Learning Interpretable Concepts in Deep Learning: Desiderata from Causality and Neuro-Symbolic Reasoning Shortcuts

Emanuele Marconato[♣,♠]
[♣]DISI, University of Trento, [♠]University of Pisa

SOTA Deep Learning models are Black-boxes

✓ High-dimensional, sub-symbolic inputs

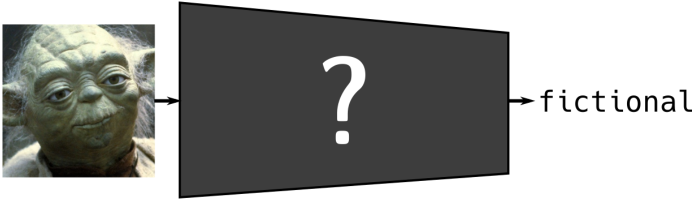
✓ Learn from IID

✓ High performance in specific tasks

✗ Not self-explainable

✗ Cannot learn over time

✗ Bad performances on OOD generalization



Concept in Explainable AI

Concept Bottleneck Models

Pang Wei Koh [*1] **Thao Nguyen** [*1 2] **Yew Siang Tang** [*1]
Stephen Mussmann [1] **Emma Pierson** [1] **Been Kim** [2] **Percy Liang** [1]

Abstract

We seek to learn models that we can interact with using high-level concepts: if the model did not think there was a bone spur in the x-ray, would it still predict severe arthritis? State-of-the-art models today do not typically support the manipulation of concepts like "the existence of bone spurs", as they are trained end-to-end to go directly from raw input (e.g., pixels) to output (e.g., arthritis severity). We revisit the classic idea of first predicting concepts that are provided at training time, and then using these concepts to predict the label. By construction, we can intervene on these *concept bottleneck models* by editing their predicted concept values and propagating these changes to the final prediction. On x-ray grading and bird identification, concept bottleneck models achieve competitive accuracy with standard end-to-end models, while enabling interpretation in terms of high-level clinical concepts ("bone spurs") or bird attributes ("wing color"). These

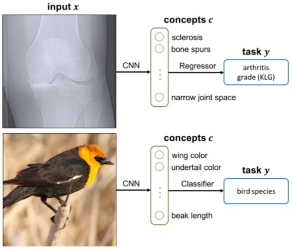


Figure 1. We study concept bottleneck models that first predict an intermediate set of human-specified concepts c, then use c to predict the final output y. We illustrate the two applications we consider: knee x-ray grading and bird identification.

(Source: [1] Also: Concept Whitening [2])

Current Strategies for Interpretability

- Concepts are "sparse"
- Concepts are "orthogonal"
- Concepts "activate highly" on concrete training examples
- Concepts "activate highly" on *parts of* training examples
- ...

Problems

- Learn with label and concept supervision (concepts are specific for the application)

✗ The label/concept accuracy trade-off (supervision on concepts?)

✗ Spurious correlations in the learned concepts (aka **Concept Leakage**)

Promises and Pitfalls of Black-Box Concept Learning Models

Anita Mahinpei [*1] Justin Clark [*1] Isaac Lage [1] Finale Doshi-Velez [1] Weiwei Pan [1]

Abstract

Machine learning models that incorporate concept learning as an intermediate step in their decision making process can match the performance of black-box predictive models while retaining the ability to explain outcomes in human understandable terms. However, we demonstrate that the concept representations learned by these models encode information beyond the pre-defined concepts, and that natural mitigation strategies do not fully work, rendering the interpretation of the downstream prediction misleading. We describe the mechanism underlying the information leakage and suggest recourse for mitigating its effects.

Losch et al. (2019)). In each case, the neural network model learns to map raw input to concepts and then map those concepts to predictions. We call the mapping from input to concepts a Concept Learning Model (CLM), although this mapping may not always be trained independently from the downstream prediction task. Models that incorporate a CLM component have been shown to match the performance of complex black-box prediction models while retaining the interpretability of decisions based on human understandable concepts, since for these models, one can explain the model decision in terms of intermediate concepts.

Unfortunately, recent work noted that black-box CLMs do not learn as expected. Specifically, Margeloiu et al. (2021) demonstrate that outputs of CLMs used in Concept Bottle-

Do Concept Bottleneck Models Learn As Intended?

Andrei Margeloiu* University of Cambridge am2770@cam.ac.uk

Matthew Ashman* University of Cambridge mca39@cam.ac.uk

Umang Bhatt* University of Cambridge usb20@cam.ac.uk

Yanzhi Chen University of Edinburgh

Mateja Jamnik University of Cambridge

Adrian Weller University of Cambridge, The Alan Turing Institute

ABSTRACT

Concept bottleneck models map from raw inputs to concepts, and then from concepts to targets. Such models aim to incorporate pre-specified, high-level concepts into the learning procedure, and have been motivated to meet three desiderata: interpretability, predictability, and intervenability. However, we find that concept bottleneck models struggle to meet these goals. Using post hoc interpretability methods, we demonstrate that concepts do not correspond to anything semantically meaningful in input space, thus calling into question the usefulness of concept bottleneck models in their current form.

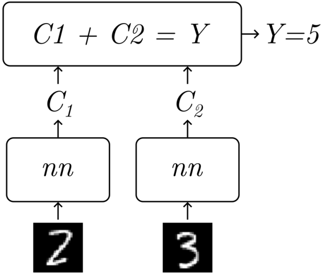
(Source: [3, 4])

Neuro-Symbolic models make also use of concepts

- **Neuro-Symbolic** (NeSy) models/predictors are deemed to be **trustworthy** due to compliance to **prior-knowledge**.

Neuro-Symbolic models make also use of concepts

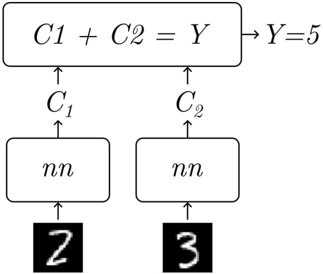
- **Neuro-Symbolic** (NeSy) models/predictors are deemed to be **trustworthy** due to compliance to **prior-knowledge**.



(1) **Perception:** Extract (binary) **concepts** to be input for the reasoning

(2) **Reasoning:** Perform probabilistic reasoning from the **concepts**

- **Neuro-Symbolic** (NeSy) models/predictors are deemed to be **trustworthy** due to compliance to **prior-knowledge**.



(1) **Perception:** Extract (binary) **concepts** to be input for the reasoning

(2) **Reasoning:** Perform probabilistic reasoning from the **concepts**

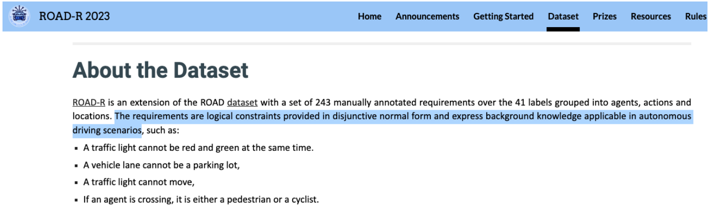
- Changing the knowledge we can use the same concepts to solve different tasks.

Trustworthiness claim

- NeSy predictors are learned with **input-output** samples (typically no concept supervision)

- **Claim:** this is sufficient to solve the *symbol grounding problem*:

  integrating the knowledge $\implies$ recovering the intended concepts ? (**False**)

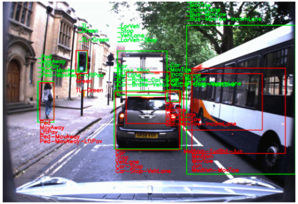

ROAD-R 2023

Home Announcements Getting Started Dataset Prizes Resources Rules

About the Dataset

ROAD-R is an extension of the ROAD dataset with a set of 243 manually annotated requirements over the 41 labels grouped into agents, actions and locations. The requirements are logical constraints provided in disjunctive normal form and express background knowledge applicable in autonomous driving scenarios, such as:

- A traffic light cannot be red and green at the same time.
- A vehicle lane cannot be a parking lot,
- A traffic light cannot move,
- If an agent is crossing, it is either a pedestrian or a cyclist.



How to make sure the learned concepts/representations are interpretable?

1. **Interpretability in Representation Learning.** What is an interpretable representation?
2. **Reasoning Shortcuts (in NeSy AI).** Do NeSy AI models learn the intended concepts?

Our Work

Accepted at NeurIPS 2022

GlanceNets: Interpretable, Leak-proof Concept-based Models

Emanuele Marconato
Department of Computer Science
University of Pisa & University of Trento
Pisa, Italy
emanuele.marconato@unitn.it

Andrea Passerini
Department of Computer Science
University of Trento
Trento, Italy
andrea.passerini@unitn.it

Stefano Teso
Department of Computer Science
University of Trento
Trento, Italy
stefano.teso@unitn.it

Published in MDPI Entropy

entropy MDPI

Article

Interpretability Is in the Mind of the Beholder: A Causal Framework for Human-Interpretable Representation Learning

Emanuele Marconato [1,2], Andrea Passerini [1] and Stefano Teso [1,3,*]

1 Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento, 38123 Trento, Italy; emanuele.marconato@unitn.it (E.M.); andrea.passerini@unitn.it (A.P.)
2 Dipartimento di Informatica, University of Pisa, 56126 Pisa, Italy
3 Centro Interdipartimentale Mente/Cervello, University of Trento, 38123 Trento, Italy
* Correspondence: stefano.teso@unitn.it

Accepted at ICML 2023

Neuro-Symbolic Continual Learning: Knowledge, Reasoning Shortcuts and Concept Rehearsal

Emanuele Marconato [*1 2] Gianpaolo Bontempo [*1 3] Elisa Ficarra [3] Simone Calderara [3] Andrea Passerini [2] Stefano Teso [4 2]

Accepted at NeurIPS 2023

Not All Neuro-Symbolic Concepts Are Created Equal: Analysis and Mitigation of Reasoning Shortcuts

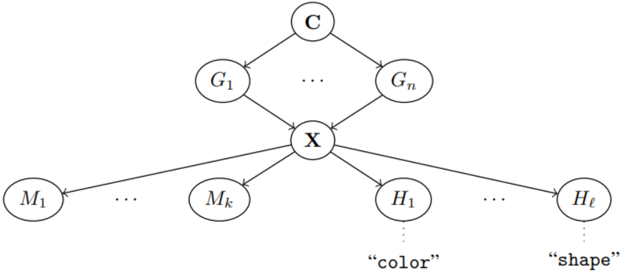
Emanuele Marconato
DISI and DI
University of Trento and University of Pisa
Trento, Italy
emanuele.marconato@unitn.it

Stefano Teso
CIMeC and DISI
University of Trento
Trento, Italy
stefano.teso@unitn.it

Antonio Vergari
School of Informatics
University of Edinburgh
Edinburgh, UK
avergari@exseed.ed.ac.uk

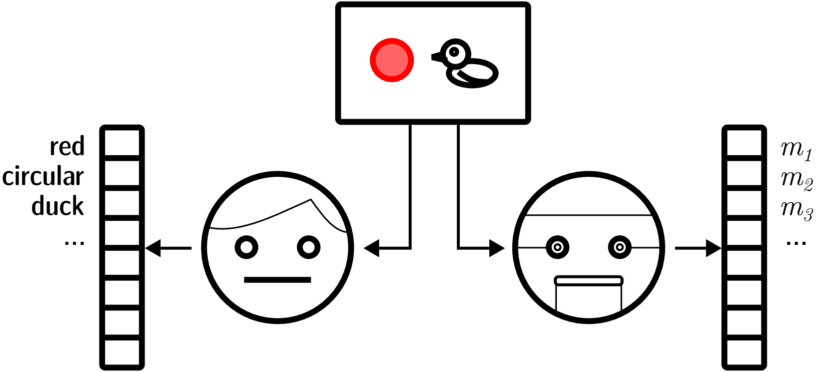
Andrea Passerini
DISI
University of Trento
Trento, Italy
andrea.passerini@unitn.it

Interpretability of the concepts

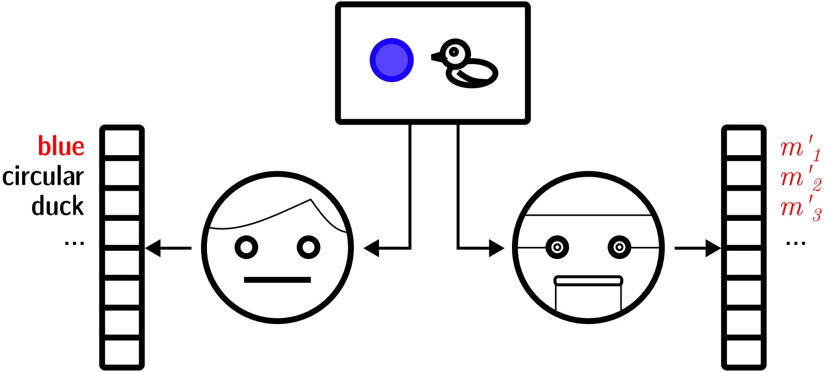


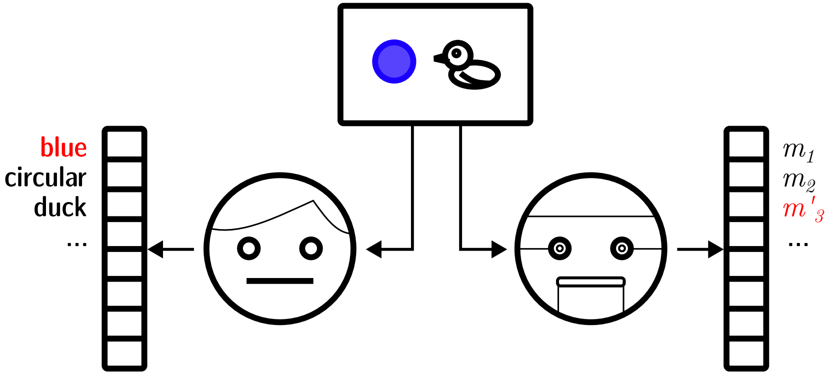
What is **concept interpretability**? If any change a makes to their **mental representation** impacts the machine representation in a way that they can **understand**, they **the two concepts share the same name**.

- The formalization includes the **human**
- Alignment for **non-disentangled repr**
- Leakage is lack of **context-style separation**
- Link to **causal abstractions**

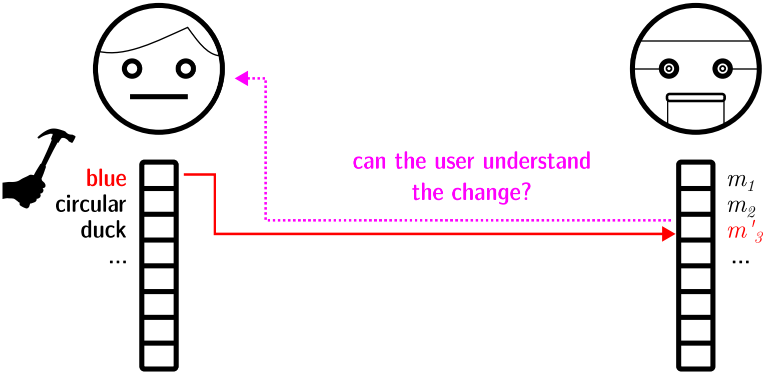


red
circular
duck
...
m_1
m_2
m_3
...


blue
circular
duck
...
m'_1
m'_2
m'_3
...



blue
circular
duck
...
m_1
m_2
m'_3
...



blue
circular
duck
...
can the user understand
the change?
m_1
m_2
m'_3
...

■ Let's formalize what we are doing:

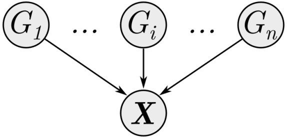
□ Several **generative factors** $\mathbf{G} = (G_1, \ldots, G_n)$, encoding `hair color`, `age`, `complexion`, $\ldots$

Generative process, adapted from [5].

■ Let's formalize what we are doing:

☐ Several **generative factors** $\mathbf{G} = (G_1, \dots, G_n)$, encoding `hair color`, `age`, `complexion`, ...

☐ They jointly cause an **observation** $\mathbf{X}$, e.g., the Yoda image



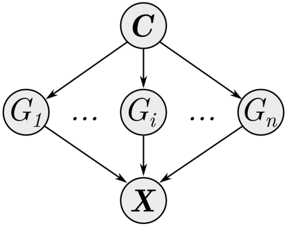
Generative process, adapted from [5].

- Let's formalize what we are doing:

☐ Several **generative factors** $\mathbf{G} = (G_1, \ldots, G_n)$, encoding `hair color`, `age`, `complexion`, ...

☐ They jointly cause an **observation** $\mathbf{X}$, e.g., the Yoda image

☐ May be correlated through hidden **confounds** $\mathbf{C}$, but can be independently manipulated



Generative process, adapted from [5].

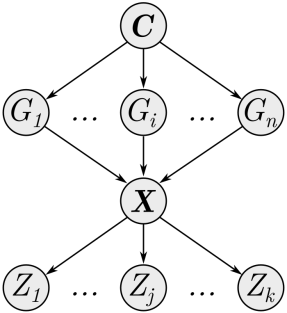
- Let's formalize what we are doing:

- Several **generative factors** $\mathbf{G} = (G_1, \ldots, G_n)$, encoding `hair color`, `age`, `complexion`, ...

- They jointly cause an **observation** $\mathbf{X}$, e.g., the Yoda image

- May be correlated through hidden **confounds** $\mathbf{C}$, but can be independently manipulated

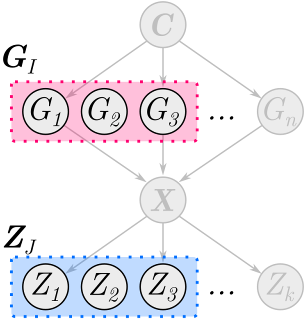
- Model acquires **latent factors** $Z_1, \ldots, Z_k$



Generative process, adapted from [5].

■ Communication is possible $\Leftrightarrow$ same **semantics**

☐ Assume subset of generative factors $\mathbf{G}_I$ are **understandable** to a human agent. For instance, G_1 = age, G_2 = hair color, G_3 = complexion.



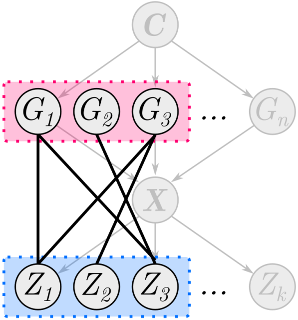
Interpretability as alignment.

Interpretability as Alignment

- Communication is possible $\Leftrightarrow$ same **semantics**

☐ Assume subset of generative factors $\mathbf{G}_I$ are **understandable** to a human agent. For instance, G_1 = age, G_2 = hair color, G_3 = complexion.

☐ CBM implicitly learns a **map** $\alpha : \mathbf{G}_{1:3} \mapsto \mathbf{Z}_{1:3}$



Interpretability as alignment.

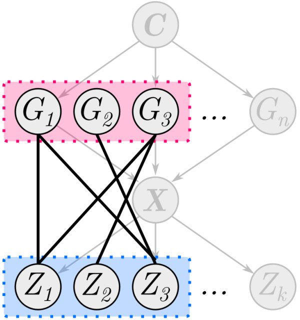
Interpretability as Alignment

- Communication is possible $\Leftrightarrow$ same **semantics**

- ☐ Assume subset of generative factors $\mathbf{G}_I$ are **understandable** to a human agent. For instance, G_1 = age, G_2 = hair color, G_3 = complexion.

- ☐ CBM implicitly learns a **map** $\alpha : \mathbf{G}_{1:3} \mapsto \mathbf{Z}_{1:3}$

- **Example**: if α is **the identity**, then $\mathbf{Z}_{1:3}$ are *by construction* interpretable!



Interpretability as alignment.

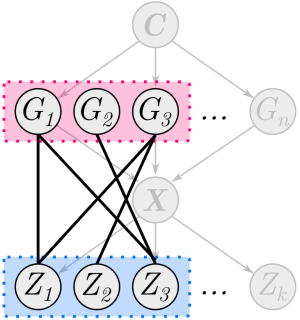
Interpretability as Alignment

- Communication is possible $\Leftrightarrow$ same **semantics**

☐ Assume subset of generative factors $\mathbf{G}_I$ are **understandable** to a human agent. For instance, G_1 = age, G_2 = hair color, G_3 = complexion.

☐ CBM implicitly learns a **map** $\alpha : \mathbf{G}_{1:3} \mapsto \mathbf{Z}_{1:3}$

- **Example**: if α is **permutes elements**, then $\mathbf{Z}_{1:3}$ are *by construction* interpretable!



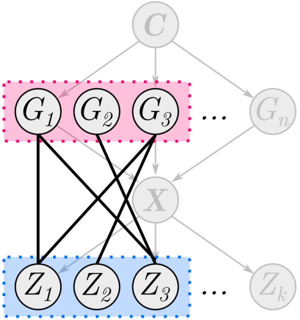
Interpretability as alignment.

- Communication is possible $\Leftrightarrow$ same **semantics**

☐ Assume subset of generative factors $\mathbf{G}_I$ are **understandable** to a human agent. For instance, $G_1 =$ age, $G_2 =$ hair color, $G_3 =$ complexion.

☐ CBM implicitly learns a **map** $\alpha : \mathbf{G}_{1:3} \mapsto \mathbf{Z}_{1:3}$

- **Example**: If α is **rescales individual elements**, then $\mathbf{Z}_{1:3}$ are *by construction* interpretable!



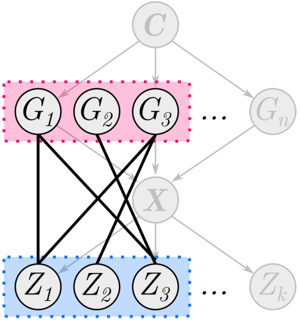
Interpretability as alignment.

- Communication is possible $\Leftrightarrow$ same **semantics**

- Assume subset of generative factors $\mathbf{G}_I$ are **understandable** to a human agent. For instance, G_1 = age, G_2 = hair color, G_3 = complexion.

- CBM implicitly learns a **map** $\alpha : \mathbf{G}_{1:3} \mapsto \mathbf{Z}_{1:3}$

- *How much can we push?*



Interpretability as alignment.

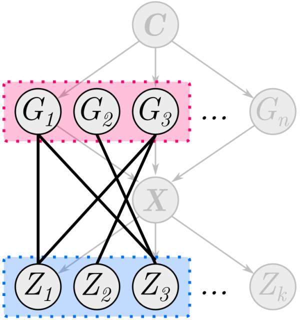
Interpretability as Alignment

- Communication is possible $\Leftrightarrow$ same **semantics**

- Assume subset of generative factors $\mathbf{G}_I$ are **understandable** to a human agent. For instance, G_1 = age, G_2 = hair color, G_3 = complexion.

- CBM implicitly learns a **map** $\alpha : \mathbf{G}_{1:3} \mapsto \mathbf{Z}_{1:3}$

- *How much can we push?* Let's be **conservative**



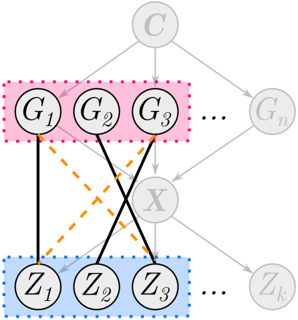
Interpretability as alignment.

Alignment

The map *alpha* from ground-truth concepts to machine concepts:

1. **mixes no** two generative factors into the same learned concept (**disentanglement**)

2. is **elementwise monotonic**.



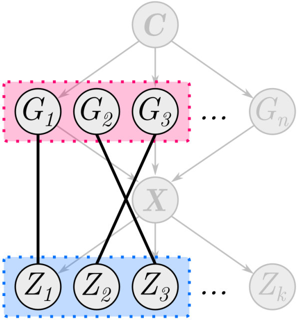
Interpretability as alignment.

Alignment

The map *alpha* from ground-truth concepts to machine concepts:

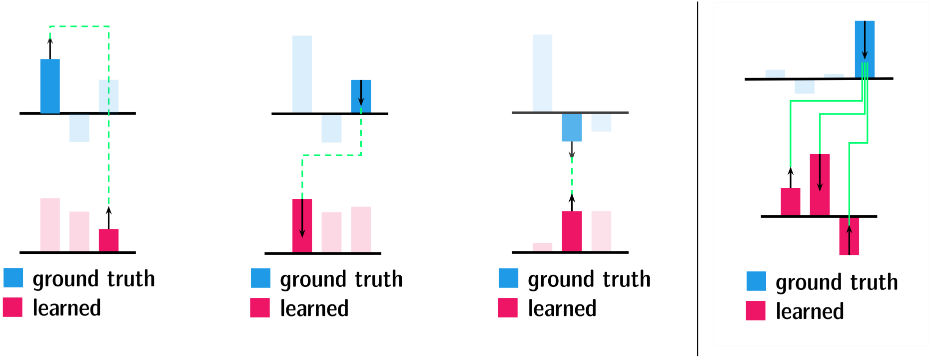
1. **mixes no** two generative factors into the same learned concept (**disentanglement**)

2. is **elementwise monotonic**.



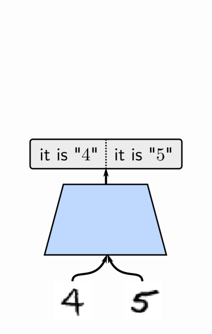
Interpretability as alignment.

Effect of *intervening* on observed $\mathbf{G}_I$ or unobserved $\mathbf{G}_{-I}$ factors:

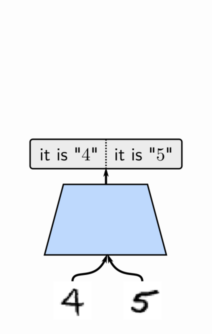


2 Concept Leakage = Unintended Semantics

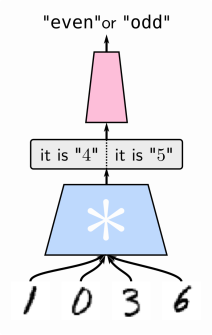
1 Fit two concepts to recognize `MNIST` images of "4"s and "5"s using **full concept annotations**



① Fit two concepts to recognize `MNIST` images of "4"s and "5"s using **full concept annotations**

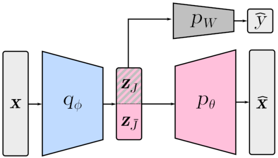


② Use the learned concepts, predict parity of **remaining digits** (i.e., all except "4" and "5")

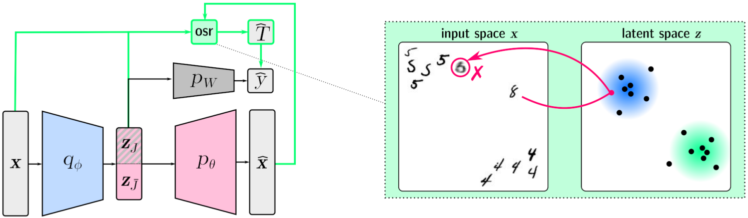


- Performance is **much better than random**!

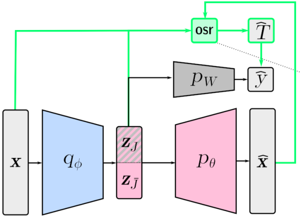
GlanceNets = VAE + concept supervision

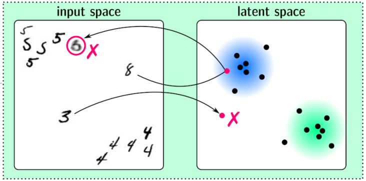


GlanceNets = VAE + concept supervision + **open-set rejection**



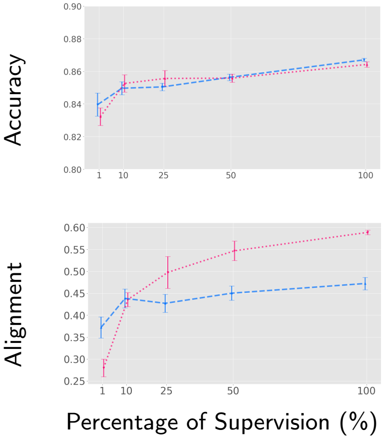
GlanceNets = VAE + concept supervision + **open-set rejection**





GlanceNets Foster Alignment

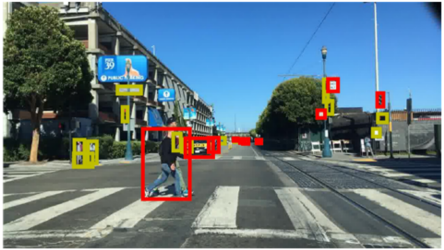
- Metrics:
  - Accuracy
  - Alignment (linear DCI)
- **CelebA** w/ supervision for 6 generative factors, realistic labels obtained via clustering.
- Same # of concepts for both GlanceNets and Concept Bottleneck models [4].



Learning the concepts/symbols in Neuro-Symbolic AI

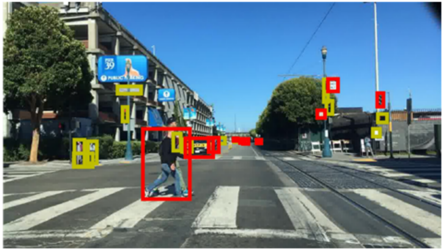
We study predictions where symbolic knowledge K must be satisfied, *e.g.*, autonomous driving with **safety constraints**

We study predictions where symbolic knowledge K must be satisfied,
e.g., autonomous driving with **safety constraints**



→ *Faster-RCNN*

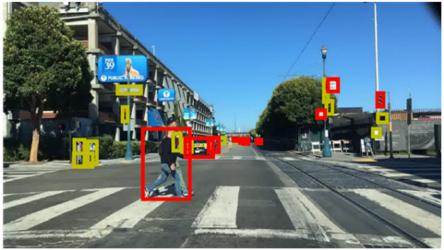
We study predictions where symbolic knowledge K must be satisfied,
e.g., autonomous driving with **safety constraints**



→ *Faster-RCNN* →

bb1: Pedestrian
bb2: Red light
bb3: Car (far)

We study predictions where symbolic knowledge K must be satisfied,
e.g., autonomous driving with **safety constraints**



→ *Faster-RCNN*

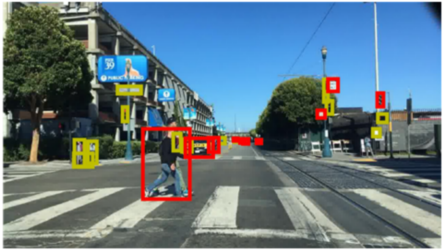
↓

bb1: Pedestrian
bb2: Red light
bb3: Car (far)

← K

$K = (\text{pedestrian} \lor \text{red} \Rightarrow \text{stop})$

We study predictions where symbolic knowledge K must be satisfied,
e.g., autonomous driving with **safety constraints**



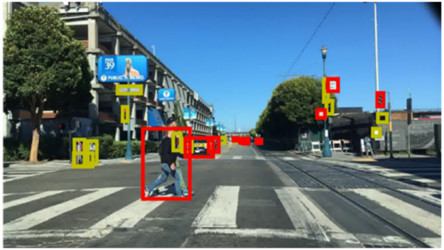
→ *Faster-RCNN* ↓

bb1: Pedestrian
bb2: Red light
bb3: Car (far)

STOP ← K ←

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

We study predictions where symbolic knowledge K must be satisfied,
e.g., autonomous driving with **safety constraints**



Faster-RCNN

bb1: Pedestrian
bb2: Red light
bb3: Car (far)

K

STOP

$$K = (\text{pedestrian} \lor \text{red} \Rightarrow \text{stop})$$

Learning on (**input**,**output**) samples (annotation on **concepts** is costly!)
like DeepProbLog [6], Semantic Loss [7], and Logic Tensor Networks [8]

Probably, you'd expect that...

Probably, you'd expect that...

- Knowledge + supervision on labels constrain the concepts to acquire the **right semantics**, i.e., to be grounded correctly.

Probably, you'd expect that...

- Knowledge + supervision on labels constrain the concepts to acquire the **right semantics**, i.e., to be grounded correctly.

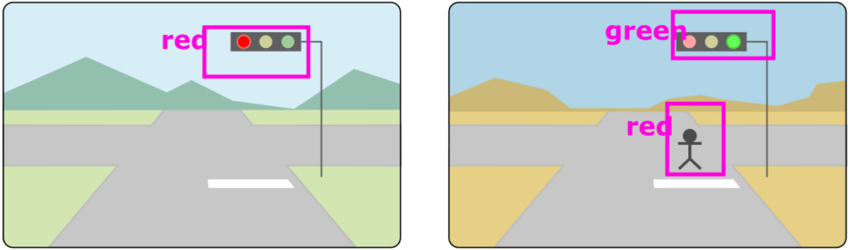
> *"if my NeSy model predicts correct actions in all examples of autonomous driving,* ***then the concepts are good****! For instance,* `red_light` $= \top$ ***iff*** *there is a red traffic light in the dashcam image!"*

- Knowledge + supervision on labels constrain the concepts to acquire the **right semantics**, i.e., to be grounded correctly.

  *"if my NeSy model predicts correct actions in all examples of autonomous driving, **then the concepts are good**! For instance,* `red_light` $= \top$ ***iff*** *there is a red traffic light in the dashcam image!"*

- This would be ideal: concepts with the right semantics **generalize** to new tasks (as required by, e.g., NeSy verification [9]) and support **interpretability**.

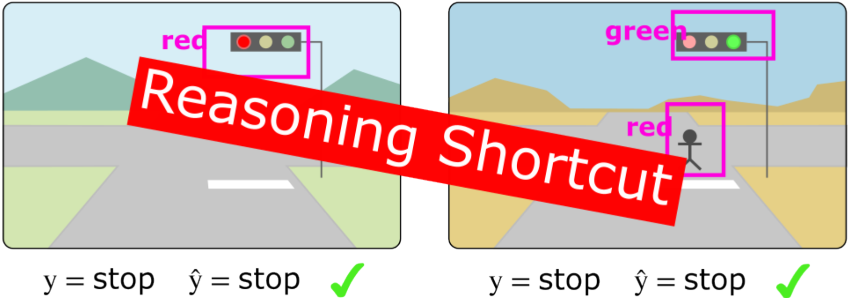
$$K_1 = (\text{pedestrian} \vee \text{red} \Rightarrow \text{stop})$$



$y = \text{stop}$ $\hat{y} = \text{stop}$ ✓ $y = \text{stop}$ $\hat{y} = \text{stop}$ ✓

- Task: predict `stop` vs. `go` using concepts "`pedestrian`", "`red`", and "`green`".

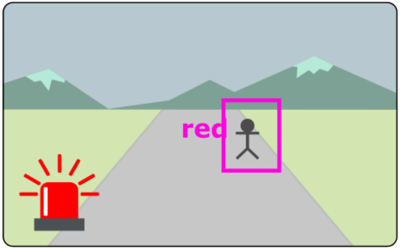
$$K_1 = (\text{pedestrian} \vee \text{red} \Rightarrow \text{stop})$$



- Task: predict `stop` vs. `go` using concepts "`pedestrian`", "`red`", and "`green`".

Perfect accuracy by predicting pedestrians as red lights!!!

$$K_2 = (\text{emergency} \wedge \neg\text{pedestrian} \Rightarrow \text{go}) \wedge K_1$$



$\hat{y} = \text{go}$ ✗

- Task: ... but now if there is an `emergency`, we can ignore "`red`" lights

We answer to the following questions:

We answer to the following questions:

1. **How can we characterize RSs**, and how many of them are present?

We answer to the following questions:

1. **How can we characterize RSs**, and how many of them are present?
2. What are the `root` **causes**?

We answer to the following questions:

1. **How can we characterize RSs**, and how many of them are present?
2. What are the `root` **causes**?
3. What are *natural* **mitigation strategies**?

We answer to the following questions:

1. **How can we characterize RSs**, and how many of them are present?
2. What are the `root` **causes**?
3. What are *natural* **mitigation strategies**?
4. Do RSs appear in **real-world tasks**?

Formalization (but not covered today :))

- Data generation: $\mathbf{g} \in \mathcal{G} \subset \mathbb{N}^k$ and $\mathbf{s} \in \mathbb{R}^q$. The joint distribution is factorized $p(\mathbf{G})p(\mathbf{S})$ and there exist:

$$p^*(\mathbf{X} \mid \mathbf{G}, \mathbf{S}) \quad \text{and} \quad p^*(\mathbf{Y} \mid \mathbf{G}; \mathsf{K})$$

where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ and $\mathbf{y} \in \mathcal{Y} \subset \mathbb{N}^\ell$. Typically, $\ell < k$.

- Learning in DPL, with $\mathcal{C} = \mathcal{G}$:

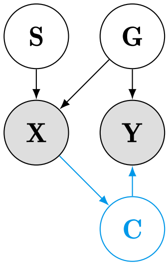
$$p_\theta(\mathbf{Y} \mid \mathbf{X}; \mathsf{K}) = \sum_{\mathbf{c} \in \mathcal{C}} p^*(\mathbf{Y} \mid \mathbf{c}; \mathsf{K}) p_\theta(\mathbf{c} \mid \mathbf{X})$$

$$\text{trained with } \max_\theta \prod_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} p_\theta(\mathbf{y} \mid \mathbf{x}; \mathsf{K})$$

- Technical Assumptions:

A1 There exists invertible (and differentiable over $\mathbf{s}$) $f : (\mathbf{g}, \mathbf{s}) \mapsto \mathbf{x}$ underlying $p^*(\mathbf{X} \mid \mathbf{G}, \mathbf{S})$: $p^*(\mathbf{x} \mid \mathbf{g}, \mathbf{s}) = \delta(\mathbf{x} - f(\mathbf{g}, \mathbf{s}))$

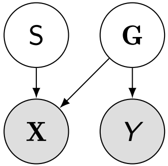
A2 The knowledge K is deterministic: there exists $\beta_\mathsf{K} : \mathbf{g} \mapsto \mathbf{y}$ such that $p^*(\mathbf{y} \mid \mathbf{g}; \mathsf{K}) = \mathbb{1}\{\mathbf{y} = \beta_\mathsf{K}(\mathbf{g})\}$



The data generation process.

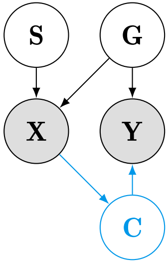
1 Intended Semantics vs Reasoning Shortcuts

- We construct the data generation process from Causal Representation Learning where:
  - $\mathbf{G}$ are (*binary or discrete*) ground-truth concepts
  - $\mathbf{S}$ are (*real-valued*) stylistic factors



The data generation process.

- We construct the data generation process from Causal Representation Learning where:
  - $\mathbf{G}$ are (*binary or discrete*) ground-truth concepts
  - $\mathbf{S}$ are (*real-valued*) stylistic factors

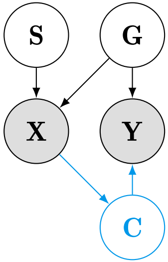


The NeSy model extracts $\mathbf{C}$ and computes $\mathbf{Y}$.

① Intended Semantics vs Reasoning Shortcuts

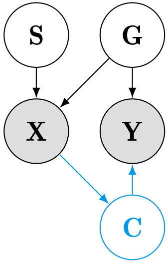
- We construct the data generation process from Causal Representation Learning where:
  - $\mathbf{G}$ are (*binary or discrete*) ground-truth concepts
  - $\mathbf{S}$ are (*real-valued*) stylistic factors

- We consider two key properties:



The NeSy model extracts $\mathbf{C}$ and computes $\mathbf{Y}$.

- We construct the data generation process from Causal Representation Learning where:
  - $\mathbf{G}$ are (*binary or discrete*) ground-truth concepts
  - $\mathbf{S}$ are (*real-valued*) stylistic factors
- We consider two key properties:
  1. **Optimality** **when** $\mathbf{Y}$ *is correct* $\wedge$ K *is satisfied*

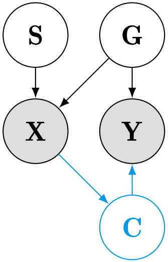


The NeSy model extracts $\mathbf{C}$ and computes $\mathbf{Y}$.

① Intended Semantics vs Reasoning Shortcuts

- We construct the data generation process from Causal Representation Learning where:
  - $\mathbf{G}$ are (*binary or discrete*) ground-truth concepts
  - $\mathbf{S}$ are (*real-valued*) stylistic factors

- We consider two key properties:
  1. **Optimality when** $\mathbf{Y}$ *is correct* $\wedge$ K *is satisfied*
  2. **Intended Semantics when** $(\forall \mathbf{x},\ \mathbf{c} = \mathbf{g})$



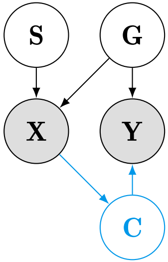
The NeSy model extracts $\mathbf{C}$ and computes $\mathbf{Y}$.

① Intended Semantics vs Reasoning Shortcuts

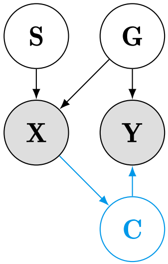
- We construct the data generation process from Causal Representation Learning where:
  - $\mathbf{G}$ are (*binary or discrete*) ground-truth concepts
  - $\mathbf{S}$ are (*real-valued*) stylistic factors

- We consider two key properties:
  1. **Optimality** **when** $\mathbf{Y}$ *is correct* $\wedge$ K *is satisfied*
  2. **Intended Semantics** **when** $(\forall \mathbf{x},\ \mathbf{c} = \mathbf{g})$

Def A **Reasoning Shortcut** (RS) occurs whenever the model achieves **optimality** but learns unintended concepts.

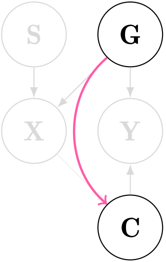


The NeSy model extracts $\mathbf{C}$ and computes $\mathbf{Y}$.



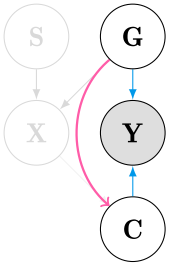
Generative process +
the NeSy predictor.

- Every NeSy predictor entails a map $\alpha : \mathbf{g} \mapsto \mathbf{c}$ from ground-truth to learned concepts. Ideally, it'd be the **identity**!



Map α entailed by a NeSy predictor.

- Every NeSy predictor entails a map $\boldsymbol{\alpha} : \mathbf{g} \mapsto \mathbf{c}$ from ground-truth to learned concepts. Ideally, it'd be the **identity**!
- We consider those α's that are optimal for the reasoning $\beta_{\mathsf{K}} : \mathbf{g} \mapsto \mathbf{y}$, underlying the prior knowledge.



Map α and knowledge β_{K}.

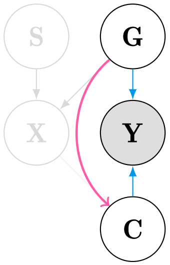
- Every NeSy predictor entails a map $\boldsymbol{\alpha} : \mathbf{g} \mapsto \mathbf{c}$ from ground-truth to learned concepts. Ideally, it'd be the **identity**!
- We consider those α's that are optimal for the reasoning $\beta_{\mathsf{K}} : \mathbf{g} \mapsto \mathbf{y}$, underlying the prior knowledge.

Theorem (informal): # of Reasoning Shortcuts

Under assumptions™, the # of these α's for NeSy predictors[a] is:

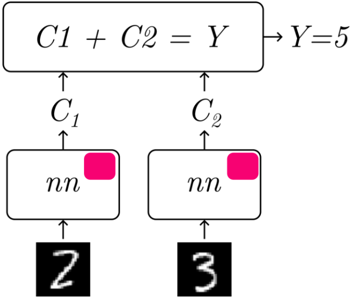
$$\sum_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbb{1}\{ \bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} (\boldsymbol{\beta}_{\mathsf{K}} \circ \boldsymbol{\alpha})(\mathbf{g}) = \boldsymbol{\beta}_{\mathsf{K}}(\mathbf{g}) \} \geq 1$$

[a] We prove they are shared between DeepProbLog [6], Semantic Loss [7], and Logic Tensor Networks [8].

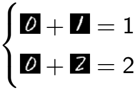


Map α and knowledge β_{K}.

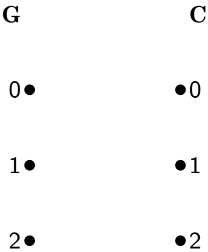
Example: `MNIST-Addition` with few sums.



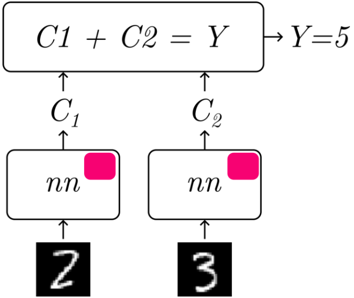
Inference in DeepProbLog with the same neural network ■



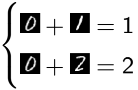
Problem. How many optimal solutions?



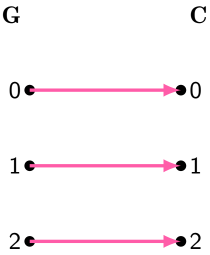
Example: `MNIST-Addition` with few sums.



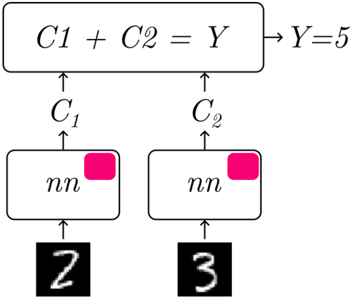
Inference in DeepProbLog with the same neural network ■



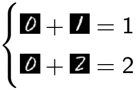
Solution 1. Intended solution



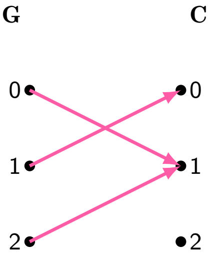
Example: `MNIST-Addition` with few sums.



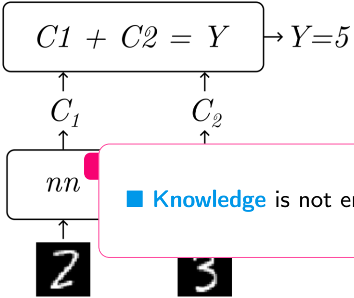
Inference in DeepProbLog with the same neural network

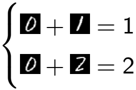


Solution 2. Reasoning Shortcut

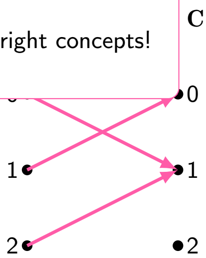


Example: `MNIST-Addition` with few sums.





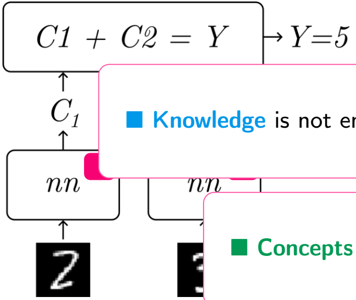
Solution 2. Reasoning Shortcut

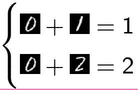


■ **Knowledge** is not enough explicit to recover the right concepts!

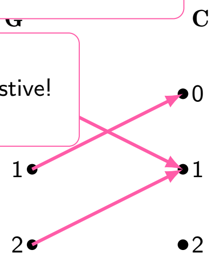
Inference in DeepProbLog with the same neural network ■

Example: `MNIST-Addition` with few sums.





■ **Knowledge** is not enough explicit to recover the right concepts!



■ **Concepts combinations** are not exhaustive!

Inference in DeepProbLog with the same neural network ■

② What are the Root Causes?

Recall the count is:

$$\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \mathbb{1}\{ \bigwedge_{\mathbf{g}\in\mathsf{supp}(\mathbf{G})} (\boldsymbol{\beta}_{\mathsf{K}} \circ \boldsymbol{\alpha})(\mathbf{g}) = \boldsymbol{\beta}_{\mathsf{K}}(\mathbf{g})\} \geq 1$$

Recall the count is:

$$\sum_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbb{1}\{ \bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} (\boldsymbol{\beta}_{\mathsf{K}} \circ \boldsymbol{\alpha})(\mathbf{g}) = \boldsymbol{\beta}_{\mathsf{K}}(\mathbf{g}) \} \geq 1$$

Causes of RSs can be read off of it:

② What are the Root Causes?

Recall the count is:

$$\sum_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbb{1}\{ \bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} (\beta_{\mathsf{K}} \circ \boldsymbol{\alpha})(\mathbf{g}) = \beta_{\mathsf{K}}(\mathbf{g}) \} \geq 1$$

Causes of RSs can be read off of it:

$\mathcal{K}$ Structure of **knowledge** K, via β_{K}

② What are the Root Causes?

Recall the count is:

$$\sum_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbb{1}\{ \bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} (\boldsymbol{\beta}_{\mathsf{K}} \circ \boldsymbol{\alpha})(\mathbf{g}) = \boldsymbol{\beta}_{\mathsf{K}}(\mathbf{g}) \} \geq 1$$

Causes of RSs can be read off of it:

$\mathcal{K}$ Structure of **knowledge** K, via $\boldsymbol{\beta}_{\mathsf{K}}$

$\mathcal{D}$ Structure of **data set**, via $\mathsf{supp}(\mathbf{G})$

Recall the count is:

$$\sum_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbb{1}\{ \bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} \underbrace{(\beta_{\mathsf{K}} \circ \boldsymbol{\alpha})(\mathbf{g}) = \beta_{\mathsf{K}}(\mathbf{g})}_{\textit{objective}} \} \geq 1$$

Causes of RSs can be read off of it:

$\mathcal{K}$ Structure of **knowledge** K, via β_{K}

$\mathcal{D}$ Structure of **data set**, via $\mathsf{supp}(\mathbf{G})$

$\mathcal{L}$ **Loss**, via optimality of *objective*

② What are the Root Causes?

Recall the count is:

$$\sum_{\alpha \in \mathcal{A}} \mathbb{1}\{ \bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} \underbrace{(\beta_{\mathsf{K}} \circ \alpha)(\mathbf{g}) = \beta_{\mathsf{K}}(\mathbf{g})}_{\textit{objective}} \} \geq 1$$

Causes of RSs can be read off of it:

- $\mathcal{K}$ Structure of **knowledge** K, via β_{K}
- $\mathcal{L}$ **Loss**, via optimality of *objective*
- $\mathcal{D}$ Structure of **data set**, via $\mathsf{supp}(\mathbf{G})$
- $\mathcal{A}$ **Architecture bias** of the model, via α

③ Natural Mitigation Strategies (are not enough)

	TARGET	REQ.
Multi-task	$\mathcal{K}$	tasks

Knowledge becomes more explicit

- Multi-task learning consists on solving more tasks in parallel

③ Natural Mitigation Strategies (are not enough)

	TARGET	REQ.
Multi-task	$\mathcal{K}$	tasks

Knowledge becomes more explicit

	TARGET	REQ.
Concept-sup	$\mathcal{D}+\mathcal{L}$	concepts

Optimizes the ideal objective

- Concept supervision involves regressing on the ground-truth concepts

③ Natural Mitigation Strategies (are not enough)

	TARGET	REQ.
Multi-task	$\mathcal{K}$	tasks

Knowledge becomes more explicit

	TARGET	REQ.
Concept-sup	$\mathcal{D}+\mathcal{L}$	concepts

Optimizes the ideal objective

	TARGET	REQ.
Reconstruction	$\mathcal{L}$	(dec.)

Avoids collapsing the concepts

- Reconstruction penalty for recovering the original input from the learned concepts

③ Natural Mitigation Strategies (are not enough)

	TARGET	REQ.
Multi-task	$\mathcal{K}$	tasks

Knowledge becomes more explicit

	TARGET	REQ.
Concept-sup	$\mathcal{D}+\mathcal{L}$	concepts

Optimizes the ideal objective

	TARGET	REQ.
Reconstruction	$\mathcal{L}$	(dec.)

Avoids collapsing the concepts

	TARGET	REQ.
Disentanglement	$\mathcal{A}$	structure

Independent concepts

- Disentanglement = independent variations of the concepts

Natural Mitigation Strategies (are not enough)

Table 1: **Impact of different mitigation strategies on the number of deterministic optima**: R is reconstruction, C supervision on $\mathbf{C}$, MTL multi-task learning, and DIS disentanglement. All strategies reduce the number of α's in Eq. (6), sometimes substantially, but require different amounts of effort to be put in place. Actual counts for our data sets are reported in Appendix C.2.

Mitigation	Requires	Constraint on α	Assumptions	Result
None	–	$\bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} ((\beta_{\mathsf{K}} \circ \alpha)(\mathbf{g}) = \beta_{\mathsf{K}}(\mathbf{g}))$	**A1**, **A2**	Theorem 2
MTL	Tasks	$\bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} \bigwedge_{t \in [T]} ((\beta_{\mathsf{K}^{(t)}} \circ \alpha)(\mathbf{g}) = \beta_{\mathsf{K}^{(t)}}(\mathbf{g}))$	**A1**, **A2**	Proposition 4
C	Sup. on $\mathbf{C}$	$\bigwedge_{\mathbf{g} \in \mathcal{S} \subseteq \mathsf{supp}(\mathbf{G})} \bigwedge_{i \in I} (\alpha_i(\mathbf{g}) = g_i)$	**A1**	Proposition 5
R	–	$\bigwedge_{\mathbf{g}, \mathbf{g}' \in \mathsf{supp}(\mathbf{G}) : \mathbf{g} \neq \mathbf{g}'} (\alpha(\mathbf{g}) \neq \alpha(\mathbf{g}'))$	**A1**, **A3**	Proposition 6

Natural Mitigation Strategies (are not enough)

Table 1: **Impact of different mitigation strategies on the number of deterministic optima**: R is reconstruction, C supervision on $\mathbf{C}$, MTL multi-task learning, and DIS disentanglement. All strategies reduce the number of α's in Eq. (6), sometimes substantially, but require different amounts of effort to be put in place. Actual counts for our data sets are reported in Appendix C.2.

MITIGATION	REQUIRES	CONSTRAINT ON α	ASSUMPTIONS	RESULT
None	–	$\bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} ((\beta_{\mathsf{K}} \circ \alpha)(\mathbf{g}) = \beta_{\mathsf{K}}(\mathbf{g}))$	**A1, A2**	Theorem 2
MTL	Tasks	$\bigwedge_{\mathbf{g} \in \mathsf{supp}(\mathbf{G})} \bigwedge_{t \in [T]} ((\beta_{\mathsf{K}^{(t)}} \circ \alpha)(\mathbf{g}) = \beta_{\mathsf{K}^{(t)}}(\mathbf{g}))$	**A1, A2**	Proposition 4
C	Sup. on $\mathbf{C}$	$\bigwedge_{\mathbf{g} \in \mathcal{S} \subseteq \mathsf{supp}(\mathbf{G})} \bigwedge_{i \in I} (\alpha_i(\mathbf{g}) = g_i)$	**A1**	Proposition 5
R	–	$\bigwedge_{\mathbf{g}, \mathbf{g}' \in \mathsf{supp}(\mathbf{G}) : \mathbf{g} \neq \mathbf{g}'} (\alpha(\mathbf{g}) \neq \alpha(\mathbf{g}'))$	**A1, A3**	Proposition 6

- Plenty of **experiments**: **no existing mitigation strategy is sufficient in all cases**!

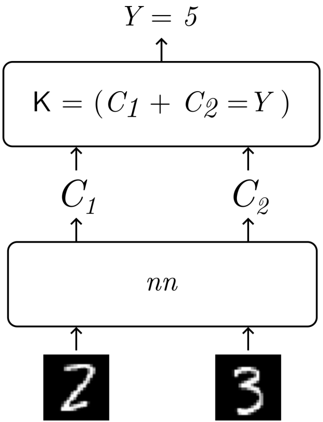
Natural Mitigation Strategies (are not enough)

Table 1: **Impact of different mitigation strategies on the number of deterministic optima**: R is reconstruction, C supervision on $\mathbf{C}$, MTL multi-task learning, and DIS disentanglement. All strategies reduce the number of α's in Eq. (6), sometimes substantially, but require different amounts of effort to be put in place. Actual counts for our data sets are reported in Appendix C.2.

MITIGATION	REQUIRES	CONSTRAINT ON α	ASSUMPTIONS	RESULT
None	–	$\bigwedge_{\mathbf{g}\in\mathsf{supp}(\mathbf{G})} ((\beta_\mathsf{K} \circ \alpha)(\mathbf{g}) = \beta_\mathsf{K}(\mathbf{g}))$	**A1, A2**	Theorem 2
MTL	Tasks	$\bigwedge_{\mathbf{g}\in\mathsf{supp}(\mathbf{G})} \bigwedge_{t\in[T]} ((\beta_{\mathsf{K}^{(t)}} \circ \alpha)(\mathbf{g}) = \beta_{\mathsf{K}^{(t)}}(\mathbf{g}))$	**A1, A2**	Proposition 4
C	Sup. on $\mathbf{C}$	$\bigwedge_{\mathbf{g}\in\mathcal{S}\subseteq\mathsf{supp}(\mathbf{G})} \bigwedge_{i\in I} (\alpha_i(\mathbf{g}) = g_i)$	**A1**	Proposition 5
R	–	$\bigwedge_{\mathbf{g},\mathbf{g}'\in\mathsf{supp}(\mathbf{G}):\mathbf{g}\neq\mathbf{g}'} (\alpha(\mathbf{g}) \neq \alpha(\mathbf{g}'))$	**A1, A3**	Proposition 6

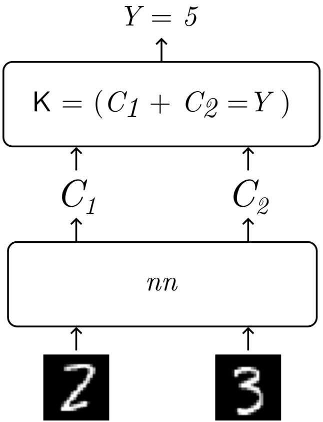
- Plenty of **experiments**: **no existing mitigation strategy is sufficient in all cases**!
- **Bonus:** We prove that optimal maps α's are the extremes of the simplex of optimal solutions in Probabilistic Logic. This can be leveraged to be agnostic about which RS to pick!

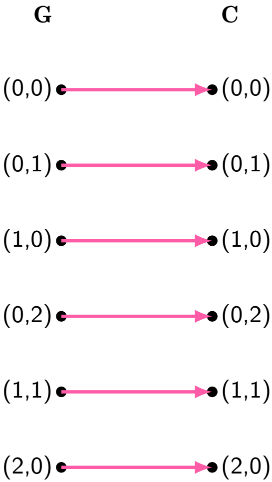
Example: `MNIST-Addition` with full dataset, without **disentanglement**



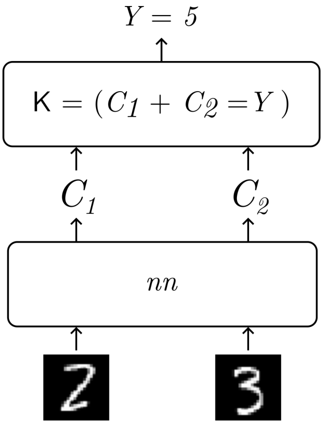
G	C
(0,0)	(0,0)
(0,1)	(0,1)
(1,0)	(1,0)
(0,2)	(0,2)
(1,1)	(1,1)
(2,0)	(2,0)

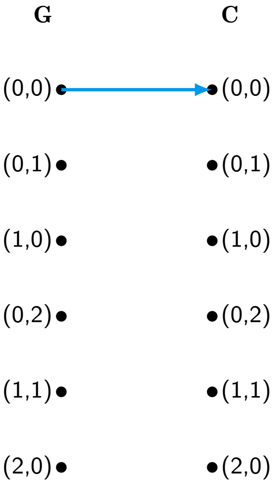
Example: `MNIST-Addition` with full dataset, without **disentanglement**



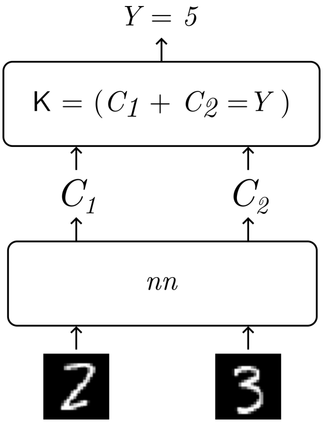


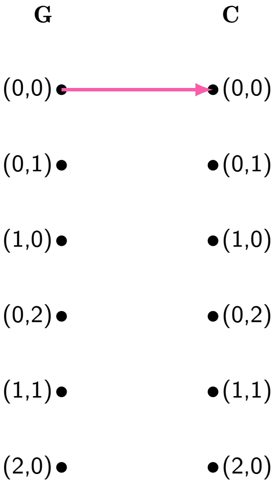
Example: `MNIST-Addition` with full dataset, without **disentanglement**



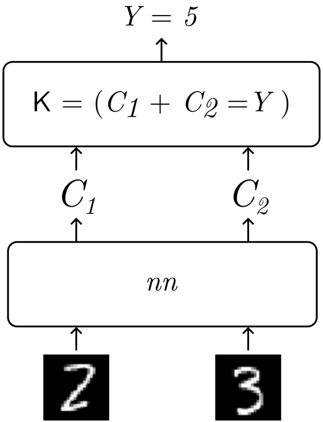


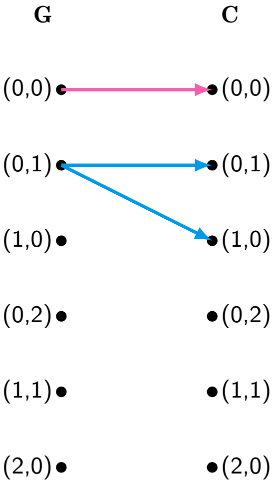
Example: `MNIST-Addition` with full dataset, without **disentanglement**



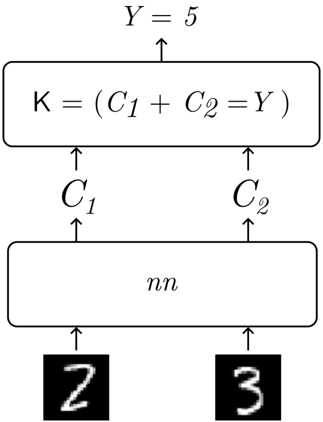


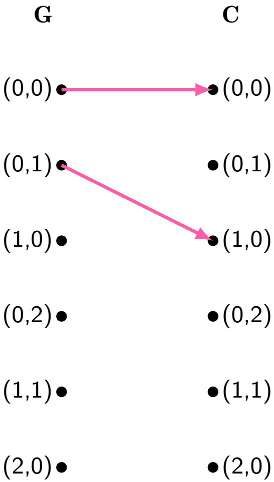
Example: `MNIST-Addition` with full dataset, without **disentanglement**



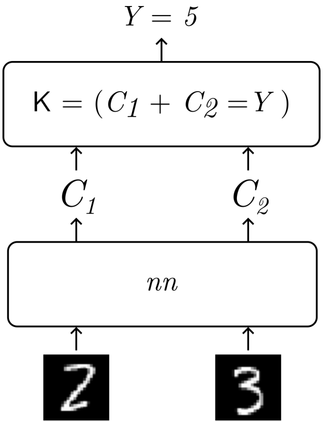


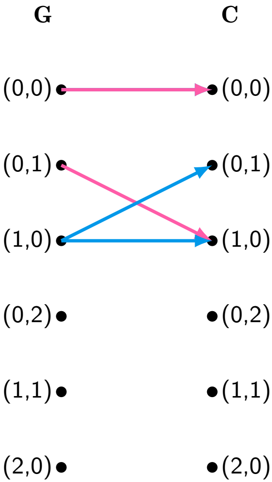
Example: `MNIST-Addition` with full dataset, without **disentanglement**



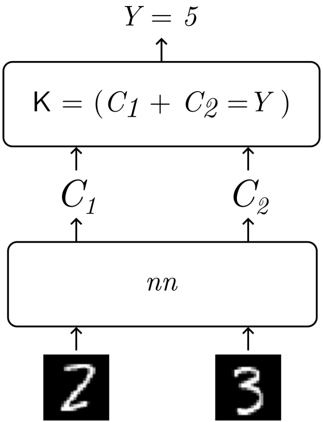


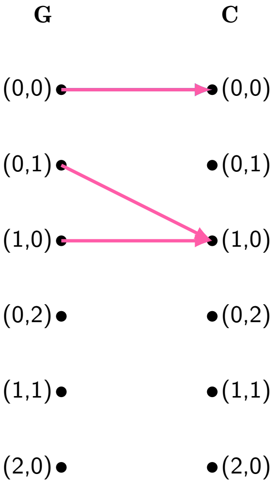
Example: `MNIST-Addition` with full dataset, without **disentanglement**



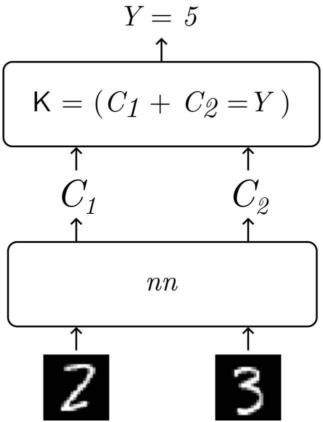


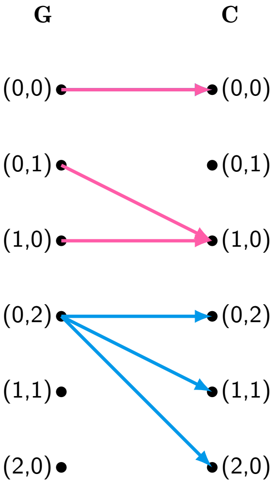
Example: `MNIST-Addition` with full dataset, without **disentanglement**



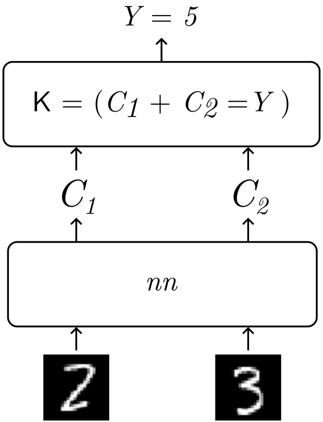


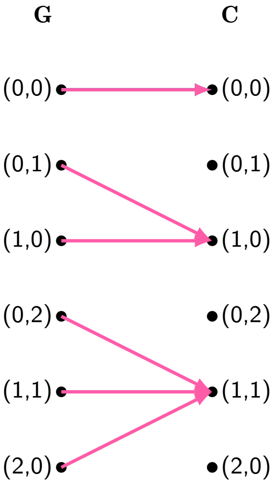
Example: `MNIST-Addition` with full dataset, without **disentanglement**



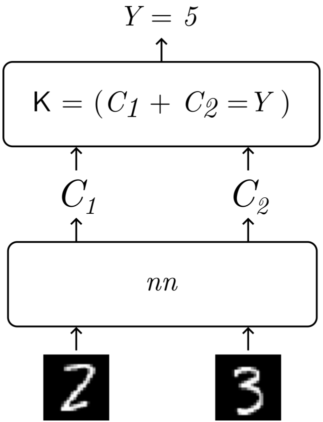


Example: `MNIST-Addition` with full dataset, without **disentanglement**



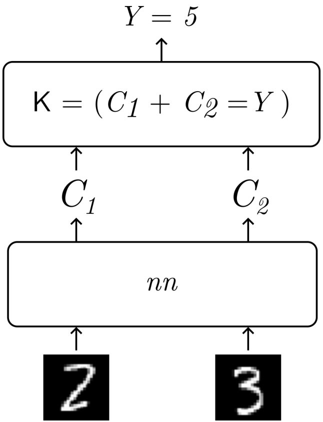


Example: `MNIST-Addition` with full dataset, without **disentanglement**



In this setting, the number of Reasoning Shortcuts scales like:

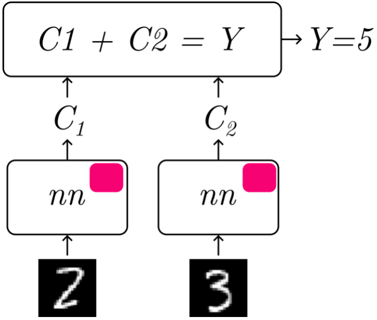
Example: `MNIST-Addition` with full dataset, without **disentanglement**



In this setting, the number of Reasoning Shortcuts scales like:

$$\sim 10^{78}$$

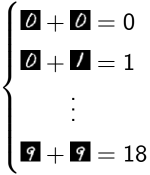
Example: `MNIST-Addition` with full dataset, without **disentanglement**



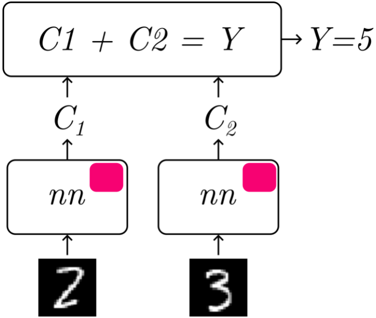
In this setting, the number of Reasoning Shortcuts scales like:

$$\sim 10^{78}$$

If we **disentangle** the concepts, Reasoning Shortcuts completely vanish!



Example: `MNIST-Addition` with full dataset, without **disentanglement**



In this setting, the number of Reasoning Shortcuts scales like:

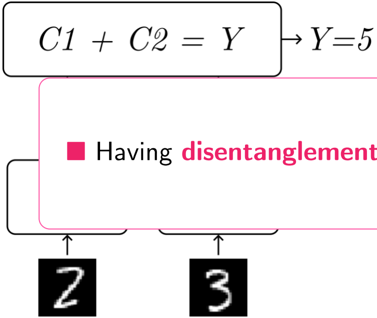
$$\sim 10^{78}$$

If we **disentangle** the concepts, Reasoning Shortcuts completely vanish!

Frequency in % of RSs over 30 optimal runs.

	`XOR`			`MNIST-Addition`		
	DPL	SL	LTN	DPL	SL	LTN
–	100%	100%	100%	96.7%	82.9%	100%
DIS	0%	0%	0%	0%	0%	0%

Example: `MNIST-Addition` with full dataset, without **disentanglement**



In this setting, the number of Reasoning Shortcuts scales like:

$$\sim 10^{78}$$

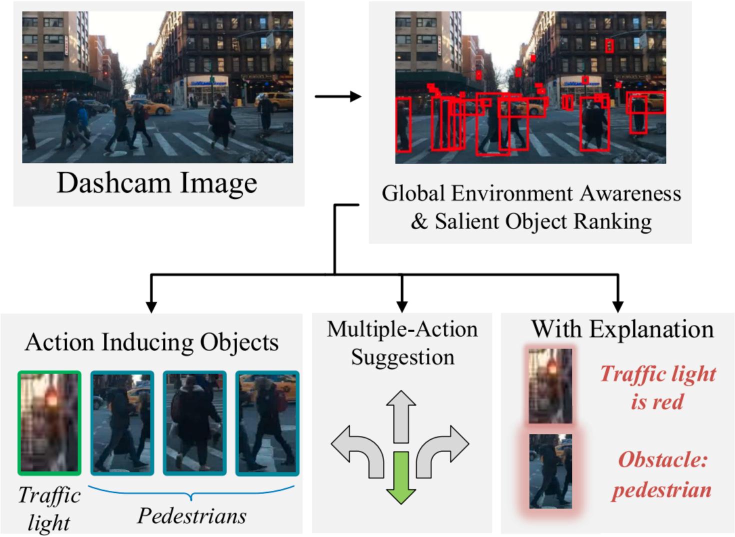
g

- Having **disentanglement** in practice is hard, the # of RSs can explode!

runs.

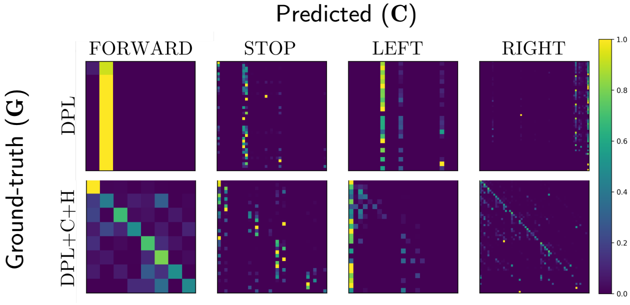
	XOR			MNIST-Addition		
	DPL	SL	LTN	DPL	SL	LTN
–	100%	100%	100%	96.7%	82.9%	100%
DIS	0%	0%	0%	0%	0%	0%

BDD-OIA: Real-world Autonomous Vehicle Data Set [10]



④ BDD-OIA: Real-world Autonomous Vehicle Data Set [10]

- Predict one or more actions:
  - `move_forward` / `stop`
  - `turn_left`
  - `turn_right`
- 20-ish concepts including:
  - `red_light` / `green_light`
  - `obstacle` / `road_clear`
  - ...



Top: No supervision on concepts

Bottom: Full supervision on concepts + entropy

Unintended Concepts do not Transfer in NeSy Continual Learning [11]

Unintended Concepts do not Transfer in NeSy Continual Learning [11]

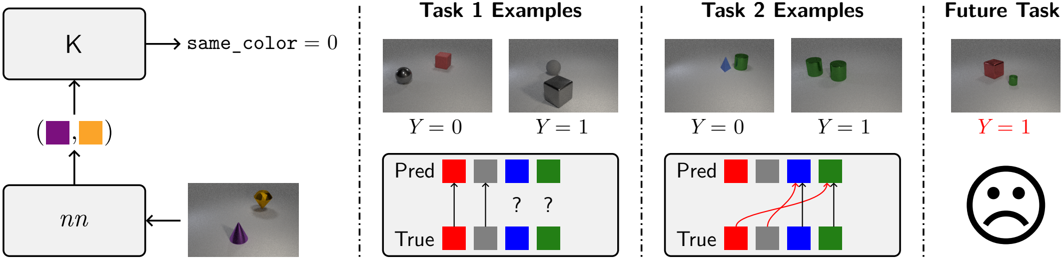
- Learn to solve a sequence of NeSy predictions over *different episodes/tasks*

Unintended Concepts do not Transfer in NeSy Continual Learning [11]

- Learn to solve a sequence of NeSy predictions over *different episodes/tasks*
- We show that here Reasoning Shortcuts are also likely to happen and standard CL strategies fail to preserve the intended concepts

Unintended Concepts do not Transfer in NeSy Continual Learning [11]

- Learn to solve a sequence of NeSy predictions over *different episodes/tasks*
- We show that here Reasoning Shortcuts are also likely to happen and standard CL strategies fail to preserve the intended concepts



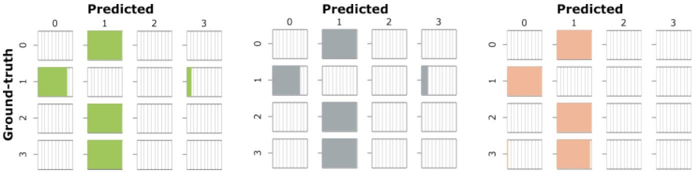
E. Marconato, G. Bontempo, E. Ficarra, S. Calderara, A. Passerini, and S. Teso; Neuro-Symbolic Continual Learning: Knowledge, Reasoning Shortcuts, and Concept Rehearsal, **ICML** (2023).

Take-home message:

- Interpretability of concepts can be framed within a Causal framework and allows to define properly some notions, like *interpretability*, **concept leakage**, and **completeness**.

- Reasoning shortcuts constitute a **severe problem** for **perception** (the maps α) + **reasoning** (the knowledge K), undermining **trustworthiness** and **interpretability**.
    - **Existing mitigation strategies are not effective** and more research is needed!

- Fruitful intersection between Concept Learning, NeSy AI with Causal Representation Learning.



DeepProbLog (left), **Semantic Loss** (center), and **Logic Tensor Networks** (right) pick similar Reasoning Shortcuts.

Future extension to learning end-to-end

- New works are appearing to learn both the concepts and the knowledge:
  - **DSL** (**D**eep **S**ymbolic **L**earning) [12]
  - **ROAP** (**R**egularize, **O**verparametrize, and **A**moritize for **P**rograms) [13]

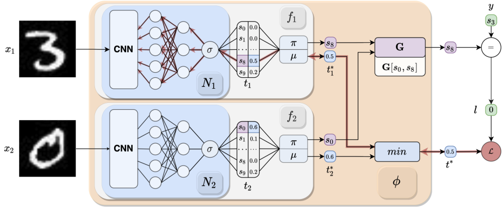


Figure 1: Architecture of Deep Symbolic Learning for the Sum task. Red arrows represent the backward signal during learning.

- This problem is even less constrained than before:
  - RSs affecting models with prior knowledge **transfer** to models learning the knowledge
  - More RSs can appear by mistaking the knowledge

Do LLMs synthesize Reasoning Shortcuts?

ChatGPT 3.5

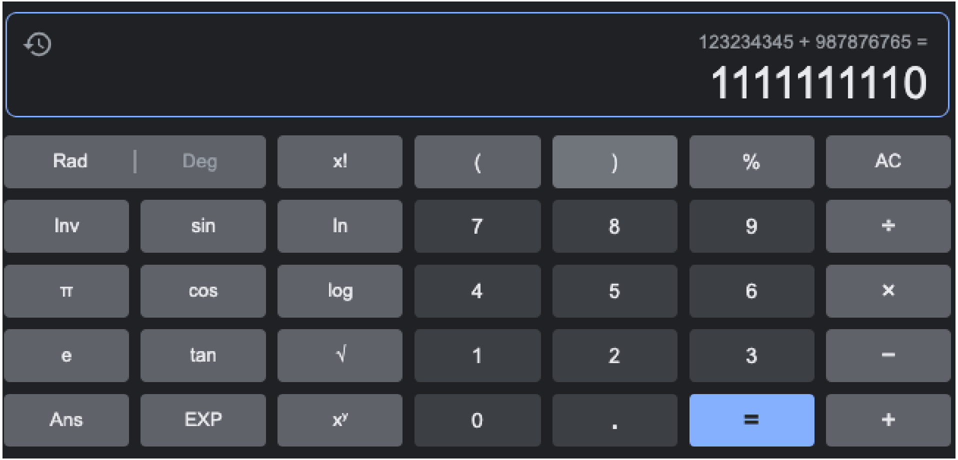
Continue until you reach the leftmost column. In this case, you'll end up with:

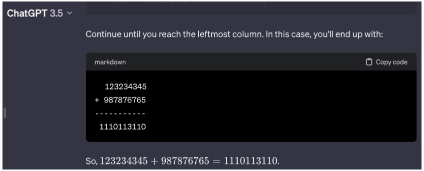
markdown Copy code

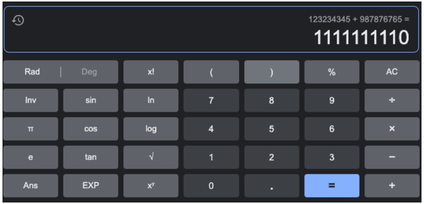
```
   123234345
 + 987876765
 -----------
  1110113110
```

So, $123234345 + 987876765 = 1110113110$.

Do LLMs synthesize Reasoning Shortcuts?







- LLMs (without plugins) fail in reasoning:
  - Shortcut behavior in NLI [14]
  - Failures in reasoning benchmarks [15]
  - Non-unique solutions in modular arithmetic [16]
  - Pitfalls of out-of-distribution generalization on Dyck grammars with 2-layer transformers [17]

Because of an interplay of **wrong concepts** and/or **wrong knowledge**?

Thank you for your attention!

paper

Contacts:

: emanuele.marconato@unitn.it
: ema-marconato
: ema_marconato

code

References

[1] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[2] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

[3] Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended? *arXiv preprint arXiv:2105.04289*, 2021.

[4] Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. In *International Conference on Machine Learning: Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*, volume 1, pages 1–13, 2021.

[5] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, pages 6056–6065. PMLR, 2019.

[6] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. DeepProbLog: Neural Probabilistic Logic Programming. *NeurIPS*, 2018.

[7] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. A semantic loss function for deep learning with symbolic knowledge. In *ICML*, 2018.

[8] Ivan Donadello, Luciano Serafini, and Artur D'Avila Garcez. Logic tensor networks for semantic image interpretation. In *IJCAI*, 2017.

[9] Xuan Xie, Kristian Kersting, and Daniel Neider. Neuro-symbolic verification of deep neural networks. *arXiv preprint arXiv:2203.00938*, 2022.

[10] Yiran Xu, Xiaoyin Yang, Lihang Gong, Hsuan-Chu Lin, Tz-Ying Wu, Yunsheng Li, and Nuno Vasconcelos. Explainable object-induced action decision for autonomous vehicles. In *CVPR*, June 2020.

[11] Emanuele Marconato, Gianpaolo Bontempo, Elisa Ficarra, Simone Calderara, Andrea Passerini, and Stefano Teso. Neuro symbolic continual learning: Knowledge, reasoning shortcuts and concept rehearsal. 2023.

[12] Alessandro Daniele, Tommaso Campari, Sagar Malhotra, and Luciano Serafini. Deep symbolic learning: Discovering symbols and rules from perceptions. *arXiv preprint arXiv:2208.11561*, 2022.

[13] Hao Tang and Kevin Ellis. From perception to programs: regularize, overparameterize, and amortize. In *International Conference on Machine Learning*, pages 33616–33631. PMLR, 2023.

[14] Xanh Ho, Johannes Mario Meissner, Saku Sugawara, and Akiko Aizawa. A survey on measuring and mitigating reasoning shortcuts in machine reading comprehension. *arXiv preprint arXiv:2209.01824*, 2022.

[15] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D Hwang, et al. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*, 2023.

[16] Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *arXiv preprint arXiv:2306.17844*, 2023.

[17] Kaiyue Wen, Yuchen Li, Bingbin Liu, and Andrej Risteski. (un) interpretability of transformers: a case study with dyck grammars. 2023.