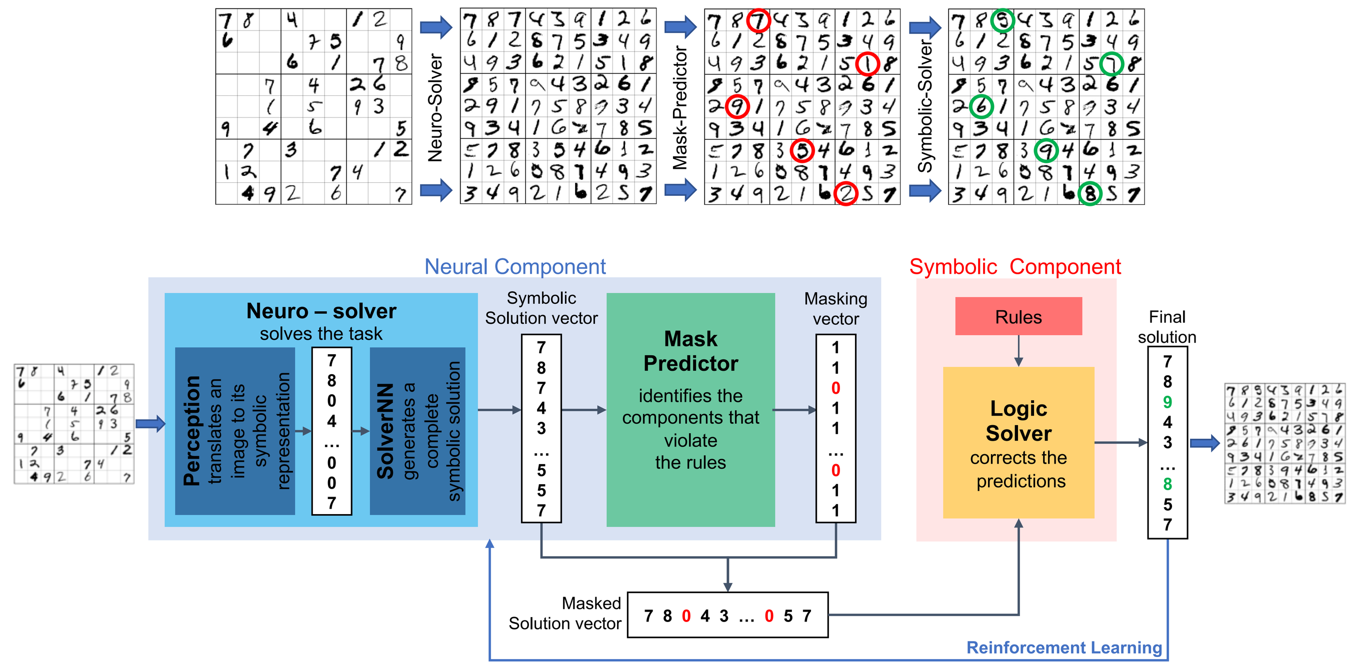
NASR is then refined using **Reinforcement learning**

$$\mathcal{L}(x;\theta) = -\ r\ /\log P_\theta(m|ns(x))$$

- $\mathcal{X}$ is the set of all possible inputs for the task under consideration
- $\mathcal{Y}$ is the set of all the possible solutions
- $\mathcal{Z} = [0,1]^k$ (where k is the dimension of $y \in \mathcal{Y}$) where 0 indicate a masked element and 1 a non-masked element
- $\mathcal{Y}'$ is equal to $\mathcal{Y}$ with an additional token class 0, corresponding to a masked solution element



Results – Visual Sudoku



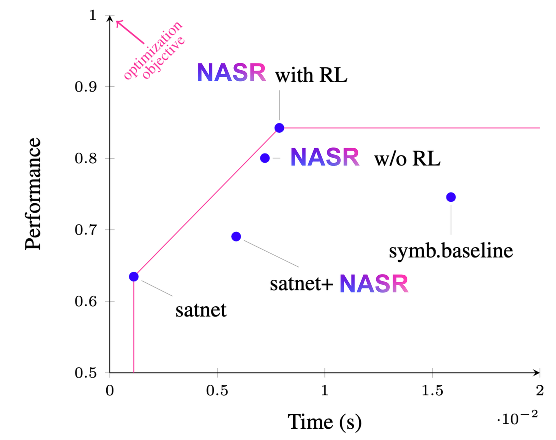
Results summary

- We significantly **outperform the baseline** in most of the cases (and never perform worst);
- We **improve** the performance of an **existing method, by integrating it** in our pipeline;
- We are more **robust to noise** compared to the symbolic baseline.

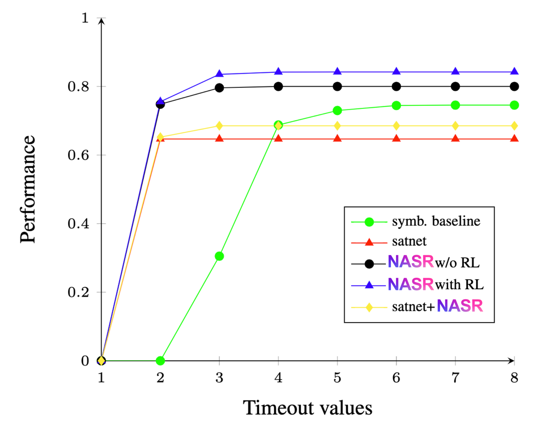
	big kaggle	minimal 17	multiple sol	satnet data
Symbolic baseline	74.56	**87.70**	63.50	63.20
NeurASP	timeout	**89.00***	timeout	timeout
SatNet	63.44	0.00	0.00	60.10
SatNet + NASR (our)	69.05	0.02	24.20	**81.40**
NASR (our)	**84.24**	87.00	**73.00**	**82.20**

Number of completely correct sudoku boards

Efficiency



Pareto front performance vs. computational time



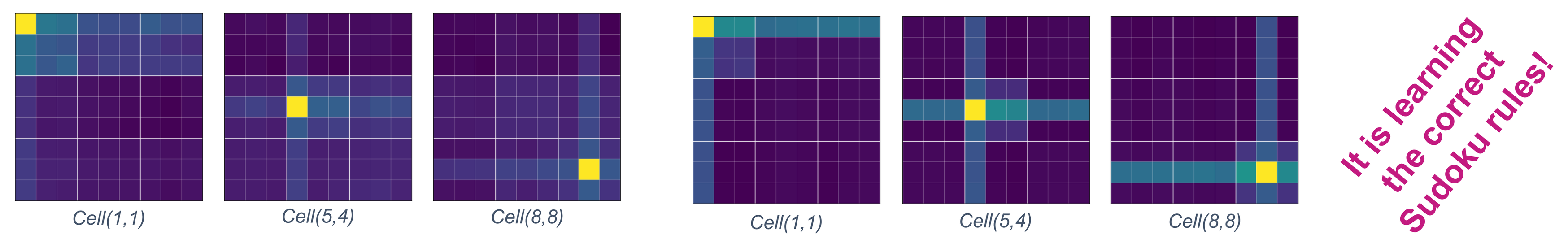
Performance limiting the computational time

Efficiency = trade-off between:

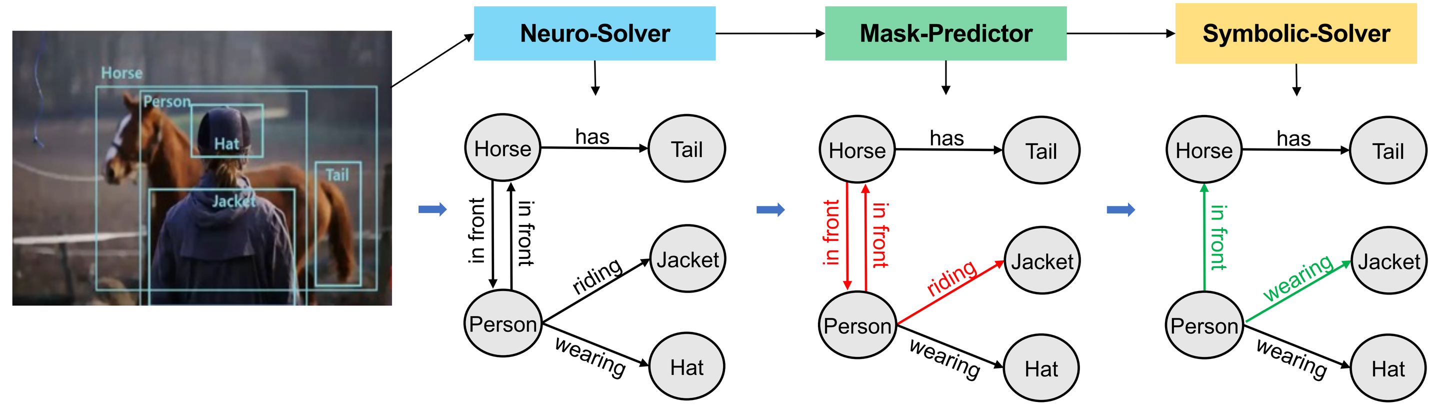
- performance (percentage of completely correct boards)
- computational time

Attention Maps

When considering the average attention in the transformer of a cell:
Focus on the **row**, the **column** and the **3×3 block** (the 3 Sudoku rules)



Results – Scene Graph (GQA)



Task

Predicate classification

- *input*: objects ground-truth bounding boxes and labels
- *output*: scene graph

Dataset

- GQA (balanced version of VG)

Rules

- Domain/range of the predicates (e.g., *domain(wear)={person}*)

Result

- *Between 1% to 2% improvement*

		R@20	R@50	R@100	R@200	R@300
All-shots	Baseline	29.22	42.35	48.48	50.75	51.11
	Max-improvement (PSB)	0.12	0.23	0.32	0.35	0.36
	% improvement of NASR	99.71	99.58	99.69	99.64	99.64
Zero-shots	Baseline	16.62	27.65	34.10	37.41	38.11
	Max-improvement (PSB)	0.91	1.43	1.93	2.18	2.33
	% improvement of NASR	100.00	100.00	100.00	100.00	100.00

NASR results: percentage of the max achievable improvement under the given ontology, defined by the Probabilistic Symbolic Baseline (PSB)

Take-away message

NASR: a neuro-symbolic method to manage the trade-off between the cost, expressivity, and exactness of reasoning during inference.

- Any type of rules/constraint can be used
- Results on Visual-Sudoku & Scene-Graph:
  - NASR outperforms the baseline
  - An existing method is improved when integrated in NASR
  - NASR is more efficient
  - NASR is more robust to noise compared to the symbolic baseline.

Open-Source GIT repository:

https://github.com/corneliocristina/NASR