Introduction to Neurosymbolic AI
Day 4: Reasoning Shortcuts

Emile van Krieken
Vrije Universiteit Amsterdam

ESSLLI 2025
7th of August

VU VRIJE UNIVERSITEIT AMSTERDAM

This course

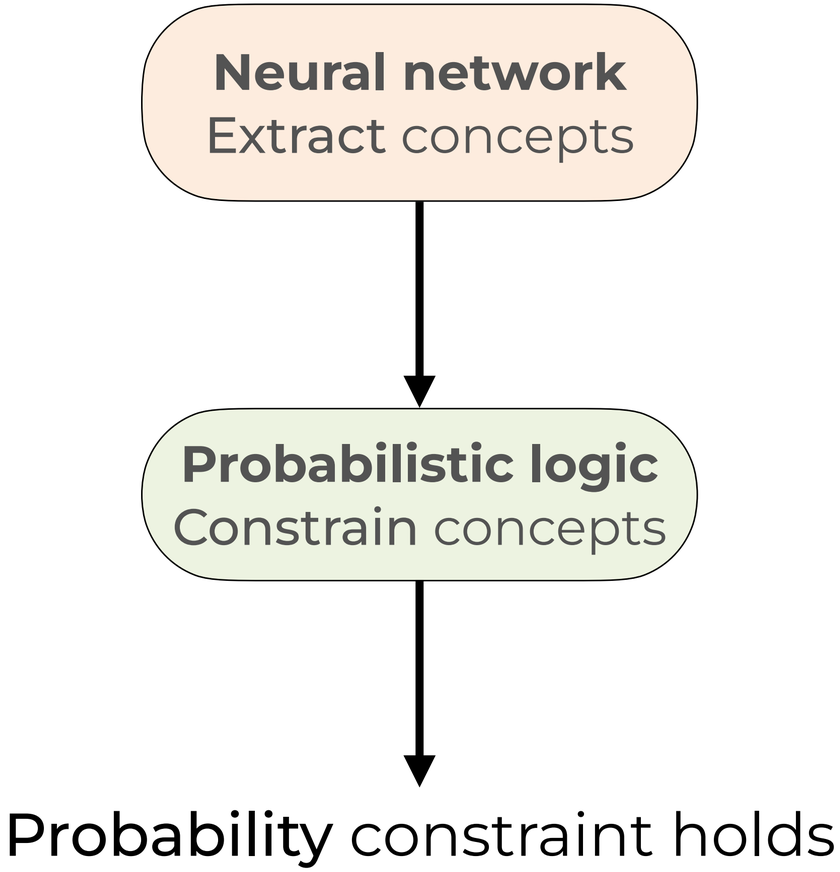
Day 1: Introduction to NeSy

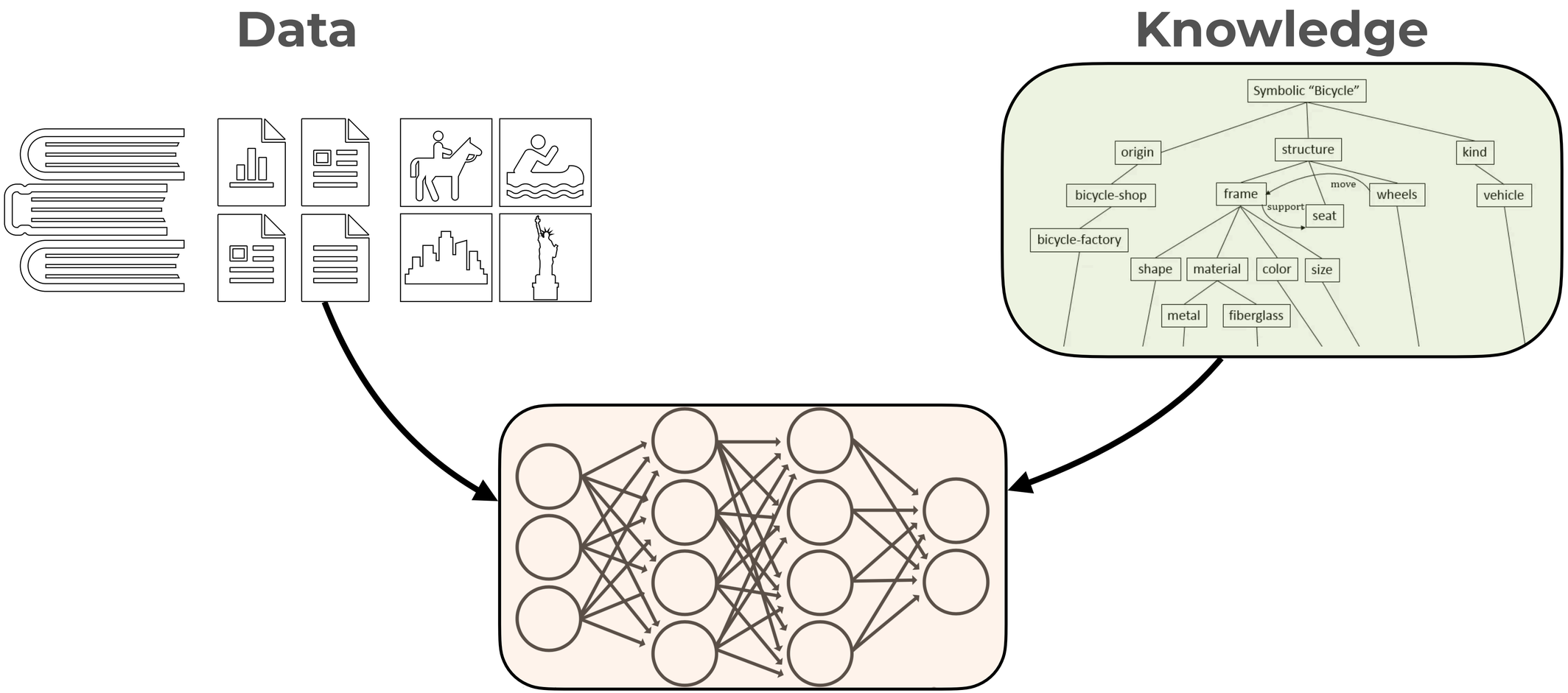
Day 2: Learnability

Day 3: Learning Imbalances in NeSy

Day 4: Reasoning Shortcuts

Day 5: Probabilistic Reasoning




Data
Knowledge
Symbolic “Bicycle”
origin
structure
kind
bicycle-shop
frame
move
wheels
vehicle
support
seat
bicycle-factory
shape
material
color
size
metal
fiberglass

Neurosymbolic AI

How to build reliable AI methods that integrate expert knowledge?

Content

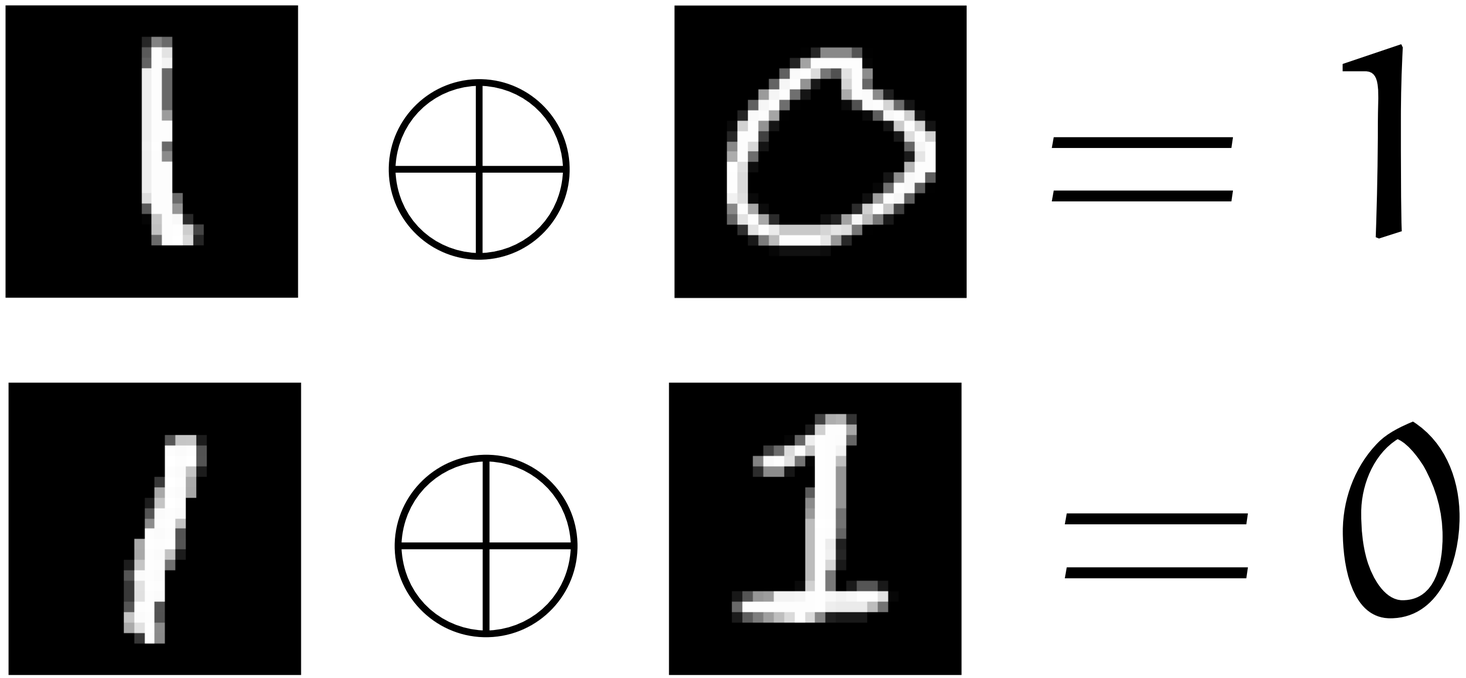
Introduction to Reasoning Shortcuts

Mitigation strategies

Awareness strategies

I have an machine learning background where we have very different customs. Let me know about anything that's unclear!

Material from our upcoming journal article on Reasoning Shortcuts. Thanks to Emanuele Marconato, Stefano Teso, Samuele Bortolotti, Antonio Vergari, Andrea Passerini and Efi.

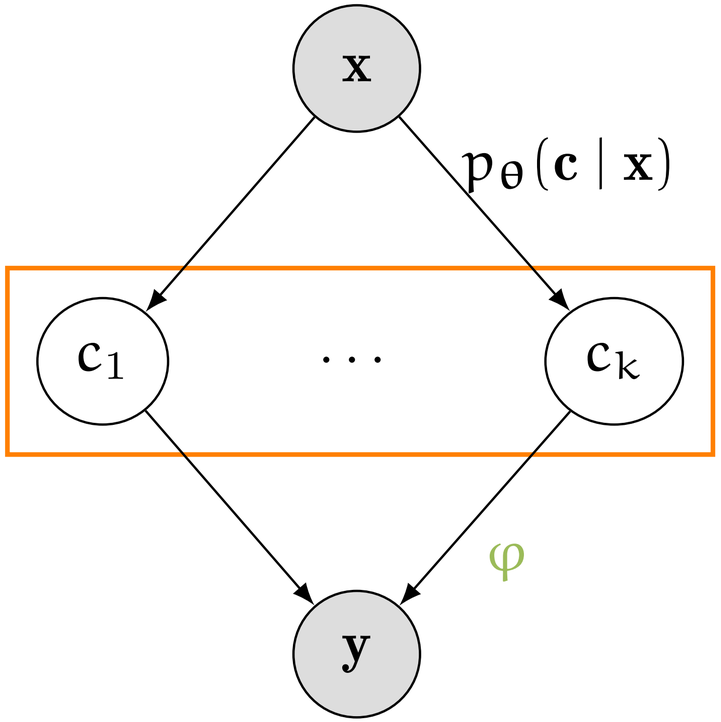


MNIST XOR

Predict **XOR value** $\mathbf{y}$ of two MNIST digits $\mathbf{x}$.

Labels of digits $\mathbf{c}$ **not given**!

Boring notation slide



$\mathbf{x} \in \mathcal{X}$	High-dim input, eg image
$\mathbf{c} \in \mathcal{C}$	Concept vector of concepts $\mathbf{c}_i$
$\mathbf{c}^* \in \mathcal{C}$	Ground-truth concept vector
$\mathbf{y} \in \mathcal{Y}$	Output labels
$\varphi : \mathcal{C} \to \mathcal{Y}$	Known program
$p_\theta(\mathbf{c} \mid \mathbf{x})$	Probability neural network assigns to $\mathbf{c}$ given input $\mathbf{x}$
θ	Neural network parameters
$p_\theta(\mathbf{y} \mid \mathbf{x})$	Predict label $\mathbf{y}$ given input $\mathbf{x}$

$$p_\theta(\mathbf{y} \mid \mathbf{x}) = \sum_{\mathbf{c} \in \mathcal{C}} p_\theta(\mathbf{c} \mid \mathbf{x}) \cdot \mathbb{1}[\varphi(\mathbf{c}) = \mathbf{y}]$$

Modelling MNIST XOR

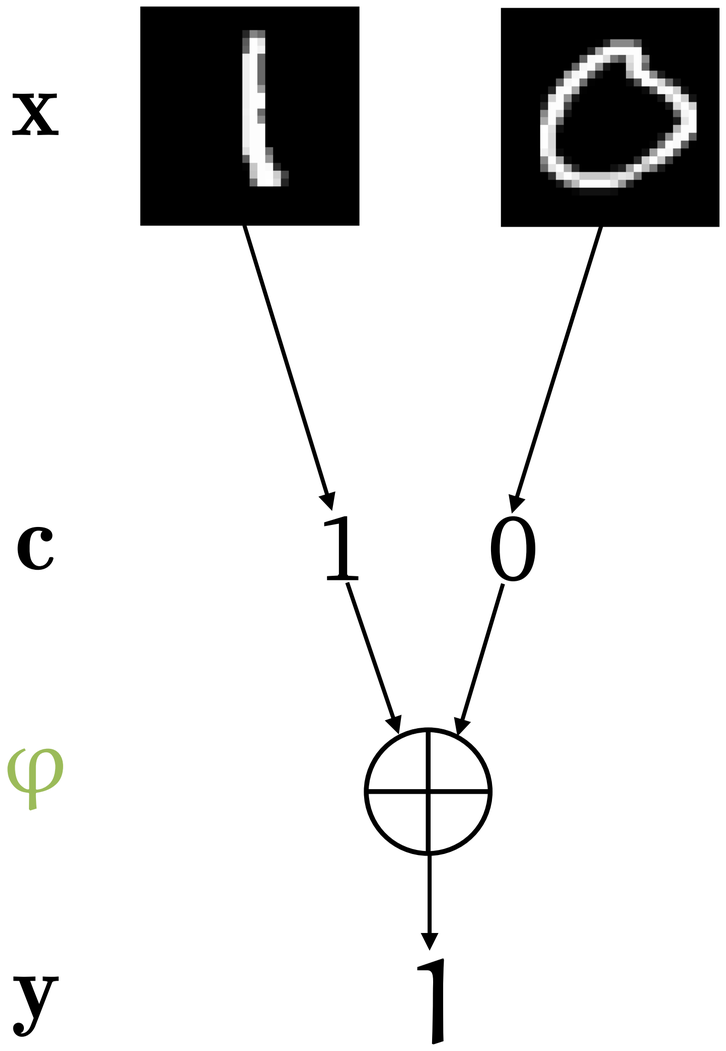
$\mathcal{X}$: Pairs of MNIST digits of 0 or 1

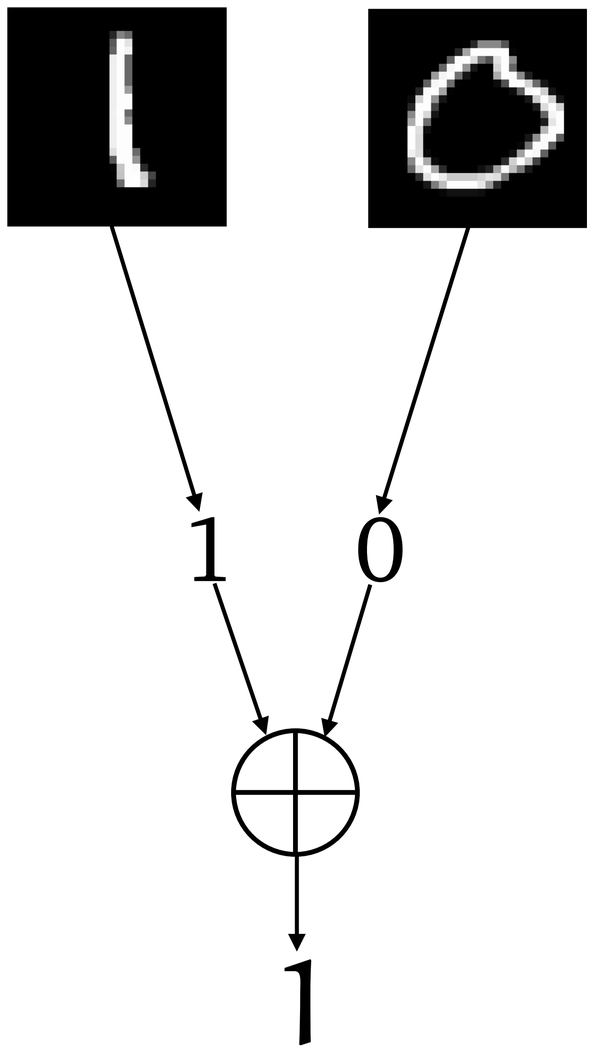
$\mathcal{C} = \{0,1\} \times \{0,1\} = \{(0,0),(0,1),(1,0),(1,1)\}$:

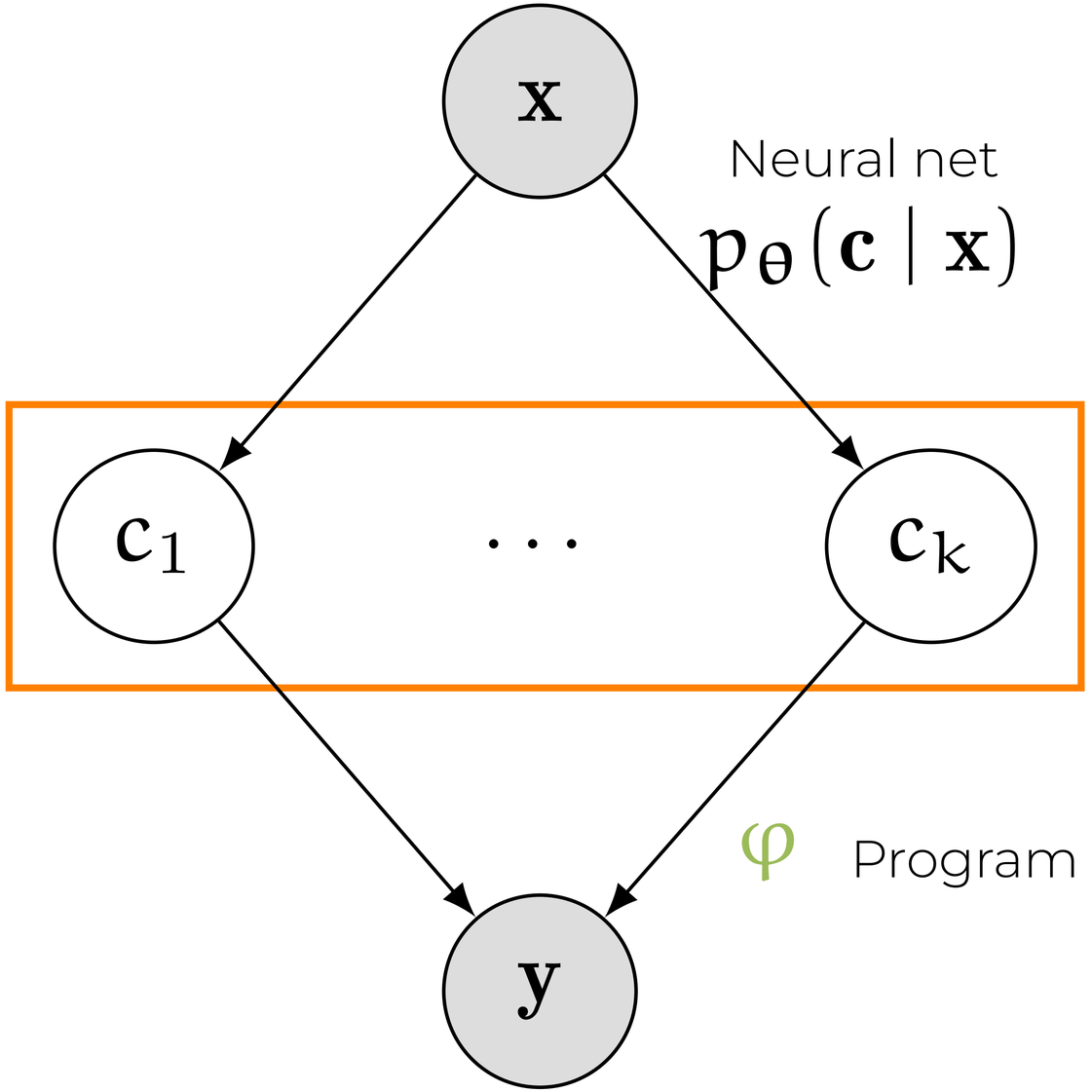
The possible '**concept vectors**'

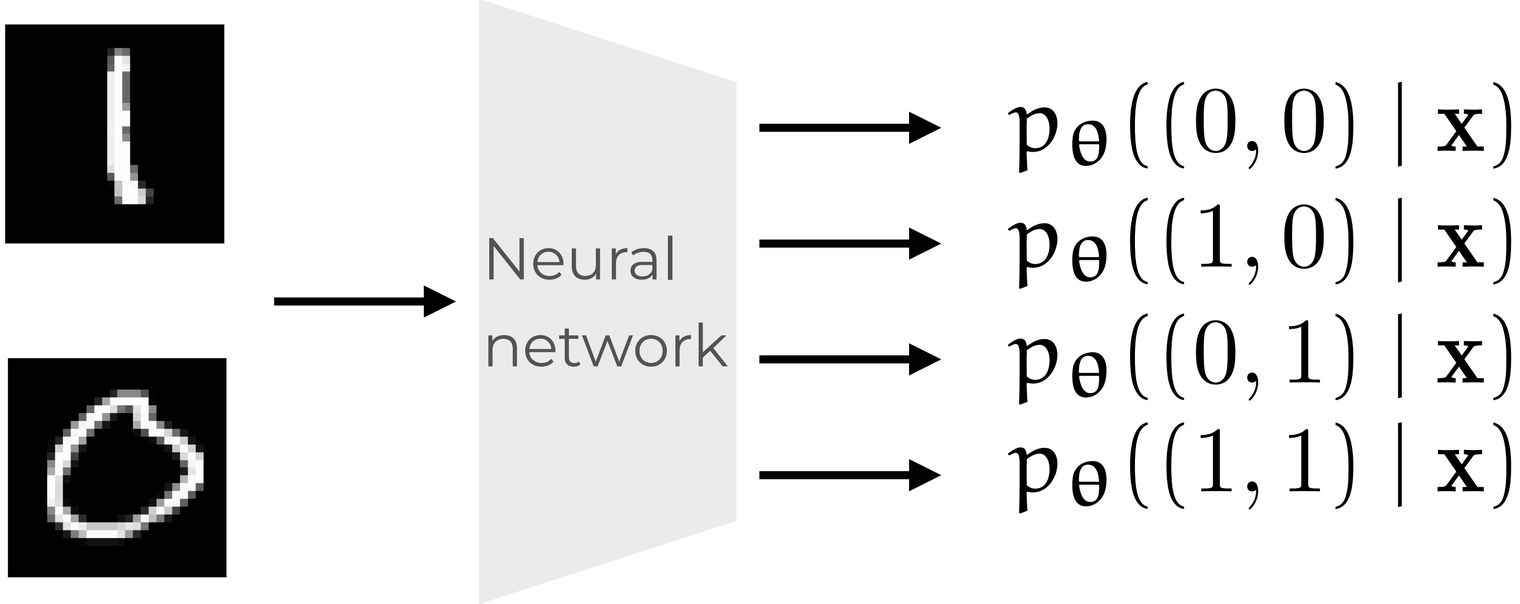
$\varphi(\mathbf{c}) = \mathbf{c}_1 \oplus \mathbf{c}_2$

$\mathcal{Y} = \{0,1\}$




1
0
1

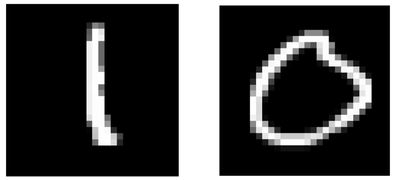

$\mathbf{x}$
Neural net
$p_\theta(\mathbf{c} \mid \mathbf{x})$
$\mathbf{c}_1$
$\cdots$
$\mathbf{c}_k$
φ
Program
$\mathbf{y}$



Neural networks produce a probability for **each possible concept vector** $\mathbf{c}$. These probabilities **sum to 1**.

Many ways to compute these probabilities in practice...
We'll see some examples later!

Training Neurosymbolic Predictors with Weighted Model Count



Neural network Program

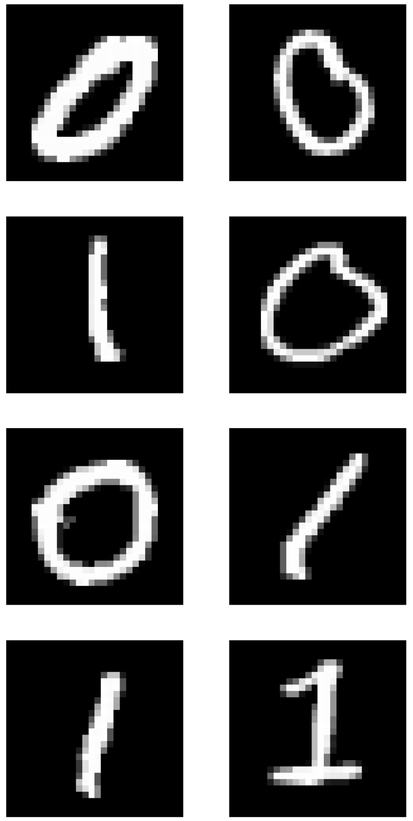
$$\mathbb{P}_\theta[c_1 \oplus c_2 = 1 \mid \mathbf{x}] = \sum_{\mathbf{c}} p_\theta(\mathbf{c} \mid \mathbf{x}) \cdot \mathbb{1}[c_1 \oplus c_2 = 1]$$

$$1 \oplus 0 = 1, \quad 0 \oplus 1 = 1$$

I will go in detail on WMC tomorrow, it is not important for this lecture!

Challenge:

Can we always trust neurosymbolic predictors when they get good accuracy?



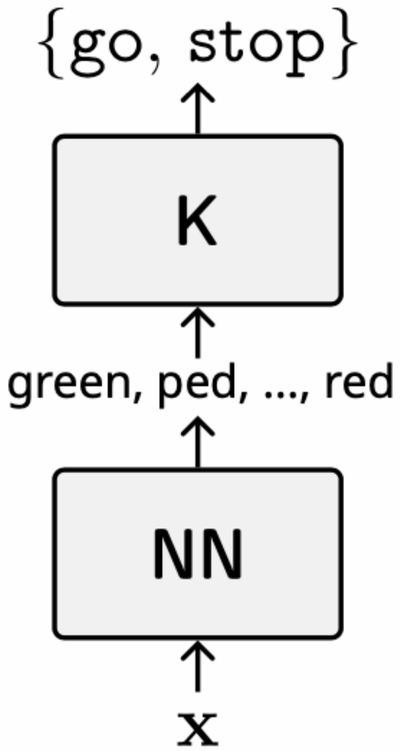
label	$\mathbf{c}^*$	$\mathbf{c}'$
0	$(0,0)$	$(1,1)$
1	$(1,0)$	$(0,1)$
1	$(0,1)$	$(1,0)$
0	$(1,1)$	$(0,0)$

NeSy predictor learning reasoning shortcut

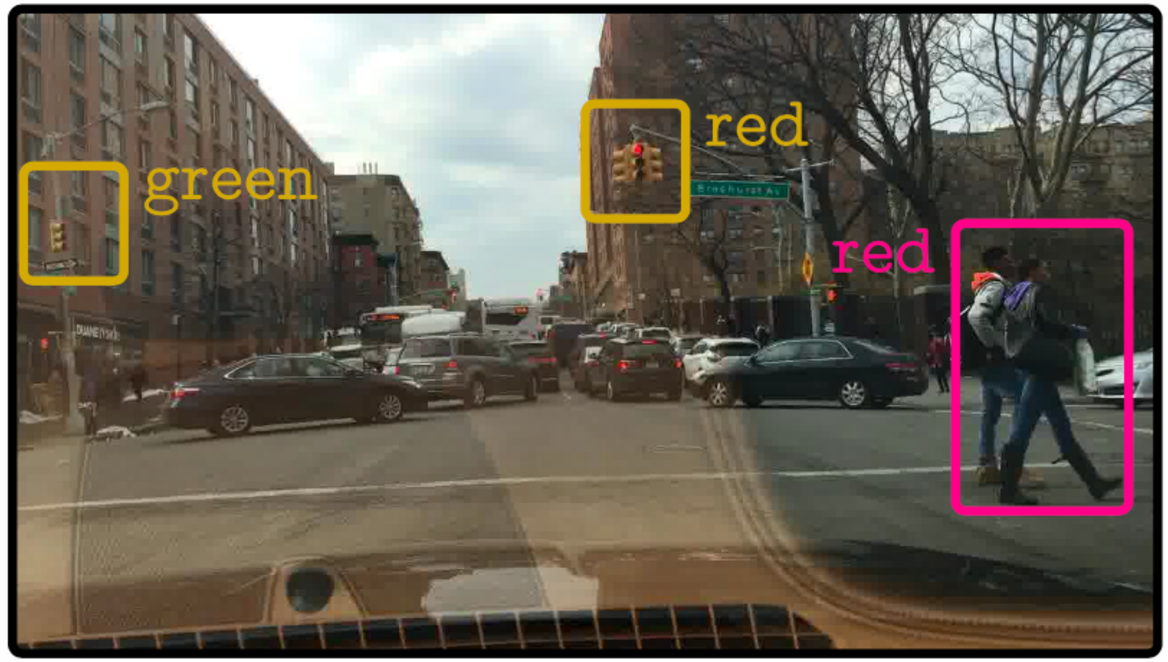
$(0,0)$ $(0,1)$ $(1,0)$ $(1,1)$

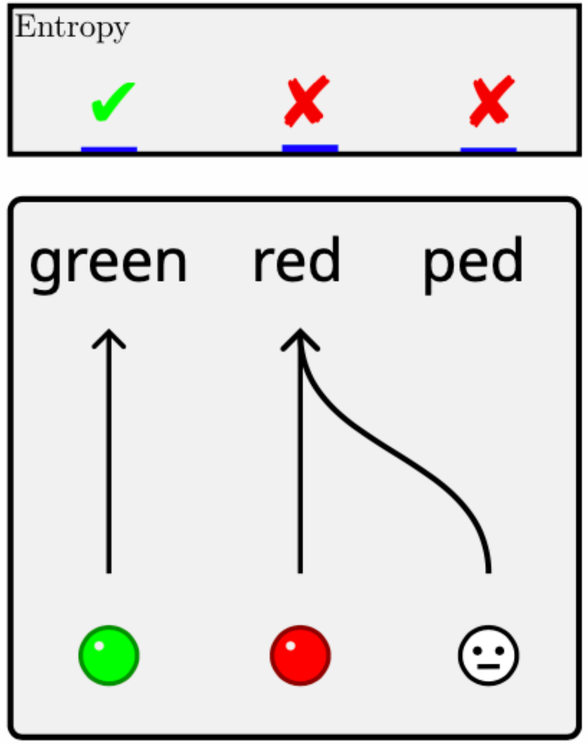
NeSy predictors MUST get high input-output accuracy while learning the wrong concepts!

$(c_1 \wedge \neg c_2) \vee (\neg c_1 \wedge c_2)$



$$\varphi = (\text{pedestrian} \vee \text{red} \Rightarrow \text{stop})$$

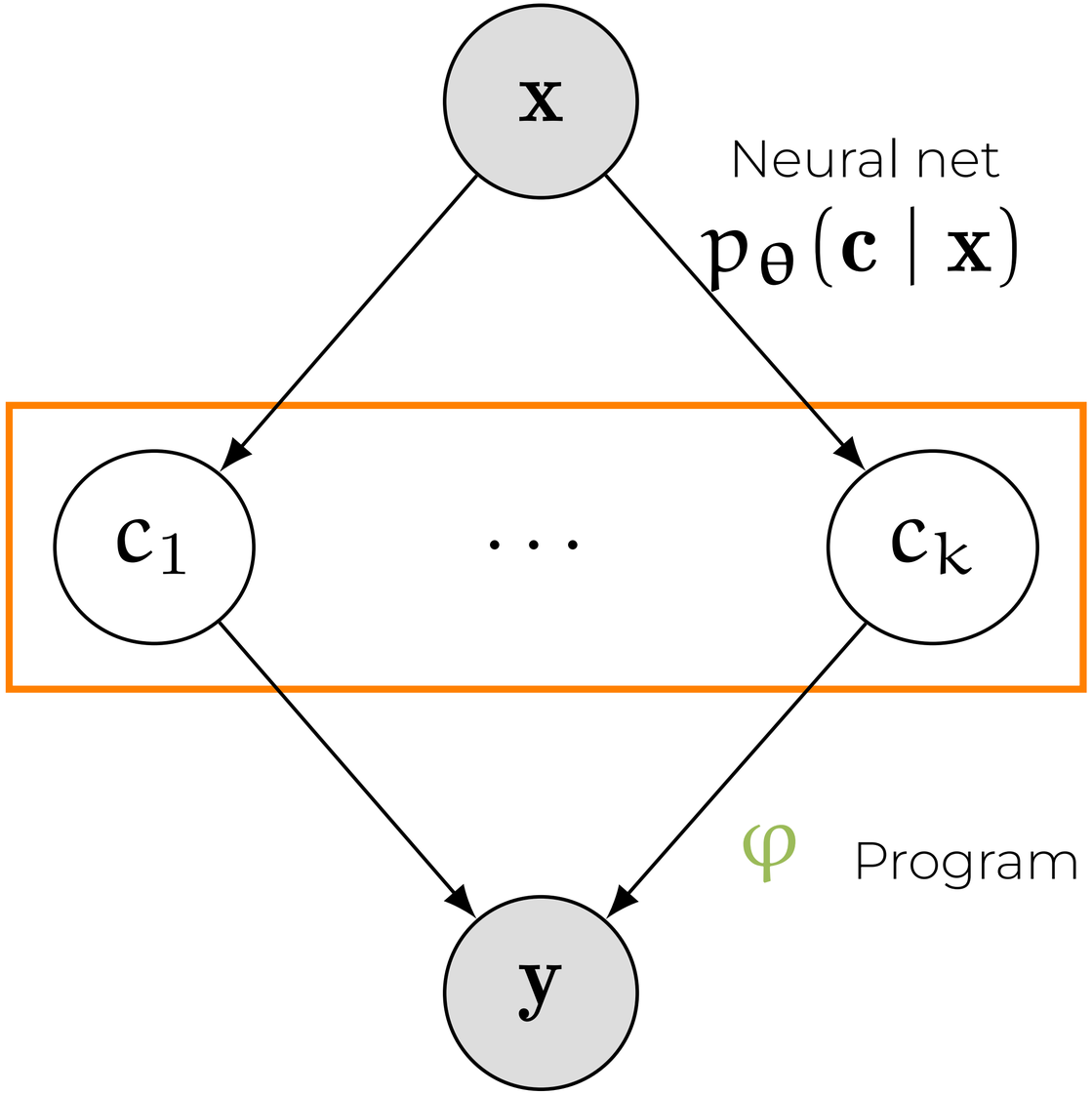


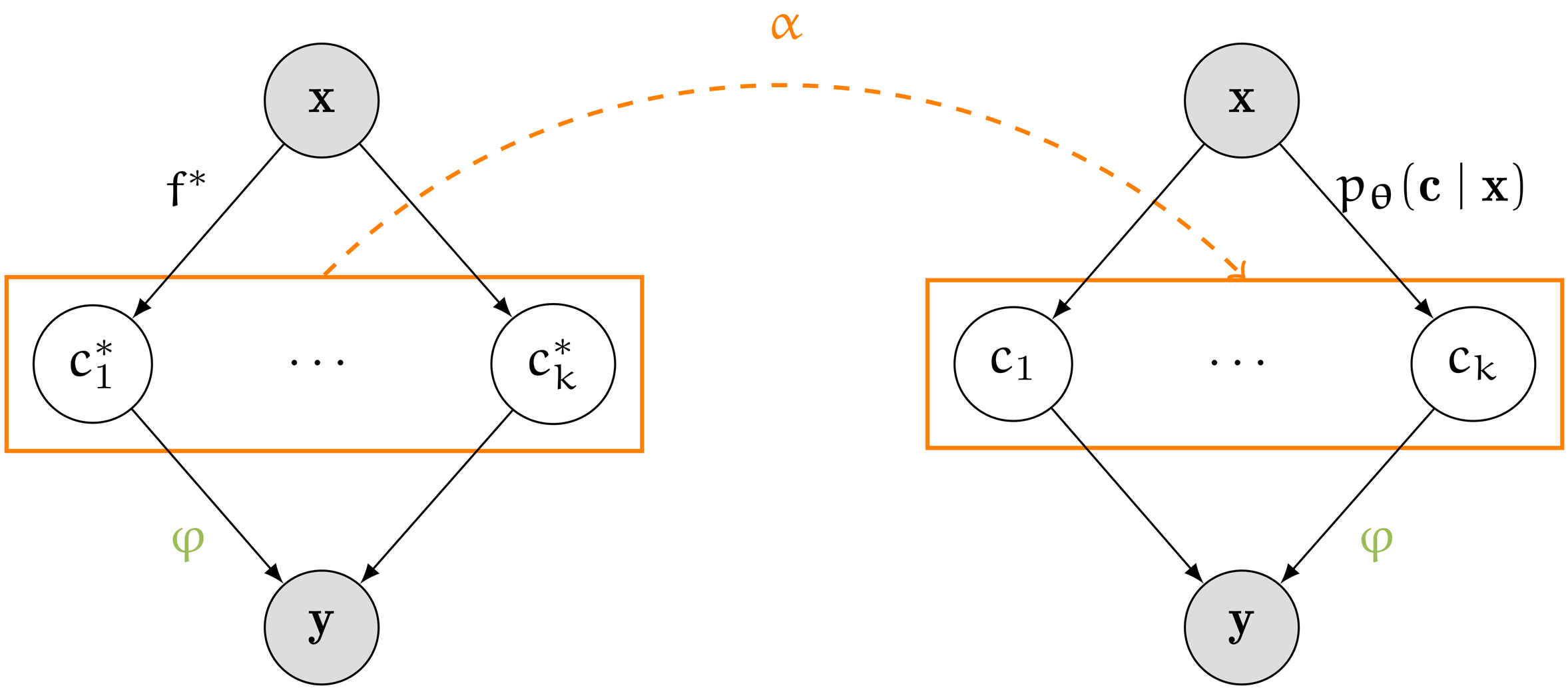


Reasoning shortcuts are a problem for **safety-critical** settings!

No **generalisation** out-of-distribution

Marconato, E., Teso, S., Vergari, A., & Passerini, A. (2024). Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. Advances in Neural Information Processing Systems, 36.


$\mathbf{x}$
Neural net
$p_\theta(\mathbf{c} \mid \mathbf{x})$
$\mathbf{c}_1$
$\cdots$
$\mathbf{c}_k$
φ
Program
$\mathbf{y}$

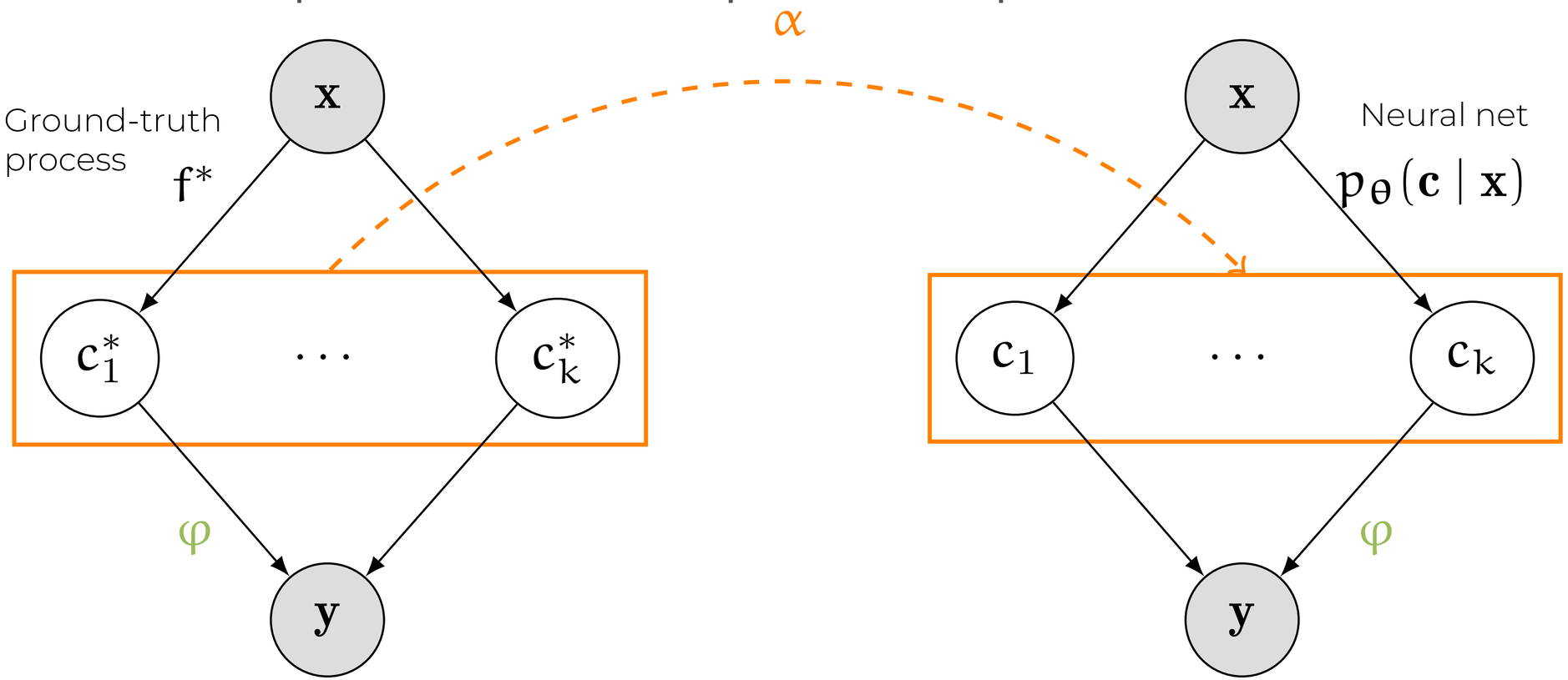


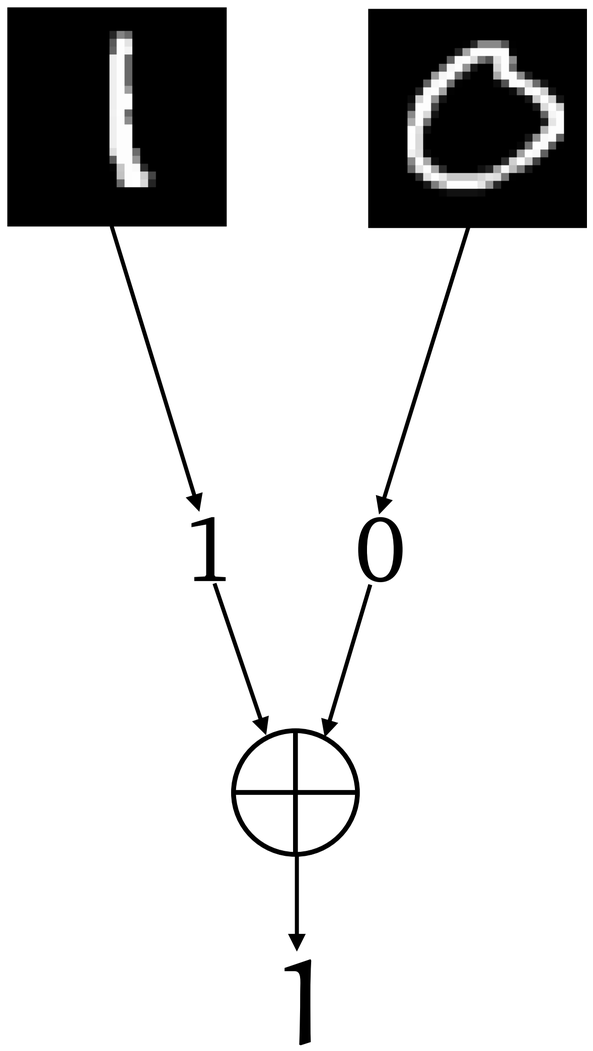
From Reasoning Shortcuts Journal paper, to appear

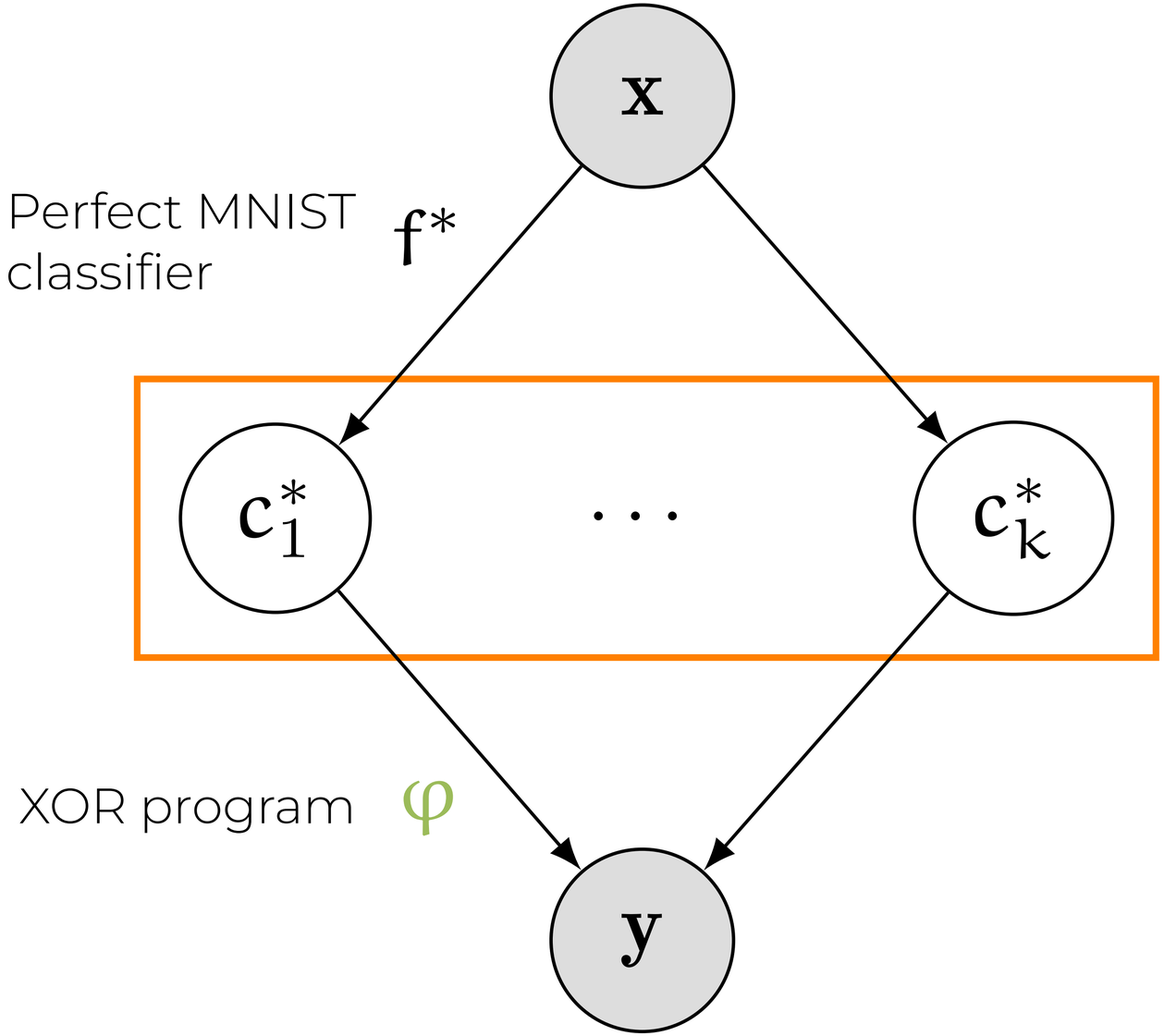
Assumptions

1. The data $(\mathbf{x}, \mathbf{y})$ is generated via the left process
2. **Ground-truth process** $p^*(\mathbf{c} \mid \mathbf{x})$ is deterministic $f^* : \mathcal{X} \to \mathcal{C}$

   Ie: Each input $\mathbf{x}$ has a unique concept vec $\mathbf{c}^*$ associated




1
0
1

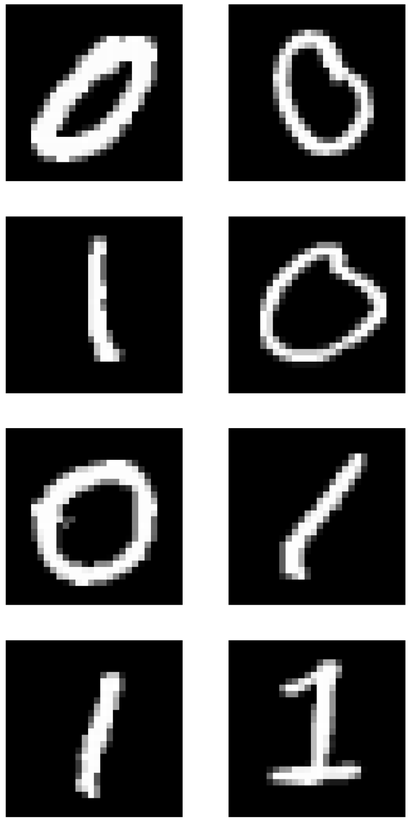

Perfect MNIST classifier
f^*
$\mathbf{x}$
$\mathbf{c}_1^*$
$\cdots$
$\mathbf{c}_k^*$
XOR program
φ
$\mathbf{y}$

Reasoning Shortcuts

Definition

A *(deterministic)* **reasoning shortcut (RS):** A non-identity map $\alpha : \mathcal{C} \to \mathcal{C}$ s.t. for all $\mathbf{c}^* \in \mathrm{supp}^*(\mathcal{C})$, we have $\varphi(\mathbf{c}^*) = (\varphi \circ \alpha)(\mathbf{c}^*)$

$$\begin{tikzcd}
\mathbf{c}^* \arrow[r, "\alpha"] \arrow[dr, "\varphi"'] & \mathbf{c} \arrow[d, "\varphi"] \\
& \mathbf{y}
\end{tikzcd}$$



label	$\mathbf{c}^*$	$\alpha(\mathbf{c}^*)$
0	$(0,0) \longrightarrow$	$(1,1)$
1	$(1,0) \longrightarrow$	$(0,1)$
1	$(0,1) \longrightarrow$	$(1,0)$
0	$(1,1) \longrightarrow$	$(0,0)$

$$\begin{tikzcd} \mathbf{c}^* \arrow[r, "\alpha"] \arrow[dr, "\varphi"'] & \mathbf{c} \arrow[d, "\varphi"] \\ & \mathbf{y} \end{tikzcd}$$

α is a shortcut for the MNIST XOR task

$$(c_1 \wedge \neg c_2) \vee (\neg c_1 \wedge c_2)$$

Neural networks may **learn reasoning shortcuts**!

Follow **ground-truth concept extractor** f^*, and **remap** with RS α

Such neural networks **perform perfectly** on test data!

$$\begin{tikzcd}
\overbrace{\mathbf{x} \arrow[r, "f^*"] & \mathbf{c}^* \arrow[r, "\alpha"] \arrow[dr, "\varphi"'] & \mathbf{c} \arrow[d, "\varphi"]}^{p_\alpha(\mathbf{c} \mid \mathbf{x})} \\
& & \mathbf{y}
\end{tikzcd}$$

We cannot detect we learned an RSs from pairs $(\mathbf{x}, \mathbf{y})$...

Can we prevent RSs?
Deal with them somehow?

$$\begin{tikzcd}
\mathbf{x} \arrow[r, "f^*"] & \mathbf{c}^* \arrow[r, "\alpha"] \arrow[dr, "\varphi"'] & \mathbf{c} \arrow[d, "\varphi"] \\
 & & \mathbf{y}
\end{tikzcd}$$

Content

Introduction to Reasoning Shortcuts

Mitigation strategies

Awareness strategies

Material from our upcoming journal article on Reasoning Shortcuts. Thanks to Emanuele Marconato, Stefano Teso, Samuele Bortolotti, Antonio Vergari, Andrea Passerini and Efi.

For problems with **more than 0 RSs**, likelihood maximisation (WMC) does not guarantee good solutions...
Then the problem is **unidentifiable**

So: figure out **the number of RSs**!
And **reduce** their number: **Mitigation strategies**

$$\begin{tikzcd}
\mathbf{x} \arrow[r, "f^*"] & \mathbf{c}^* \arrow[r, "\alpha"] \arrow[dr, "\varphi"'] & \mathbf{c} \arrow[d, "\varphi"] \\
 & & \mathbf{y}
\end{tikzcd}$$

Context in course

RSs relate to **day 2 about learnability.**
Today we focus on **identifiability**.

Identifiability: If we find a solution, ***is it the one that we want?***

(PAC)-Learnability: Is it *likely* we find this solution ***efficiently?***

So learnability is “stronger”

Counting Reasoning Shortcuts

$$\sum_{\alpha:\mathcal{C}\to\mathcal{C}} \mathbb{1}\left[\bigwedge_{\mathbf{c}^*\in\mathrm{supp}^*(\mathcal{C})} (\varphi\circ\alpha)(\mathbf{c}^*) = \varphi(\mathbf{c}^*)\right] - 1$$

Consider all *(allowed)* concept remappings

For all concept vecs that appear in the data...

...check if the program returns the same value after applying α

The identity map is not an RS

If so, we found an RS. Increase the count by 1!

label	$\mathbf{c}^*$	**RS1**	**RS2**	**RS3**	**RS4**	**RS5**	...
0	$(0,0)$	$(1,1)$	$(0,0)$	$(1,1)$	$(0,0)$	$(1,1)$	...
1	$(1,0)$	$(1,0)$	$(0,1)$	$(0,1)$	$(1,0)$	$(1,0)$	...
1	$(0,1)$	$(0,1)$	$(0,1)$	$(0,1)$	$(1,0)$	$(1,0)$	...
0	$(1,1)$	$(1,1)$	$(1,1)$	$(1,1)$	$(1,1)$	$(1,1)$	...

$$\begin{tikzcd} \mathbf{c}^* \arrow[r, "\alpha"] \arrow[dr, "\varphi"'] & \mathbf{c} \arrow[d, "\varphi"] \\ & \mathbf{y} \end{tikzcd}$$

$2^4 - 1 = 15$ reasoning shortcuts!

Without **any** mitigation

$$\prod_{\mathbf{y} \in \mathcal{Y}} |\mathcal{C}_{\mathbf{y}}|^{|\mathcal{C}_{\mathbf{y}}|} - 1$$

Multiply for all labels...

...the number of concept vecs with that output label $|\mathcal{C}_{\mathbf{y}}|$

label	$\mathbf{c}^*$
0	$(0,0)$
1	$(1,0)$
1	$(0,1)$
0	$(1,1)$

XOR example: 2 labels, both with 2 concepts vecs, so:

$$2^2 \cdot 2^2 - 1 = 4 \cdot 4 - 1 = 15$$

This grows **extremely** fast: **Exponential** in number of variables and labels... is NeSy hopeless?

How to reduce the number of Reasoning Shortcuts?

Knowledge-based mitigation

$$\prod_{\mathbf{y} \in \mathcal{Y}} |\mathcal{C}_{\mathbf{y}}|^{|\mathcal{C}_{\mathbf{y}}|} - 1$$

Multiply for all labels...

...the number of concept vecs with that output label $|\mathcal{C}_{\mathbf{y}}|$

label	$\mathbf{c}^*$
0	$(0, 0)$
1	$(1, 0)$
2	$(0, 1)$
3	$(1, 1)$

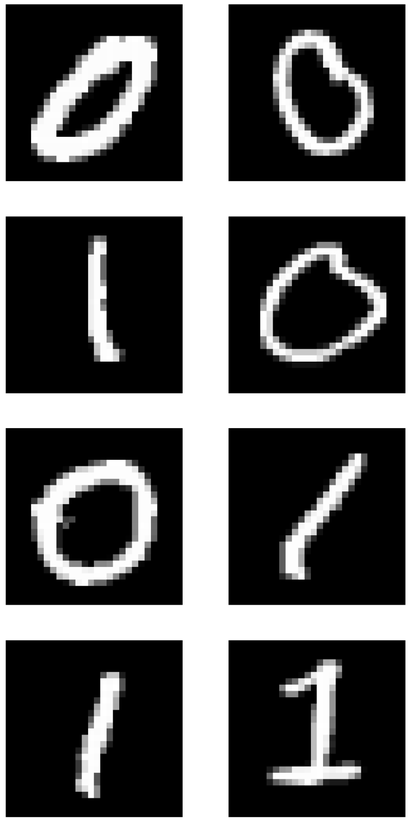
How to **change** the program φ to remove RSs?

Ensure each concept vector **has a different label**!

$1^1 \cdot 1^1 \cdot 1^1 \cdot 1^1 - 1 = 0$

(But then we're back at **regular supervised learning** as the program is **invertible**...)

Mitigation: **more fine-grained labels = better**



label	$\mathbf{c}^*$	$\alpha(\mathbf{c}^*)$
0	$(0,0) \longrightarrow$	$(1,1)$
1	$(1,0) \longrightarrow$	$(0,1)$
1	$(0,1) \longrightarrow$	$(1,0)$
0	$(1,1) \longrightarrow$	$(0,0)$

$$\begin{tikzcd} \mathbf{c}^* \arrow[r, "\alpha"] \arrow[dr, "\varphi"'] & \mathbf{c} \arrow[d, "\varphi"] \\ & \mathbf{y} \end{tikzcd}$$

15 is a lot of RSs for such a simple problem!

But intuitively, only α is relevant. Why?

Architecture-based mitigation

$$\sum_{\alpha:\mathcal{C}\to\mathcal{C}} \mathbb{1}\left[\bigwedge_{\mathbf{c}^*\in\mathrm{supp}^*(\mathcal{C})} (\varphi\circ\alpha)(\mathbf{c}^*) = \varphi(\mathbf{c}^*)\right] - 1$$

Sum over all concept maps

A neural network architecture might not be able to **implement** all maps $\alpha \circ f^*$

...

$$\begin{tikzcd}
\overbrace{\mathbf{x} \arrow[r, "f^*"] & \mathbf{c}^* \arrow[r, "\alpha"] \arrow[dr, "\varphi"'] & \mathbf{c} \arrow[d, "\varphi"]}^{p_\alpha(\mathbf{c}\mid\mathbf{x})} \\
& & \mathbf{y}
\end{tikzcd}$$

Architecture-based mitigation

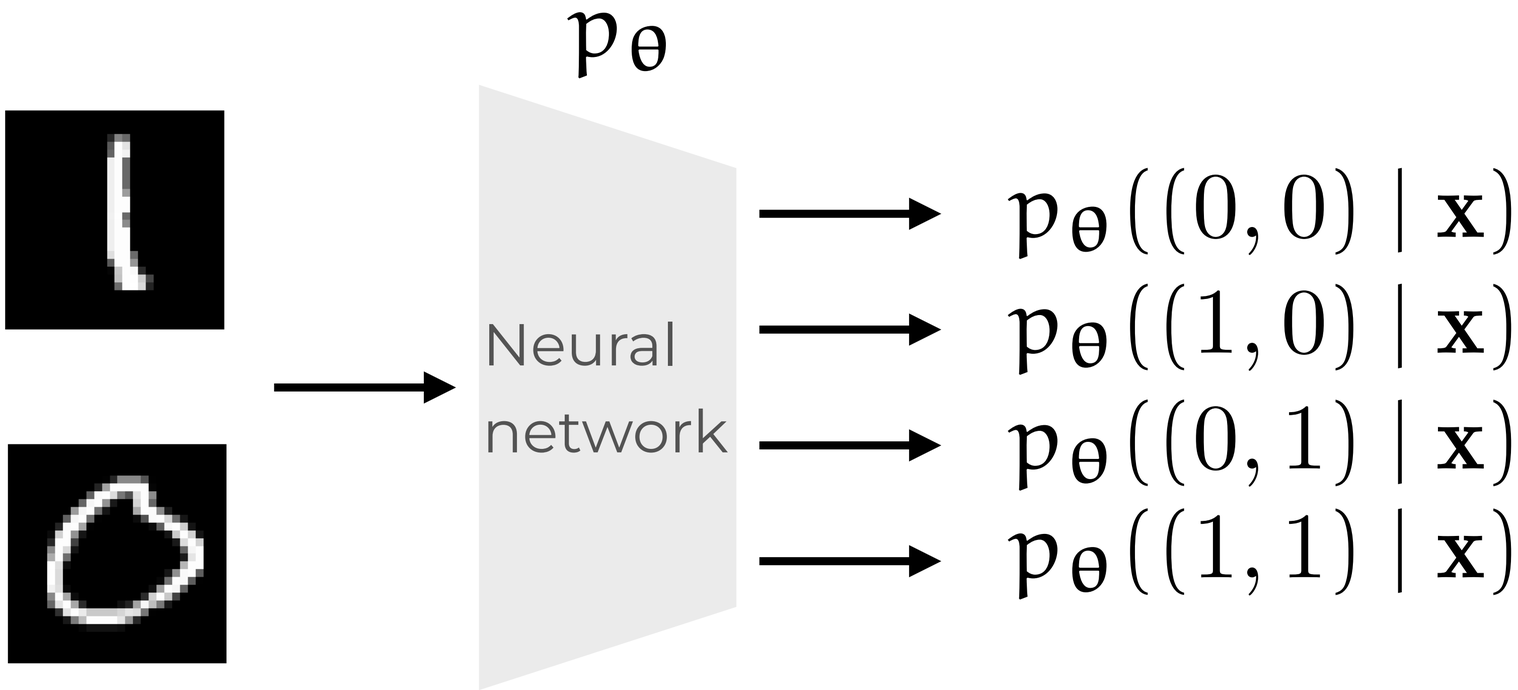
$$\sum_{\alpha \in \mathcal{A}} \mathbb{1}\left[\bigwedge_{\mathbf{c}^* \in \mathrm{supp}^*(\mathcal{C})} (\varphi \circ \alpha)(\mathbf{c}^*) = \varphi(\mathbf{c}^*)\right] - 1$$

Sum over all concept maps the neural network can implement

A neural network architecture might not be able to **implement** all maps $\alpha \circ f^*$

Then the sum is **only** over maps in $\mathcal{A}$

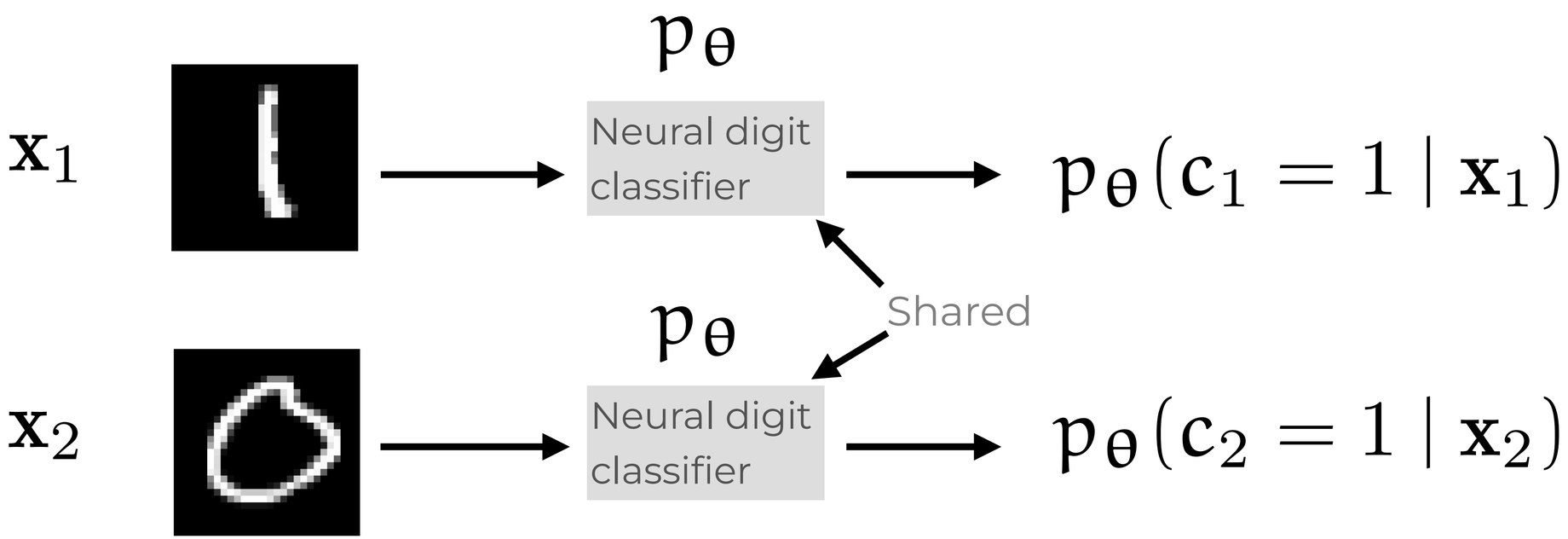
$$\begin{tikzcd}
\overbrace{\mathbf{x} \arrow[r, "f^*"] & \mathbf{c}^* \arrow[r, "\alpha"] \arrow[dr, "\varphi"'] & \mathbf{c} \arrow[d, "\varphi"]}^{p_\alpha(\mathbf{c} \mid \mathbf{x})} \\
& & \mathbf{y}
\end{tikzcd}$$



Default setting

This is a very powerful architecture! If the neural network is sufficiently big, it can model **all reasoning shortcuts.**

Can we adapt it to only focus on the ***relevant*** shortcut?



Disentangled architecture: Use a shared classifier that **independently** classifies the digits without seeing the other input

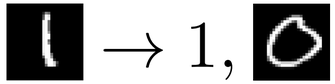
$$p_\theta((1,1) \mid \mathbf{x}_1, \mathbf{x}_2) = p_\theta(c_1 = 1 \mid \mathbf{x}_1) \cdot p_\theta(c_2 = 1 \mid \mathbf{x}_2)$$

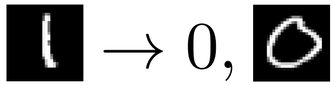
Used also in day 2 for learnability under M-unambiguity

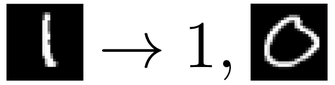
Using this independent parameterisation, we can only **remap** concept **values**, not concept **vectors***

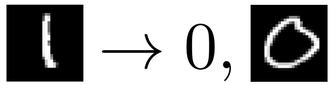
$$\begin{tikzcd} \mathbf{c}^* \arrow[r, "\alpha"] \arrow[dr, "\varphi"'] & \mathbf{c} \arrow[d, "\varphi"] \\ & \mathbf{y} \end{tikzcd}$$

All maps:

 $\rightarrow 1$, $\rightarrow 0$: **The correct solution!**

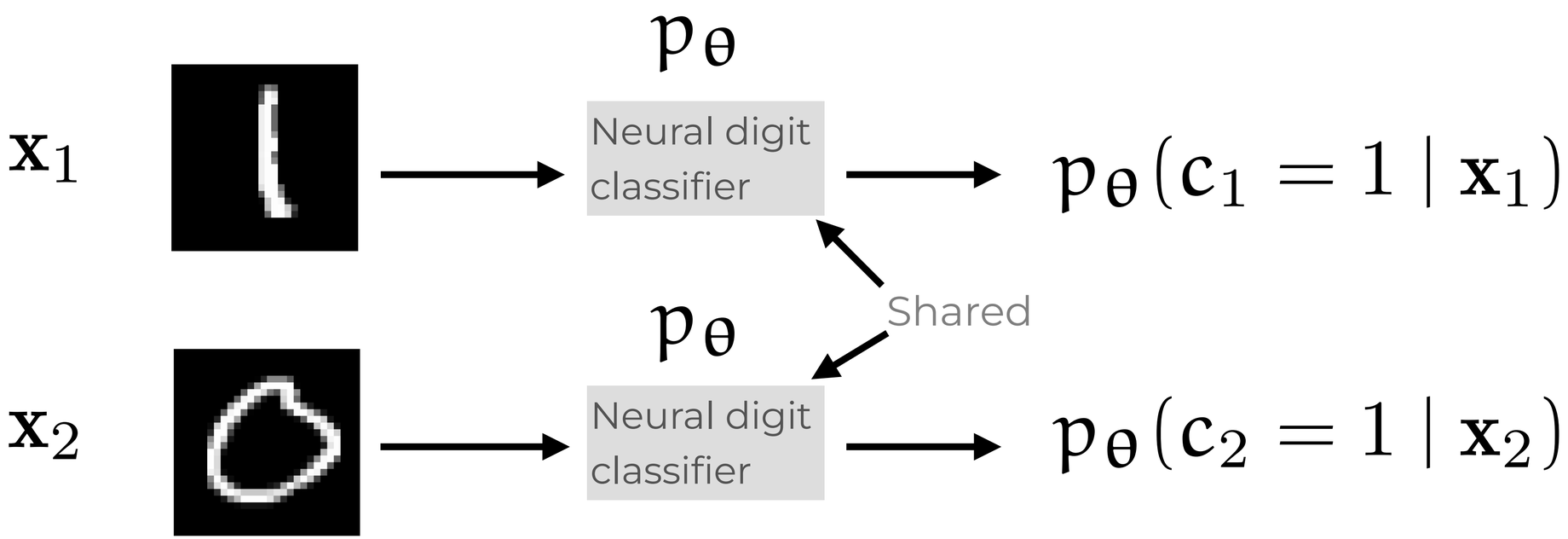
 $\rightarrow 0$, $\rightarrow 1$: **The main RS**

 $\rightarrow 1$, $\rightarrow 1$: **Always returns output** $y = 0$**, not an RS!**

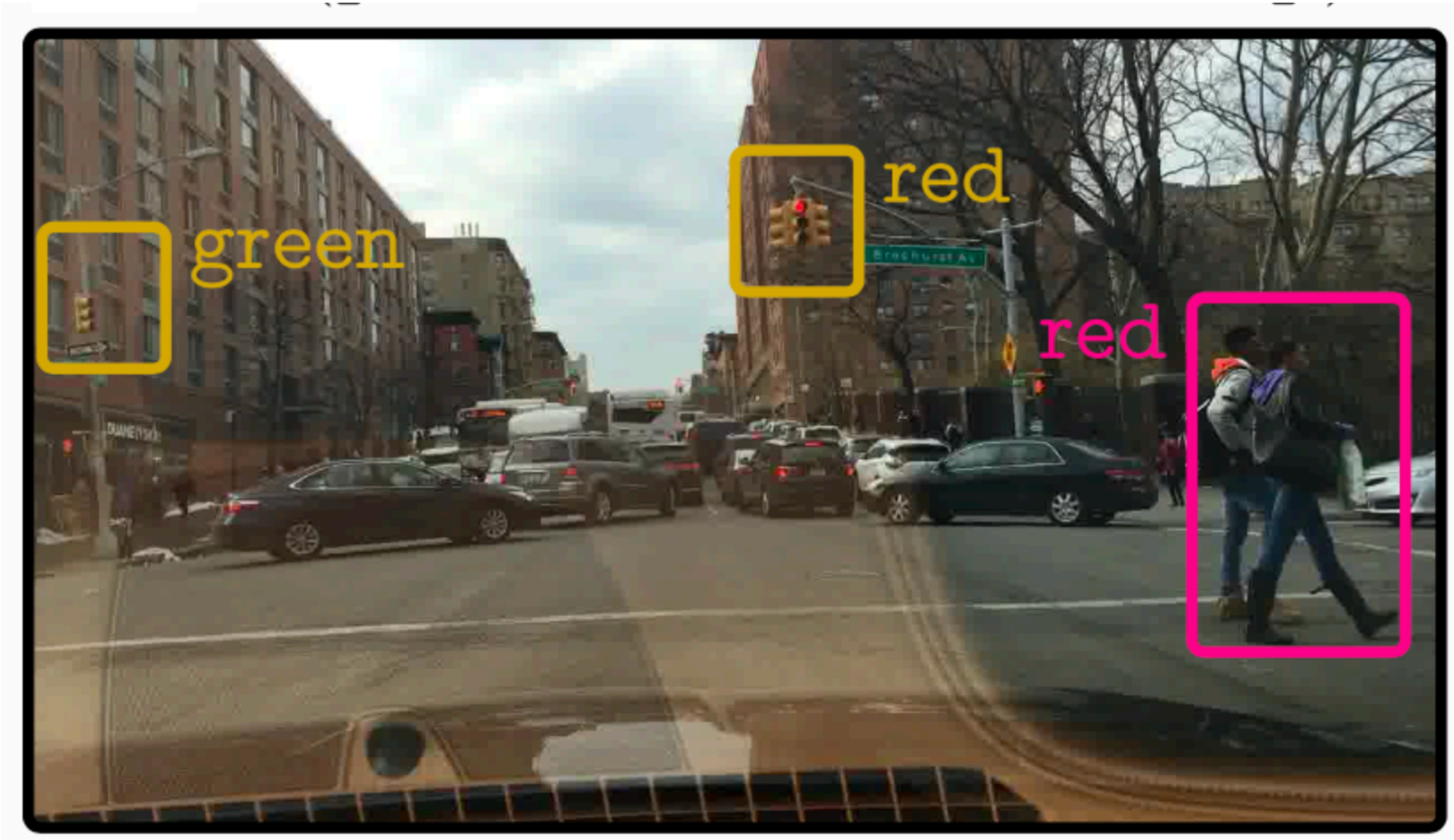
 $\rightarrow 0$, $\rightarrow 0$: **Always returns output** $y = 0$**, not an RS!**

Conclusion: With this architecture, **only 2 RSs** exist!

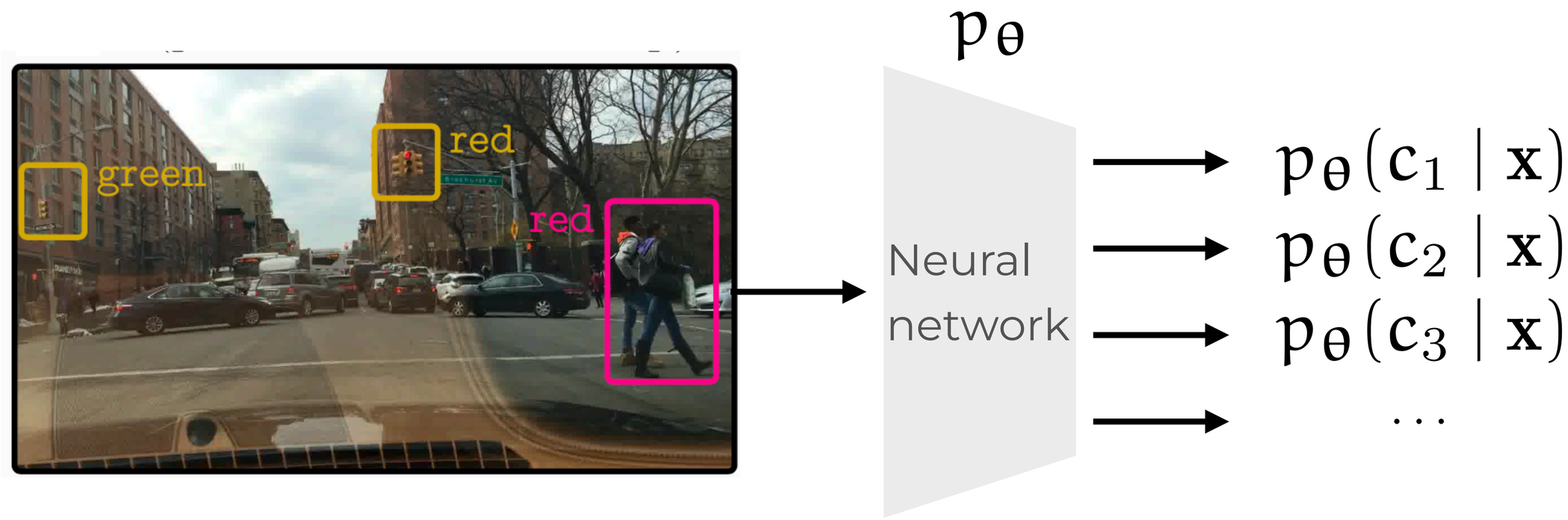
*Admittedly, the published theory is spotty here...



Is this disentangled setting realistic?



Q. Is it possible to create disentangled architectures for any problem?



Even under independent concepts, the number of Reasoning Shortcuts remains the same :(

Reason: If neural network is powerful, it can learn exactly $\alpha \circ f^*$

Distribution-based mitigation

$$\sum_{\alpha \in \mathcal{A}} \mathbb{1}\left[\bigwedge_{\mathbf{c}^* \in \mathrm{supp}^*(\mathcal{C})} (\varphi \circ \alpha)(\mathbf{c}^*) = \varphi(\mathbf{c}^*)\right] - 1$$

For all concepts that appear in the data...

If the data contains a large variety of concepts, the condition for RSs is stronger!

Distribution-based mitigation

Consider MNIST addition with only 2 examples

$$(\boxed{4\ 5}) \mapsto 9 \quad \textit{and} \quad (\boxed{3\ 2}) \mapsto 5.$$

Which has the RS (under independence)

$$\{\boxed{2} \mapsto 4, \boxed{3} \mapsto 1, \boxed{4} \mapsto 3, \boxed{5} \mapsto 6\}$$

Distribution-based mitigation

( ) $\mapsto 9$ *and* ( ) $\mapsto 5.$

Add new examples:

(, 10) , (, 8) .

Now can only be a 5: **Only value that sums with itself to 10!**

Then we can derive **, then** **, and then** **!**

Practical mitigation strategies

What are **practical strategies** for reducing #RSs?

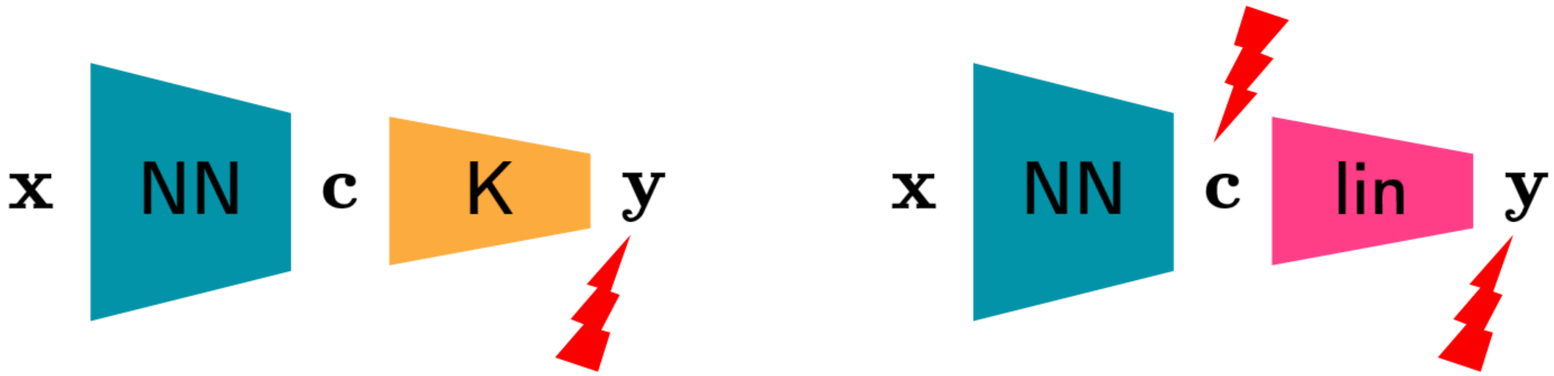
Changing the **program** is usually not possible...
Nor is changing the distribution!

These **change the actual data process**!

Strategy 1: Concept supervision

Most reliable strategy:

Directly provide ground-truth labels of concepts c_i!



(a) NeSy (DPL & LTN)

(b) CBM

Strategy 1: Concept supervision

Probably the most reliable strategy:
Directly provide ground-truth labels of concepts c_i!

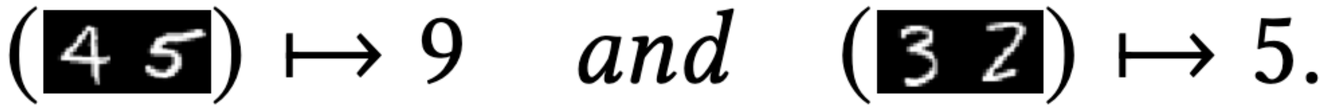
Adds additional supervision loss term.

Downside: To get labels requires **costly human labelling**...
And (worst-case) requires an exponential number of labels!

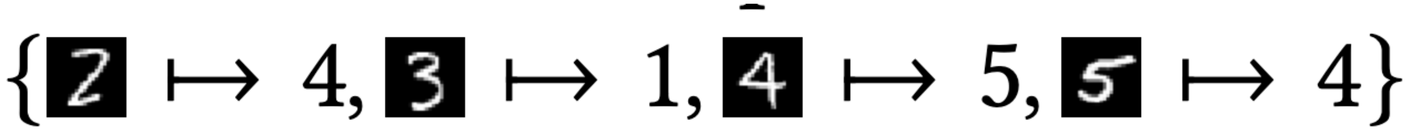
Some **explore foundation models** (LLMs etc) for getting labels!

Strategy 2: Clever loss functions

Consider again the example



This also has the RS



Which has two 4s!

We can **enforce our model** to assign many different labels to prevent assigning two 4s using **clever loss functions**

Strategy 3: Multi-task Learning

Combine **multiple tasks** on the **same concepts:**

Use **programs** $\varphi_1, \varphi_2, \ldots, \varphi_n$ for each task

Example

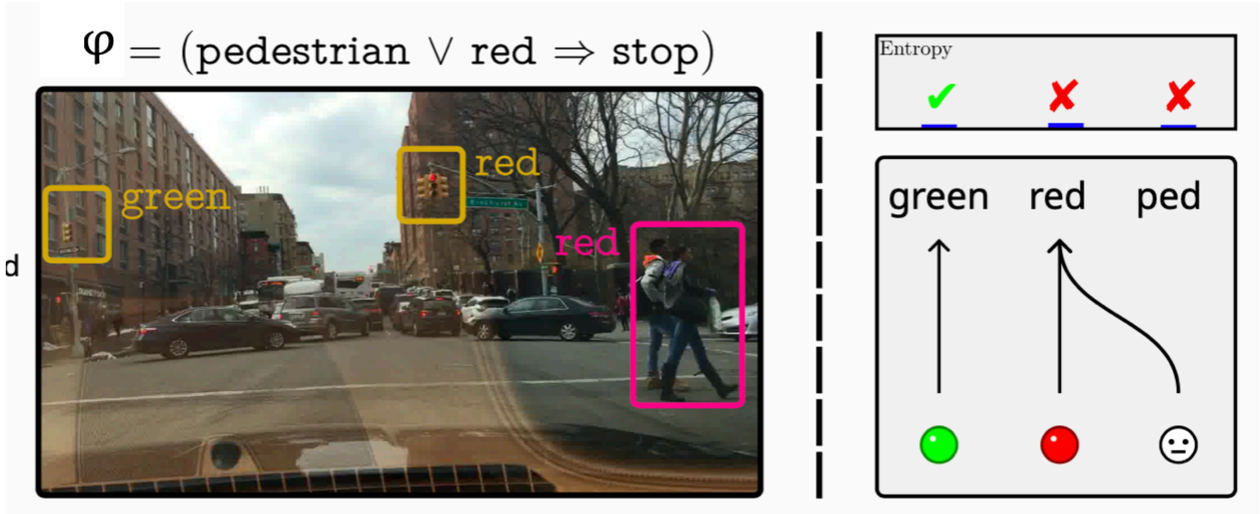
Combine MNIST Addition and MNIST Multiplication

Strategy 4: Unsupervised Learning

Unsupervised learning:

Learn **useful, summarising representations** of inputs $\mathbf{x}$

Rules out RSs that **conflate unrelated concepts (concept collapse)**, preventing simple summarisation



Strategy 4: Unsupervised Learning

Learn **useful, summarising representations** of inputs $\mathbf{x}$

Techniques:

- **Reconstruction learning**: NN should reconstruct the input from the concepts $\mathbf{c}$
  - Pedestrians and traffic lights look different - We cannot map them to the same concept!
- **Contrastive learning**: Learn input embeddings where similar inputs receive similar embeddings:
  Contrast augmentations (eg rotations) with different images

Overview of Mitigation

The number of **RSs** depend on:

1. The **program**
2. The **allowed concept remappings** (architecture)
3. The **distribution of concepts**

Mitigation strategies

1. Concept-supervision
2. Loss functions
3. Multi-task learning
4. Unsupervised learning

MITIGATION	REQUIRES	CONSTRAINT ON α
None	–	$\bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} \left((\beta_{\mathsf{K}} \circ \alpha)(\mathbf{g}) = \beta_{\mathsf{K}}(\mathbf{g}) \right)$
MTL	Tasks	$\bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} \bigwedge_{t \in [T]} \left((\beta_{\mathsf{K}^{(t)}} \circ \alpha)(\mathbf{g}) = \beta_{\mathsf{K}^{(t)}}(\mathbf{g}) \right)$
C	Sup. on $\mathbf{C}$	$\bigwedge_{\mathbf{g} \in \mathcal{S} \subseteq \mathsf{supp}(\mathbf{G})} \bigwedge_{i \in I} \left(\alpha_i(\mathbf{g}) = g_i \right)$
R	–	$\bigwedge_{\mathbf{g}, \mathbf{g}' \in \mathsf{supp}(\mathbf{G}) : \mathbf{g} \neq \mathbf{g}'} \left(\alpha(\mathbf{g}) \neq \alpha(\mathbf{g}') \right)$

RSBench

	Task	Data		
		Gen	OOD	ConL
Arithmetic	MNMath (new)	✓	✓	✓
	MNAdd-Half [25]	✗	✓✓	✗
	MNAdd-EvenOdd [17]	✗	✓✓	✓✓
Logic	MNLogic (new)	✓	✓	✓
	Kand-Logic [25]	✓	✓	✓
	CLE4EVR [17]	✓	✓	✓✓
High Stakes	BDD-OIA [20]	✗	✗	✗
	SDD-OIA (new)	✓	✓✓	✓

8 problems for **studying & mitigating Reasoning Shortcuts**

https://unitn-sml.github.io/rsbench/

Bortolotti, S., Marconato, E., Carraro, T., Morettin, P., van Krieken, E., Vergari, A., ... & Passerini, A. (2024). A Benchmark Suite for Systematically Evaluating Reasoning Shortcuts. NeurIPS 2024

Content

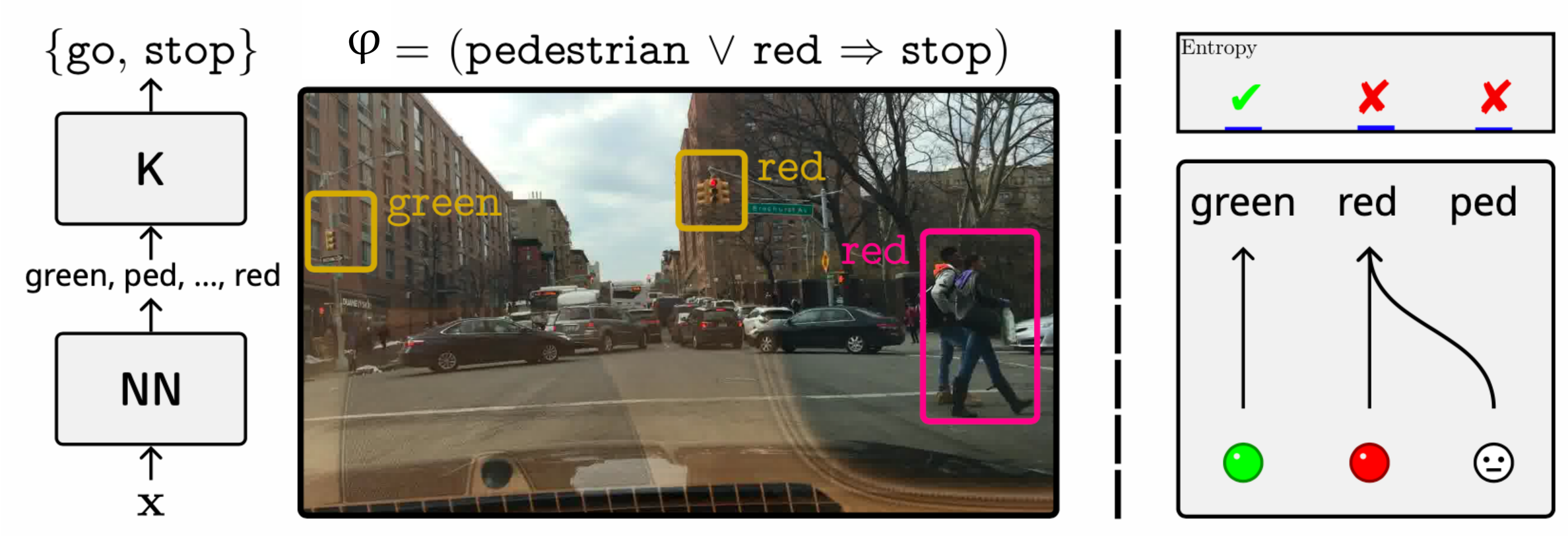
Introduction to Reasoning Shortcuts

Mitigation strategies

Awareness strategies

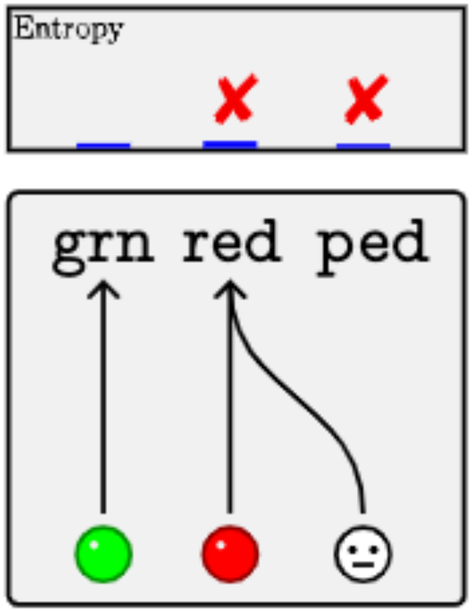
Material from our upcoming journal article on Reasoning Shortcuts. Thanks to Emanuele Marconato, Stefano Teso, Samuele Bortolotti, Antonio Vergari, Andrea Passerini and Efi.

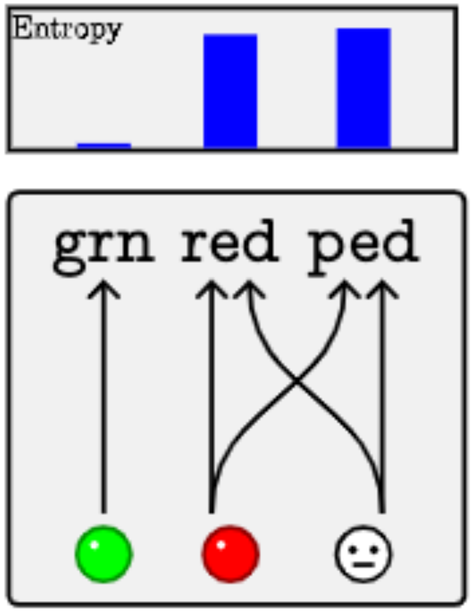
RS mitigation only gets us so far…
or is very costly!
What to do about *irreducible* RSs?


{go, stop}
K
green, ped, ..., red
NN
x
φ = (pedestrian ∨ red ⇒ stop)
green
red
red
Entropy
green
red
ped

$\varphi = (\text{pedestrian} \vee \text{red} \Rightarrow \text{stop})$

NeSy SotA





Marconato, E., Bortolotti, S., van Krieken, E., Vergari, A., Passerini, A., & Teso, S. BEARS Make Neuro-Symbolic Models Aware of their Reasoning Shortcuts. In The 40th Conference on Uncertainty in Artificial Intelligence.

Awareness strategies

Let's say we have a set $\mathcal{A}_{\mathrm{RS}}$ of irreducible* RSs.

Our neural network should **assign some confidence** in each!

* Yes, this is really vague

Awareness strategies

Assume set $\mathcal{A}_{\mathrm{RS}}$ of RSs.

An RS-aware neural network mixes between elements $\alpha \in \mathcal{A}_{\mathrm{RS}}$

$$\mathrm{p}_{\mathcal{A}_{\mathrm{RS}},\boldsymbol{\pi}}(\mathbf{c} \mid \mathbf{x}) = \sum_{\alpha \in \mathcal{A}_{\mathrm{RS}}} \pi_i \cdot \underbrace{\mathbb{1}[\alpha \circ \mathrm{f}^*(\mathbf{x}) = \mathbf{c}]}_{\mathrm{p}_{\alpha}(\mathbf{c}|\mathbf{x})}$$

where π_i sums to 1.

Can we create a practical model from this idea?

van Krieken, E., Minervini, P., Ponti, E., & Vergari, A. (2025). Neurosymbolic Reasoning Shortcuts under the Independence Assumption. arXiv preprint arXiv:2507.11357.

Neural network ensemble
Predict symbols

$$p_{\theta_1}(\mathbf{c} \mid \mathbf{x})$$
$$p_{\theta_2}(\mathbf{c} \mid \mathbf{x})$$
$$p_{\theta_3}(\mathbf{c} \mid \mathbf{x})$$

Feed each prediction through program $\varphi(\mathbf{c})$

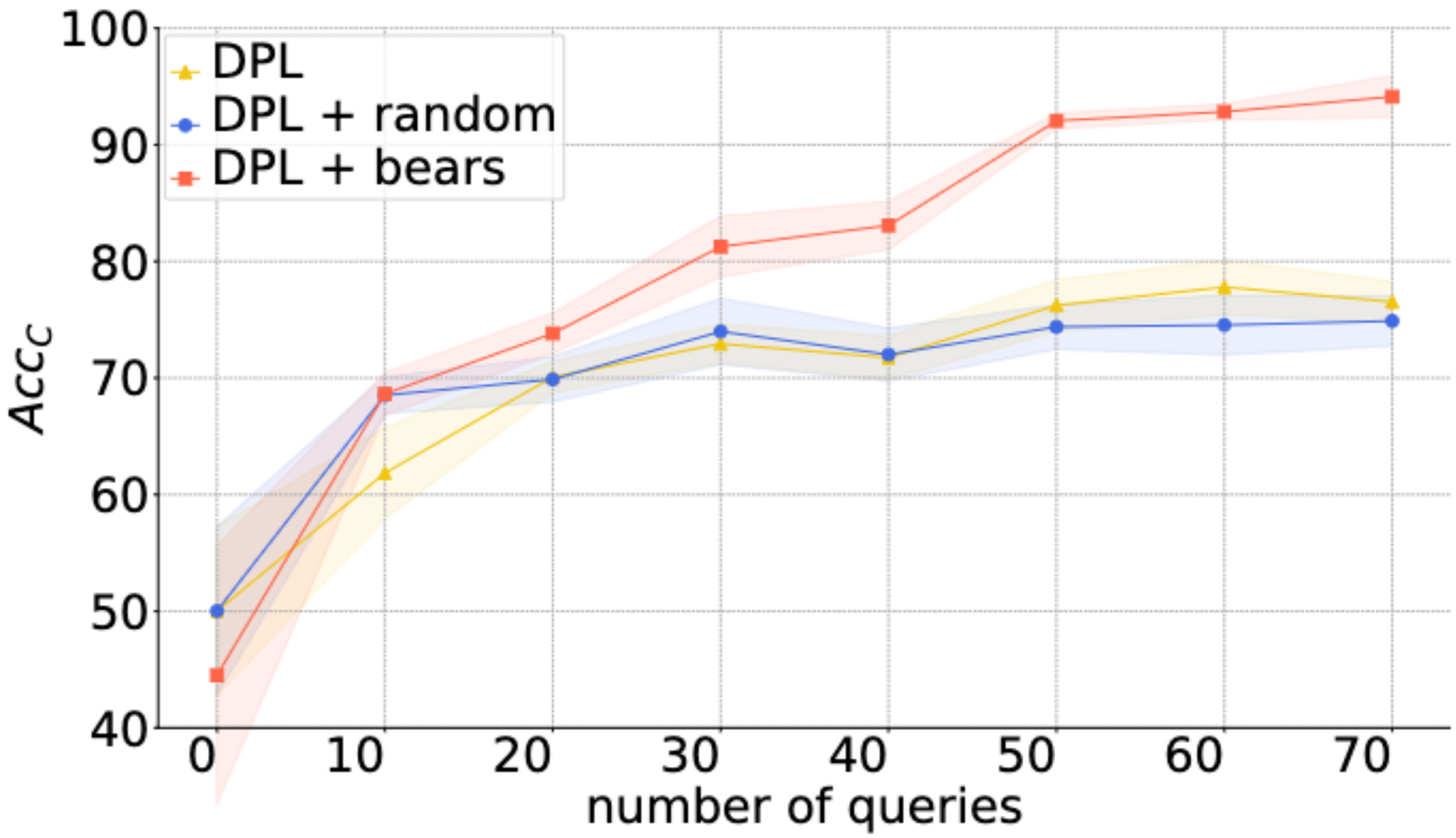
BEARS
Quantify **uncertainty**

Use **ensemble** of independent models!

Ensure each is '**different**':

Each represents **different** RSs

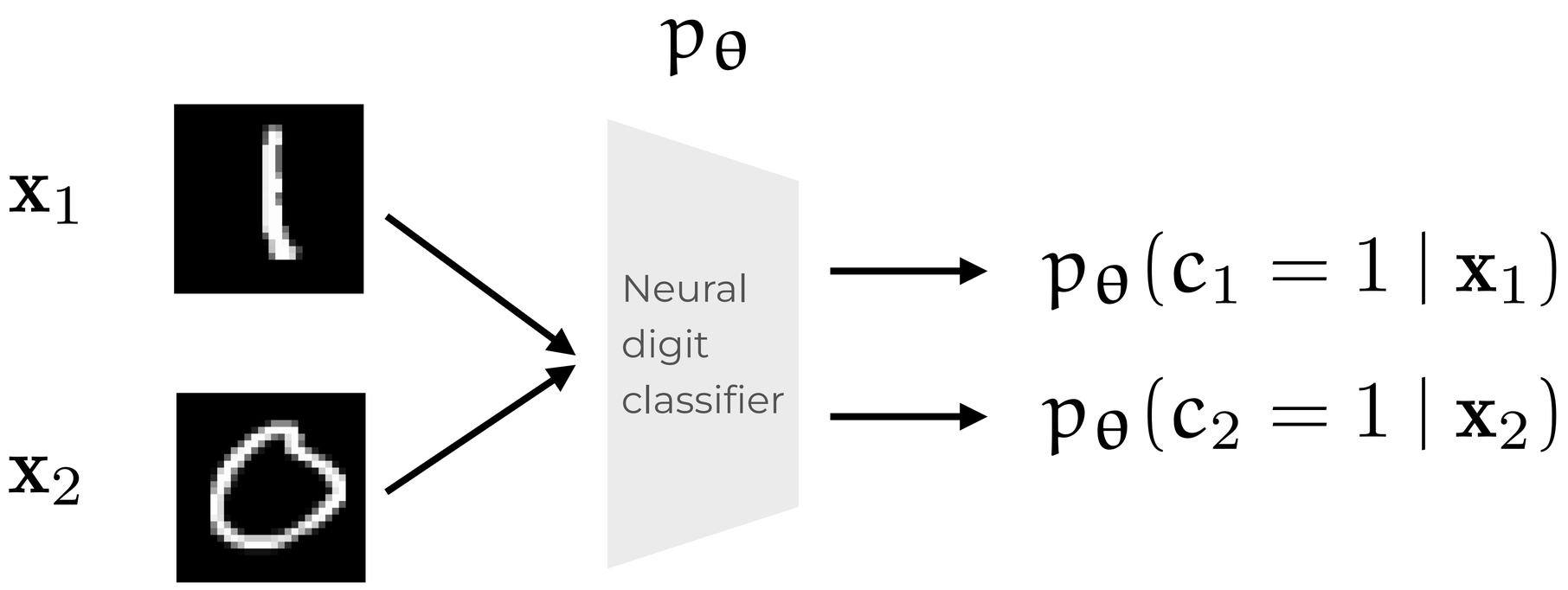
Marconato, E., Bortolotti, S., van Krieken, E., Vergari, A., Passerini, A., & Teso, S. BEARS Make Neuro-Symbolic Models Aware of their Reasoning Shortcuts. In The 40th Conference on Uncertainty in Artificial Intelligence.



Better **active learning**: Select concepts with **BEARS** that need **additional concept supervision**!

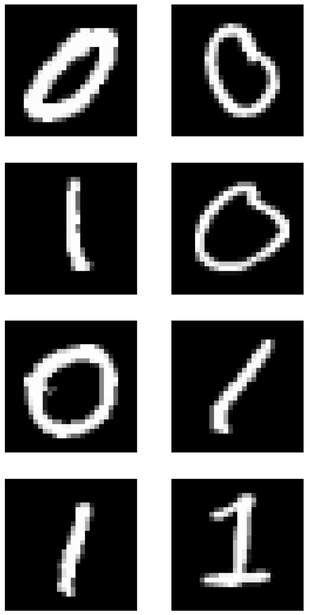
RS awareness requires mixing over possibly exponentially many RSs…

Can we do this efficiently within a single model?



Most practical NeSy architectures assume **independence** between the concepts

$$p_\theta((1,1) \mid \mathbf{x}_1, \mathbf{x}_2) = p_\theta(c_1 = 1 \mid \mathbf{x}_1) \cdot p_\theta(c_2 \mid \mathbf{x}_2)$$



label	$\mathbf{c}^*$	$\mathbf{c}'$
0	$(0,0)$	$(1,1)$
1	$(1,0)$	$(0,1)$
1	$(0,1)$	$(1,0)$
0	$(1,1)$	$(0,0)$

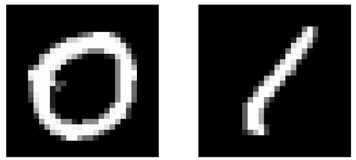
Independent model:
Find reasoning shortcut

$(0,0)$ $(0,1)$ $(1,0)$ $(1,1)$

Expressive models: **aware** of **reasoning shortcuts**

Models with **independence** assumption

cannot **represent** this!



$1 \quad (0,1) \quad (1,0)$

How to get **uniform distribution** over $(0,1),(1,0)$?

Marginal of both digits is 0.5

So under **independence**:

$$\mathrm{p}(\mathrm{c}_1, \mathrm{c}_2) = \mathrm{p}(\mathrm{c}_1)\mathrm{p}(\mathrm{c}_2) = 0.5 \cdot 0.5 = 0.25$$

And so...

$$\mathrm{p}(\mathrm{c}_1 = 0, \mathrm{c}_2 = 0) = \mathrm{p}(\mathrm{c}_1 = 1, \mathrm{c}_2 = 1) = 0.25$$

But that's **not** the uniform distribution!

van Krieken, E., Minervini, P., Ponti, E., & Vergari, A. (2025). Neurosymbolic Reasoning Shortcuts under the Independence Assumption. arXiv preprint arXiv:2507.11357.

Can we do be efficiently
RS aware within a single model?

Yes, by going beyond independence

- But we'll discuss that tomorrow :-)

Content

Introduction to Reasoning Shortcuts

Mitigation strategies

Awareness strategies

Secret side-track if we make it:
More on the independence assumption

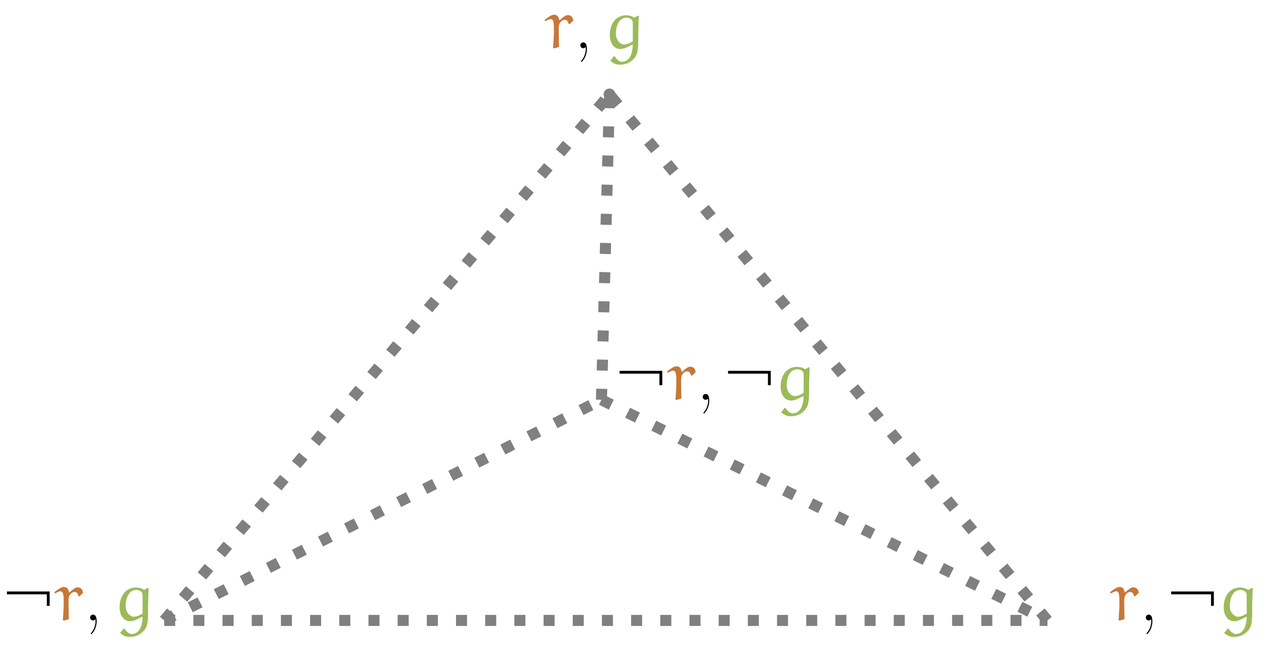
Knowledge: *'Traffic lights cannot be* ***both*** *red and green at the same time'*

How to correct the neural network?

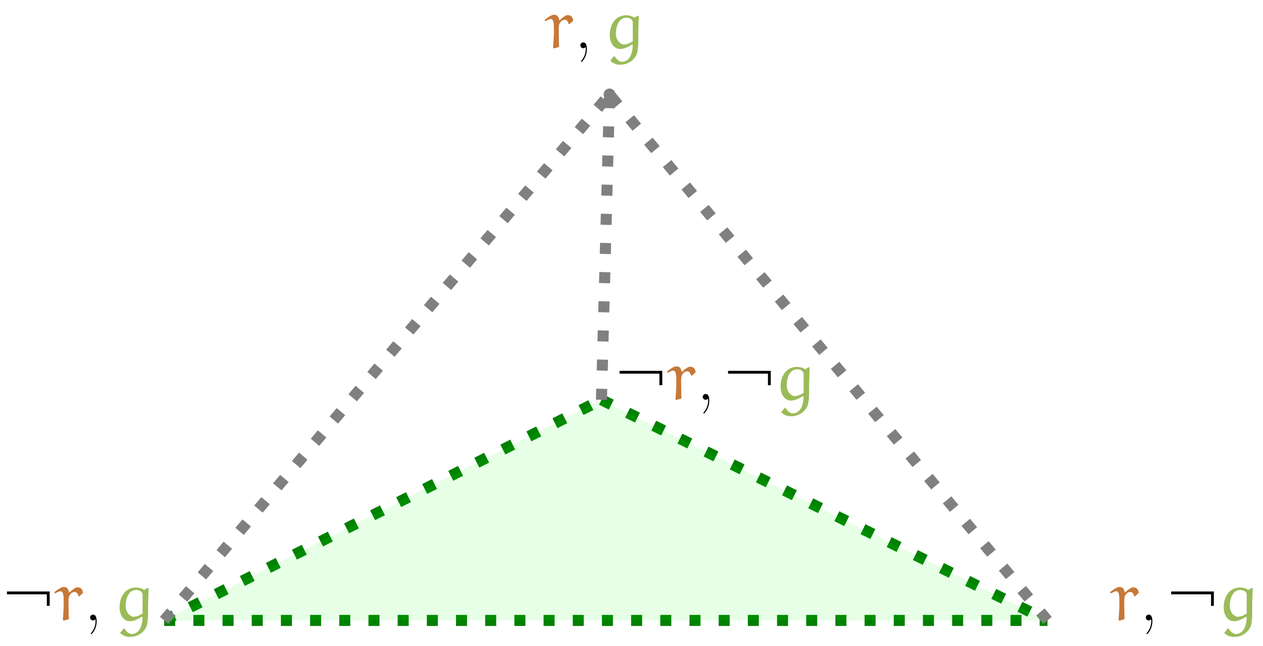
My opinion: **Equal belief** in

red light on

green light on

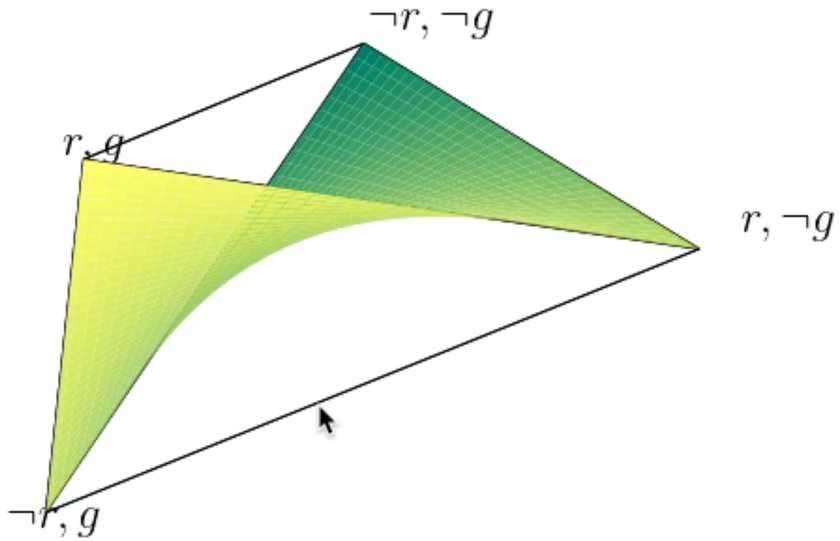


Each point in a **tetrahedron (a 3-dim simplex)**
represents a distribution / belief

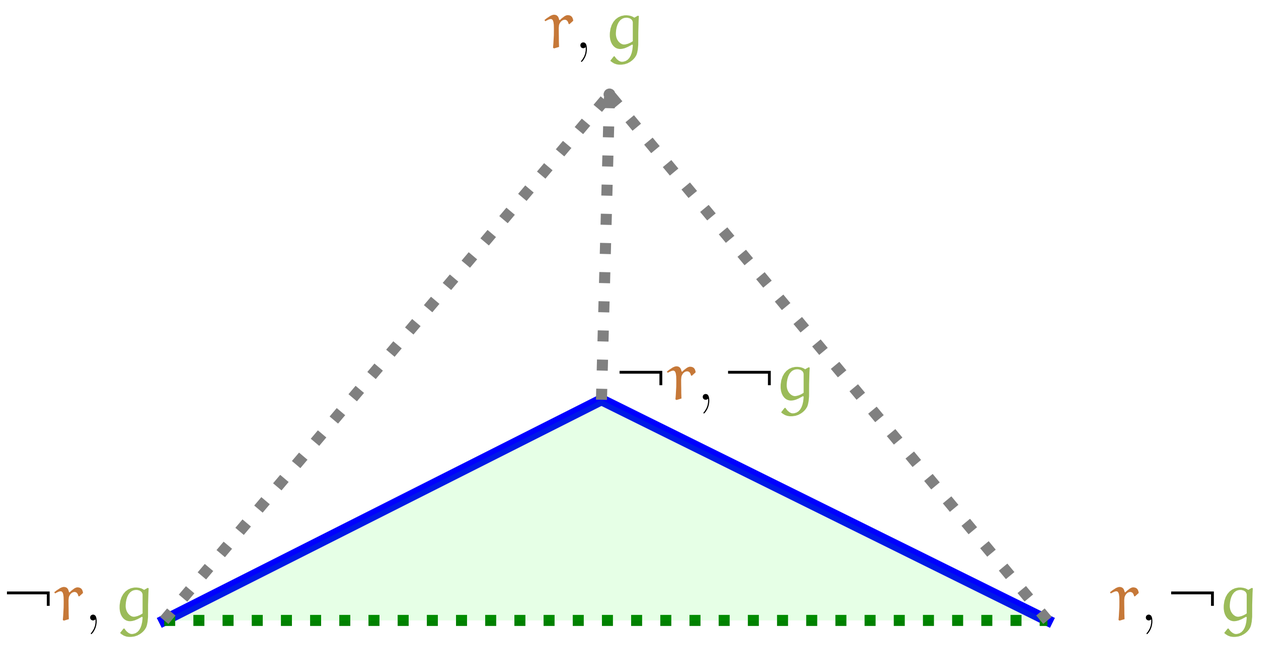


$$p_\theta(\text{red}, \text{green}) = 0$$

Independent distributions have
2 parameters, but live in a **3D space**!



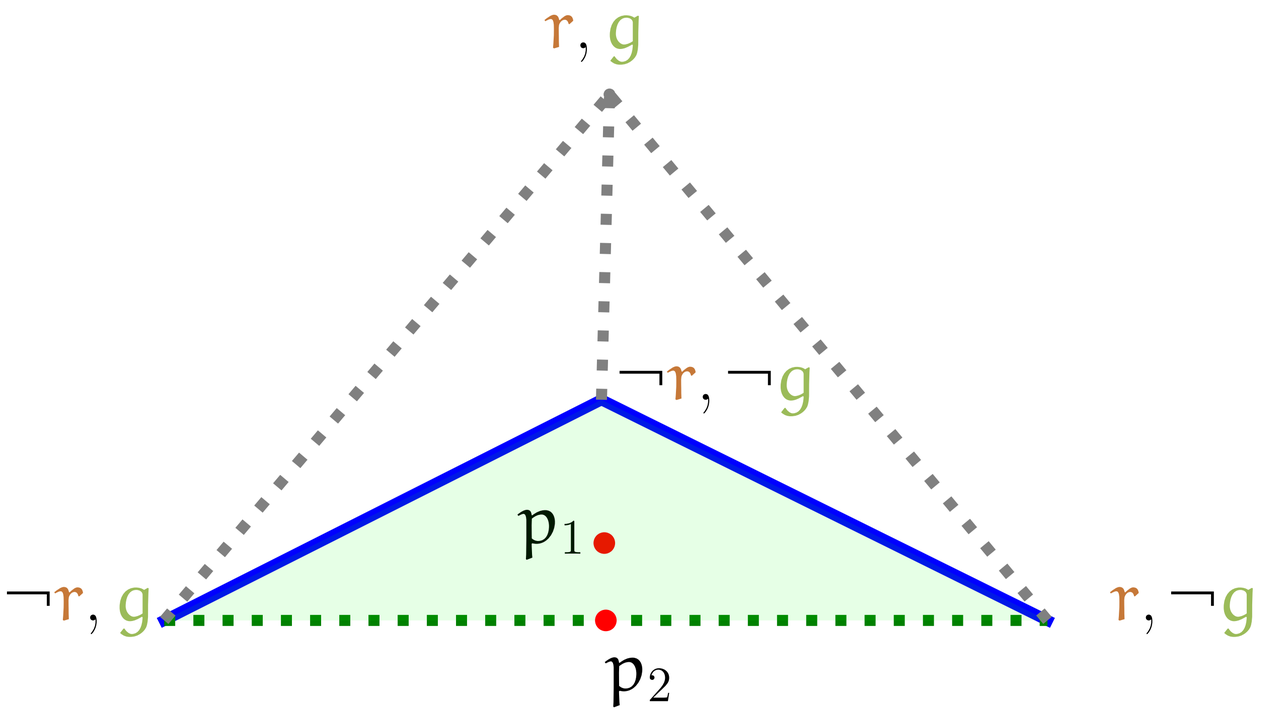
$$p_\theta(\text{red}, \text{green}) = p_\theta(\text{red})p_\theta(\text{green})$$



Under **independence**, the **minima** satisfy

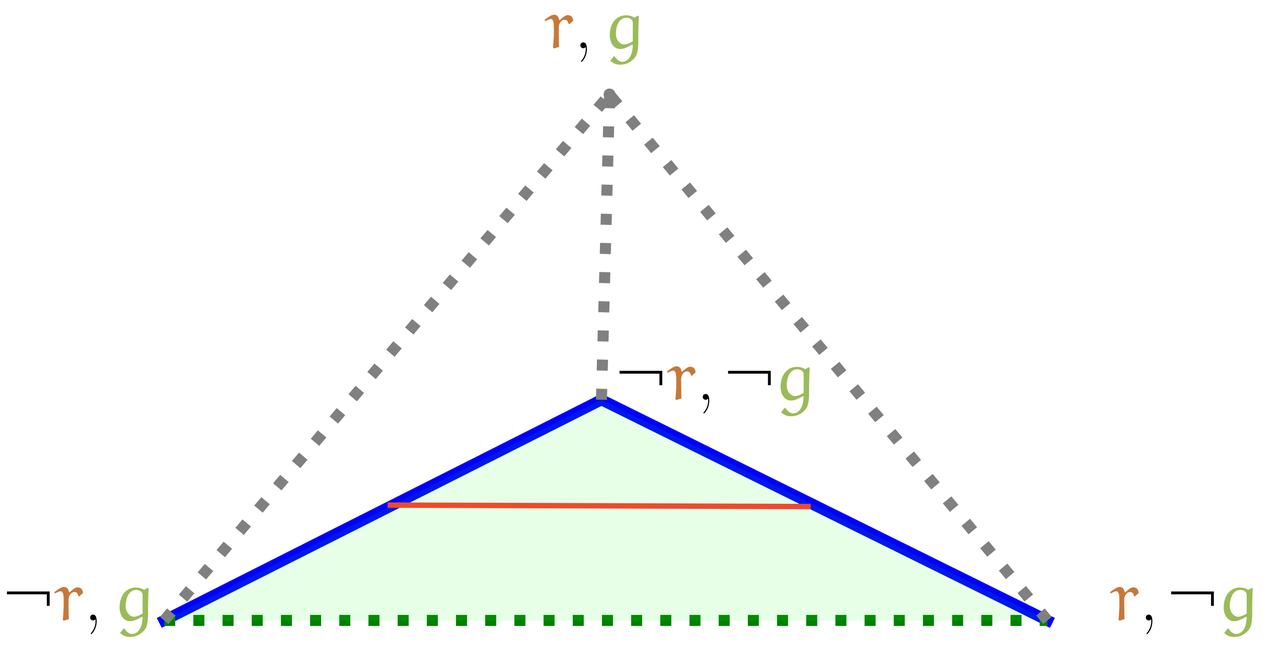
$$p_\theta(\text{red})p_\theta(\text{green}) = 0$$

"**At least one** of the probabilities is 0"



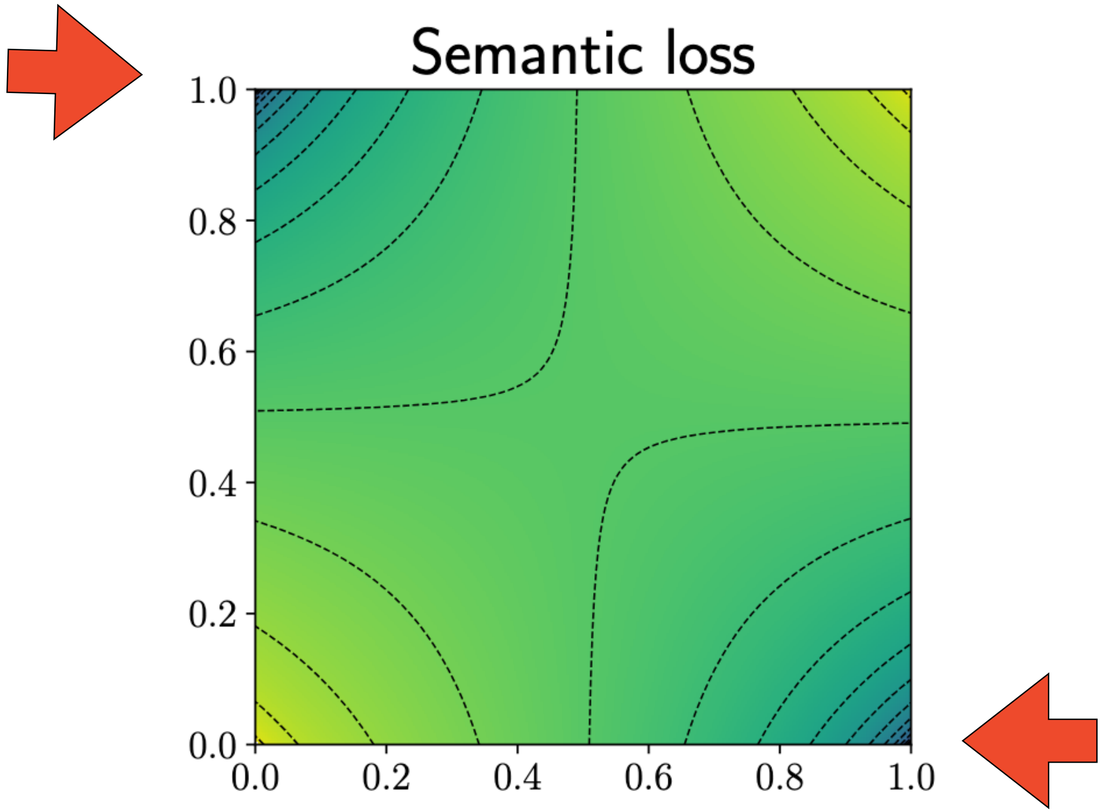
$$p_\theta(\text{red})p_\theta(\text{green}) = 0$$

"At least one of the probabilities is 0"



Minima are usually **non-convex**!

(Exact conditions in paper)



Minima are often **disconnected**!

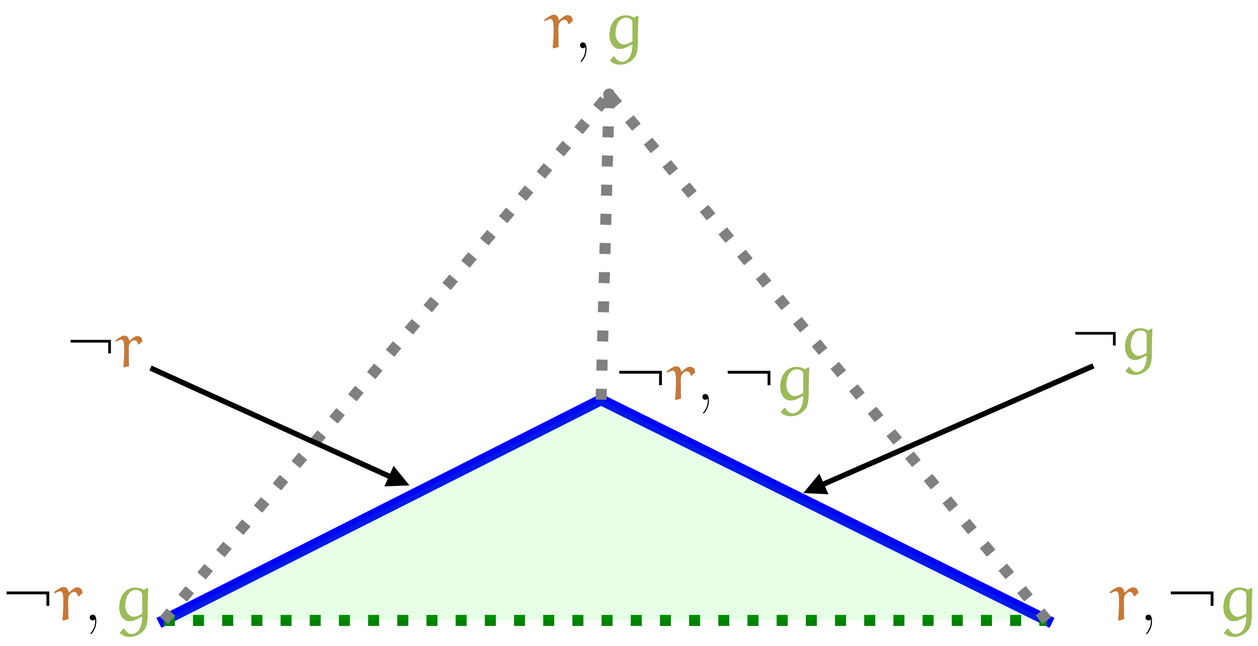
Examples: XOR and MNIST Addition

(Exact conditions in paper)

“Nice examples and all that…
but does it generalise?”

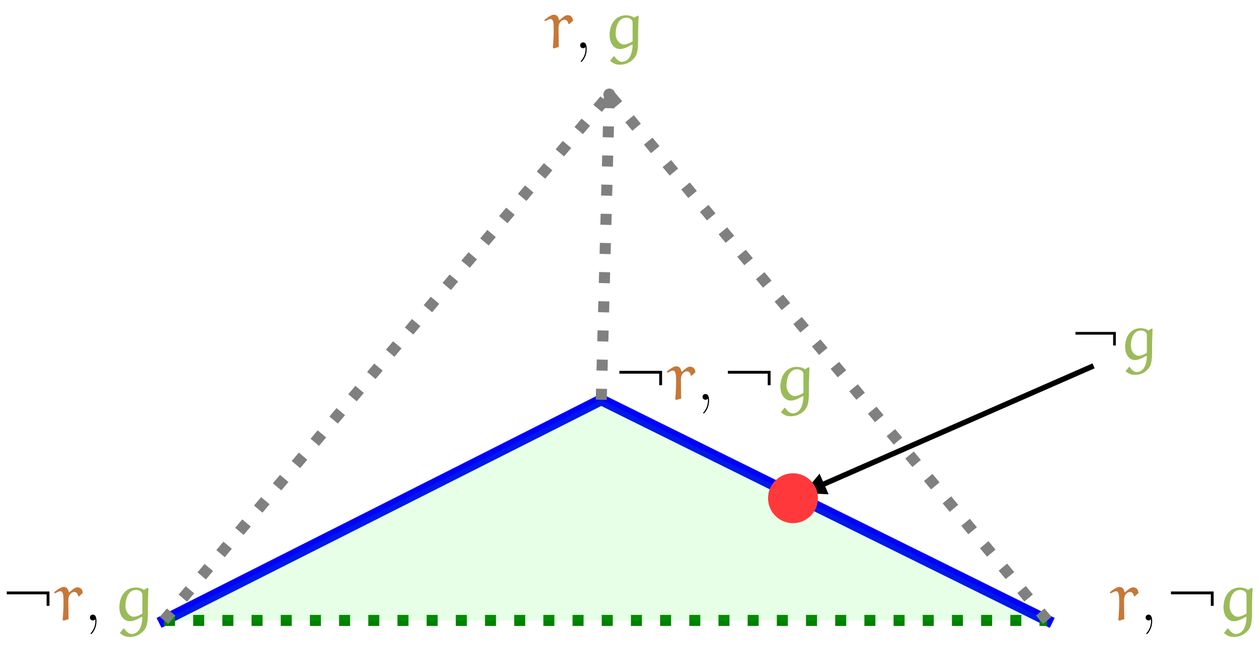
Yes! With *implicants*

$$\boxed{l_1 \wedge \ldots \wedge l_k} \models \phi$$



$$\phi = \neg r \vee \neg g$$

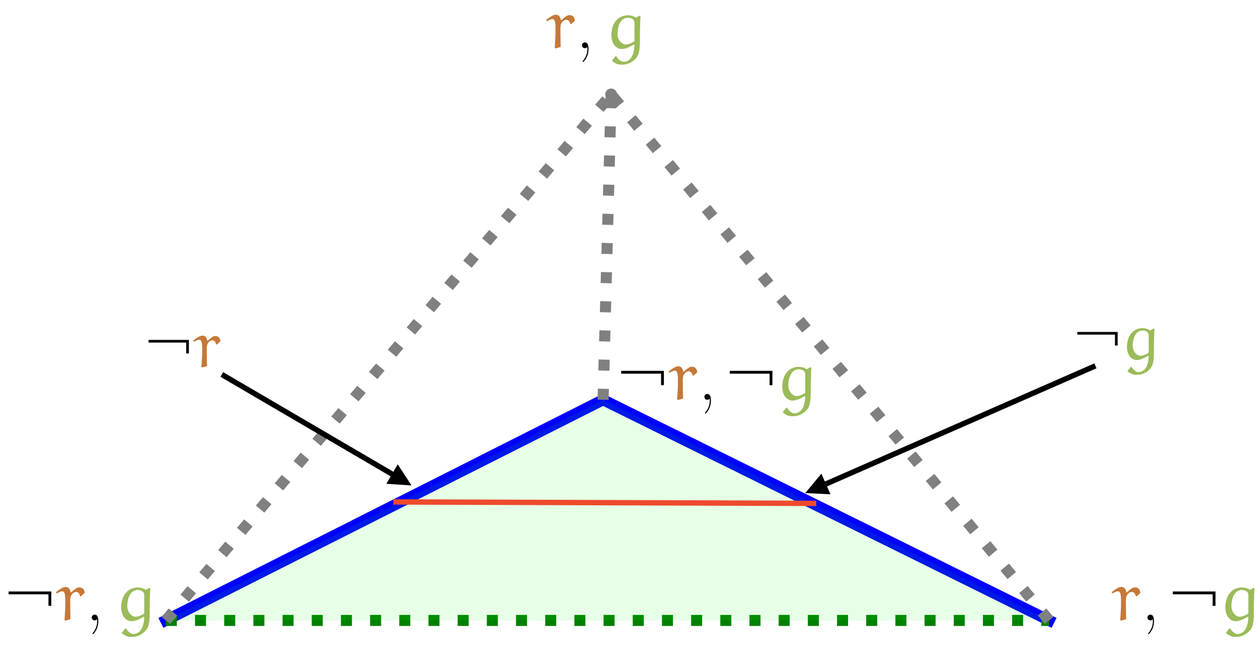
An **independent** distribution p satisfies ϕ iff p's **deterministic component** is an implicant.



$$p(r) = 0.5, p(g) = 0$$

Yes! Because $\neg g$ is an implicant

Convex *iff* there is *exactly one prime* implicant of the formula

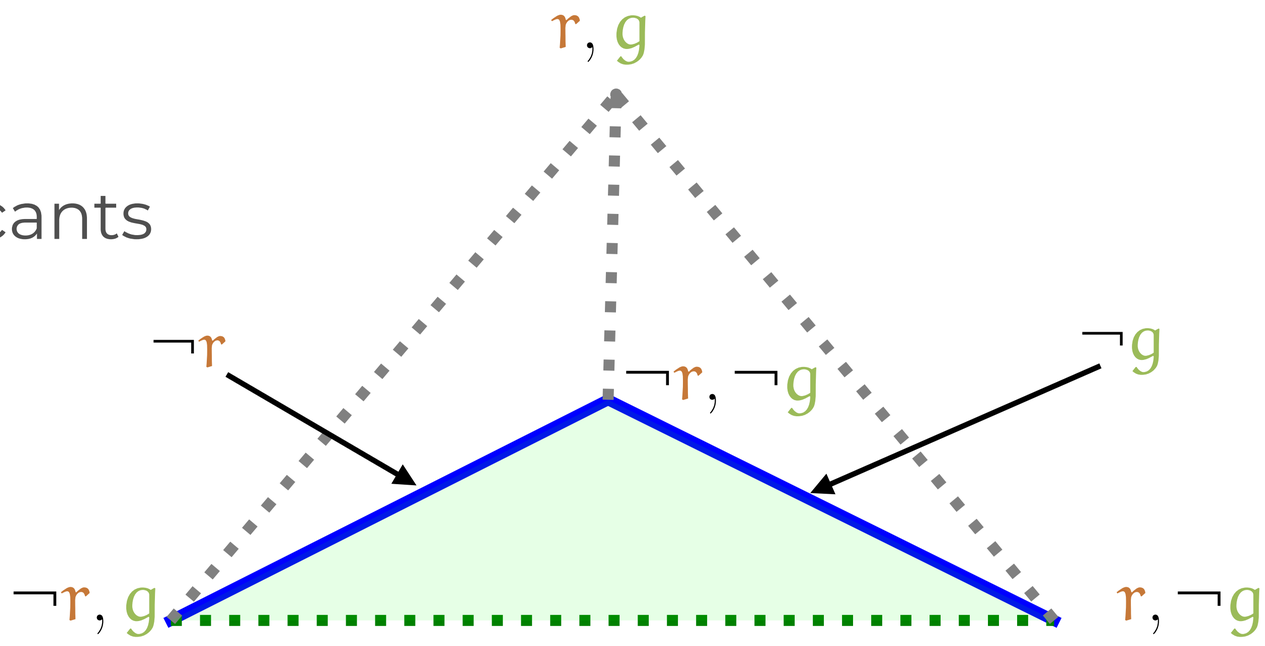


See ref for precise theorems

$$\phi = \neg r \vee \neg g$$

van Krieken, E., Minervini, P., Ponti, E., & Vergari, A. On the Independence Assumption in Neurosymbolic Learning. In Forty-first International Conference on Machine Learning.

Connected *iff* the prime implicants form a ‘connected’ graph



See ref for precise theorems

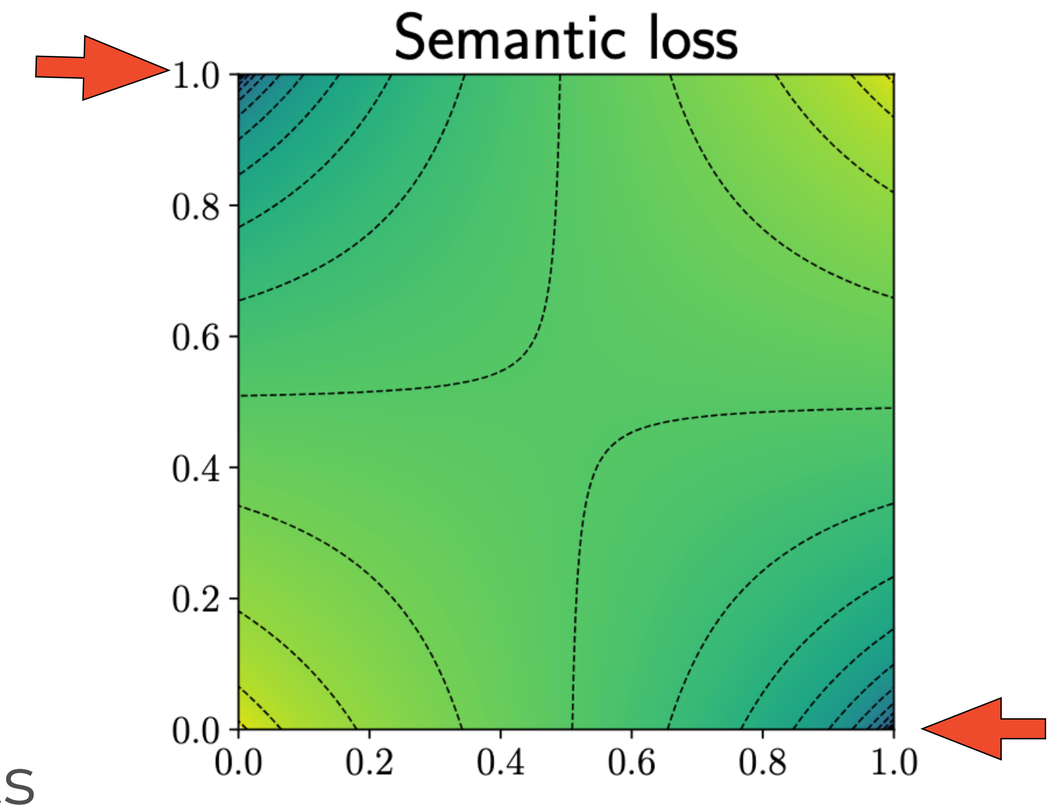
$$\phi = \neg r \vee \neg g$$

van Krieken, E., Minervini, P., Ponti, E., & Vergari, A. On the Independence Assumption in Neurosymbolic Learning. In Forty-first International Conference on Machine Learning.

We find the set of ***all*** independent distributions satisfying ϕ with the **prime** (*minimal*) **implicants** of ϕ

Convex *iff* there is ***exactly one*** prime implicant

Connected *iff* the prime implicants form a 'connected' graph

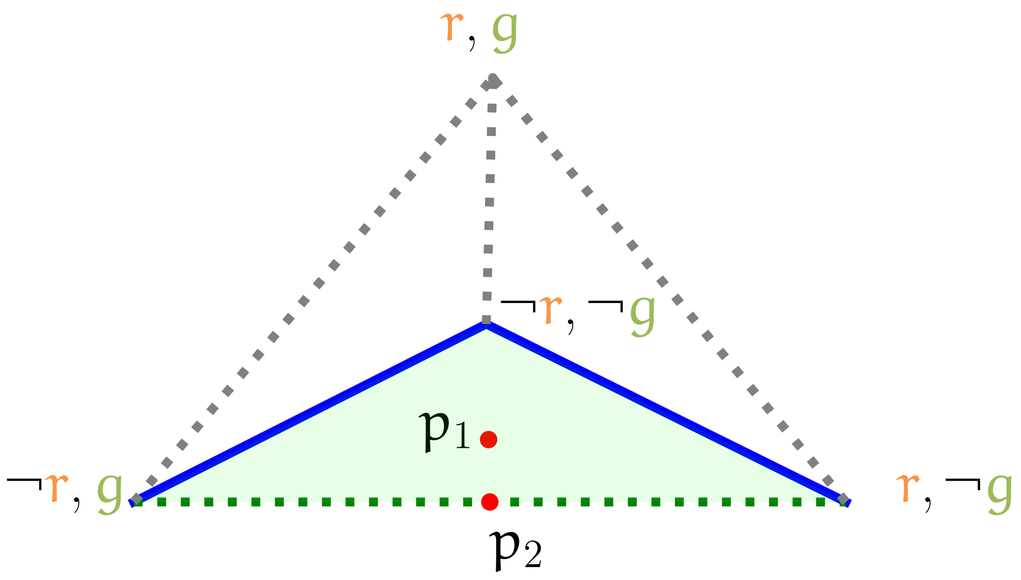


$\varphi = c_1 \oplus c_2$

1. **Non-convex** loss landscape
2. Cannot model **uncertainty**
3. Often **disconnected minima**

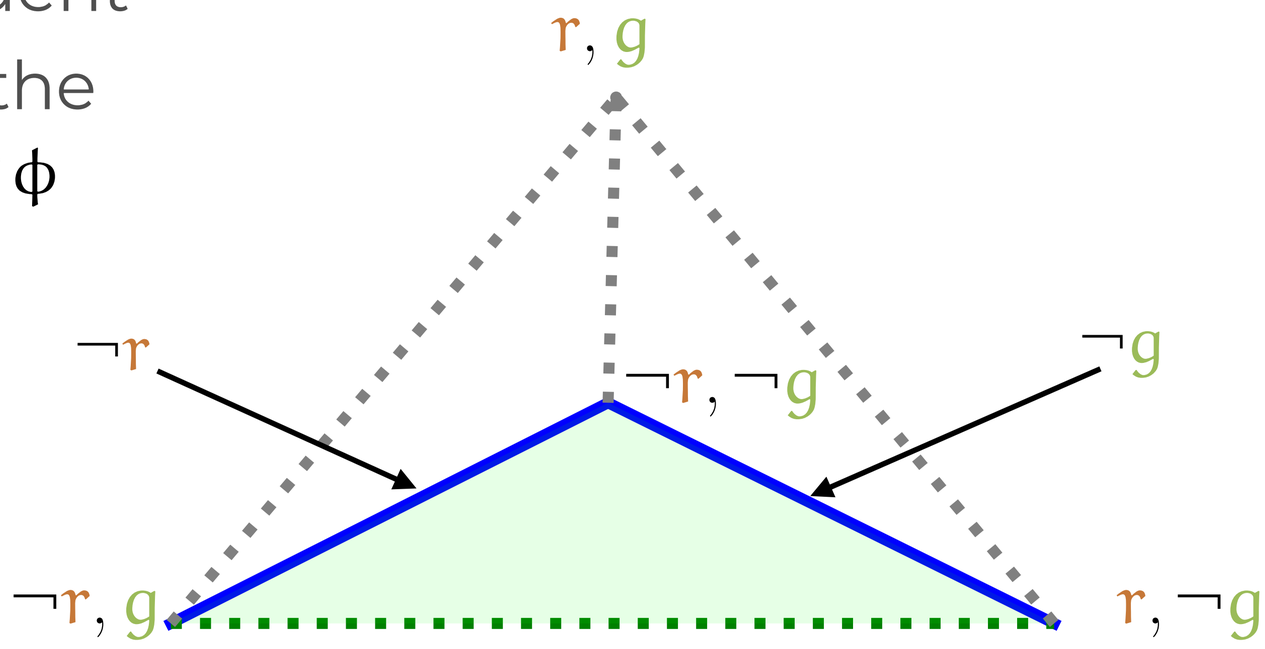
Challenging optimization!

By **relaxing** independence assumption, **these issues decrease**!



Emile van Krieken, Pasquale Minervini, Edoardo Ponti, & Antonio Vergari
On the Independence Assumption in Neurosymbolic Learning. ICML

We find the set of ***all*** independent distributions satisfying ϕ with the **prime** (*minimal*) **implicants** of ϕ



$$\phi = \neg r \vee \neg g$$

We find the set of ***all*** independent distributions satisfying ϕ with the **prime** (*minimal*) **implicants** of ϕ

A similar construction was already studied in the 50s!

Roth, J. Paul. "Algebraic topological methods for the synthesis of switching systems. I." Transactions of the American Mathematical Society 88.2 (1958): 301-326.

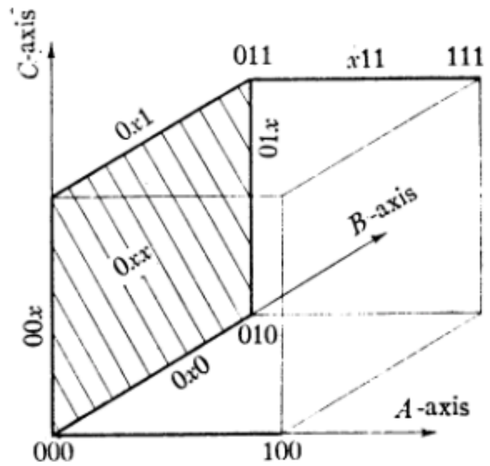


FIG. 1

$x00 = \overline{BC}$, $0xx = \overline{A}$, $x11 = BC$

Complementary complex is $001 \cup 101$.

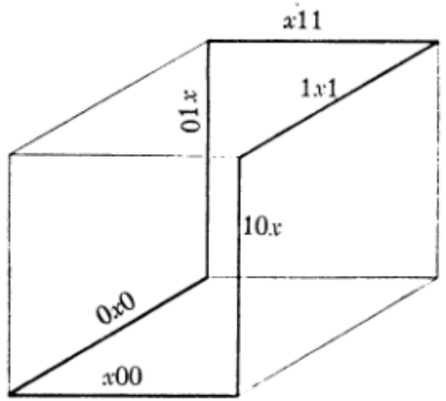


FIG. 2

$f = \overline{BC} \vee AC \vee \overline{A}B = A\overline{B} \vee BC \vee \overline{AC}$

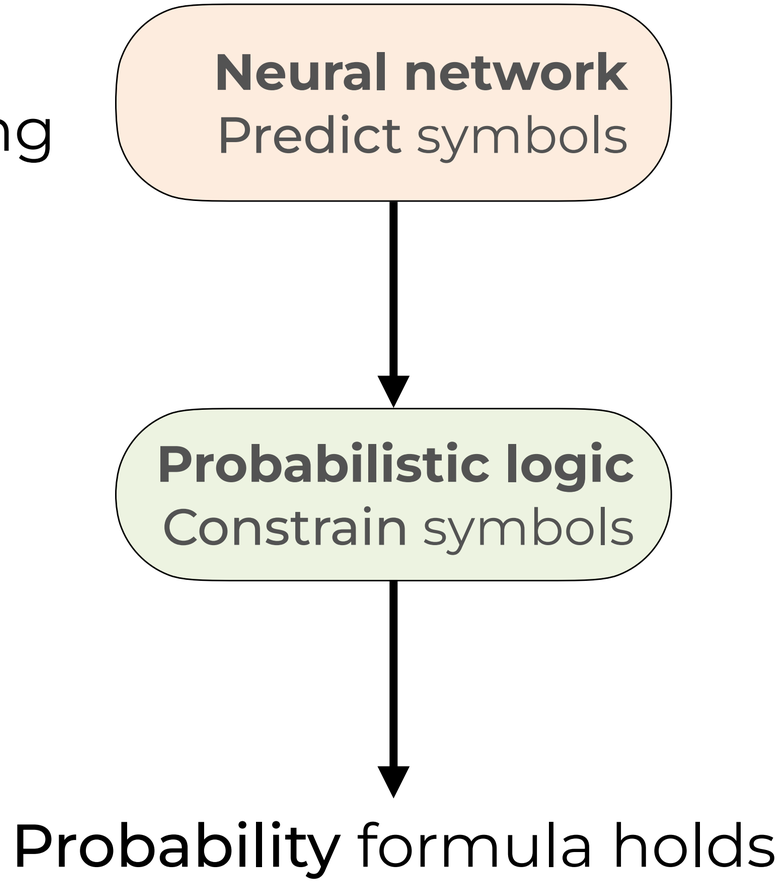
Reasoning shortcuts can complicate NeSy!

We will find solutions that look right by solving input-label mapping...

But completely learn the wrong concepts!

What to do:
- Mitigation
- Awareness

Tomorrow: Probabilistic reasoning
Go to the evening lecture! It will be great preparation for tomorrow :-)



Contact: Emile.van.Krieken@ed.ac.uk *Bluesky:* @EmilevanKrieken.com