CONCEPT-BASED INTERPRETABLE DEEP LEARNING

AAAI 2025 – Philadelphia, USA

WHO ARE WE?

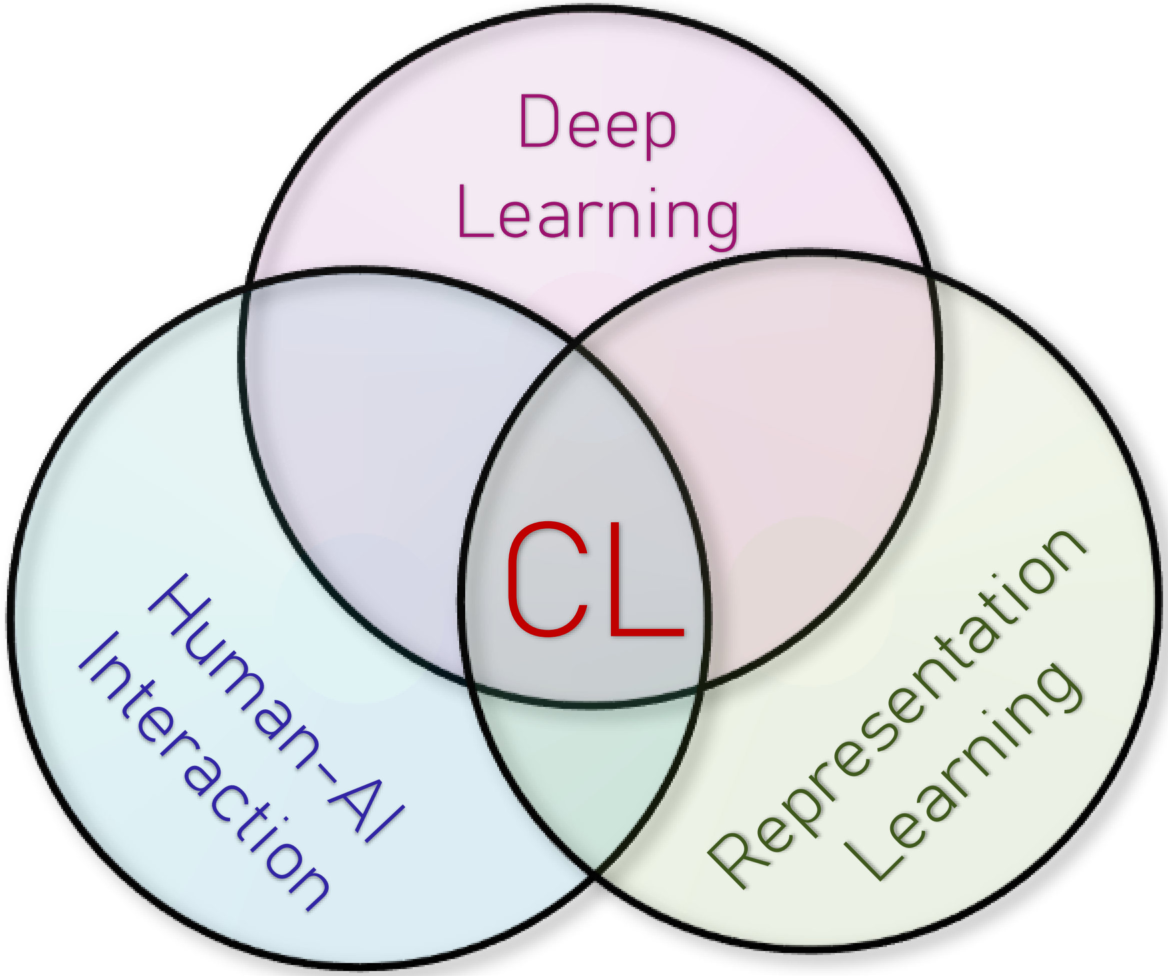
Mateo Espinosa Zarlenga
Final-year PhD Student
University of Cambridge
me466@cam.ac.uk

UNIVERSITY OF CAMBRIDGE

Pietro Barbiero
Swiss Postdoctoral Fellow
IBM Research (Switzerland)
pietro.barbiero@ibm.com

IBM Research

WHAT IS THIS TUTORIAL ABOUT?



We will look at how Concept Learning (CL) can be used to design interpretable Deep Neural Networks

TUTORIAL GOALS

Our **main goals** for this tutorial are threefold:

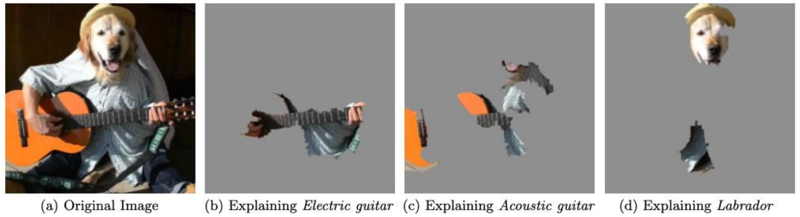
1. Provide a **non-exhaustive but well-rounded overview** of **concept learning (CL).**
2. Convince you that **concept representations** can be very useful for **designing powerful but interpretable** neural models.
3. Bring together a variety of **resources** (surveys, method papers, libraries, etc.) to **facilitate access to the current state of CL**.

WHAT IS THIS TUTORIAL **NOT** ABOUT?

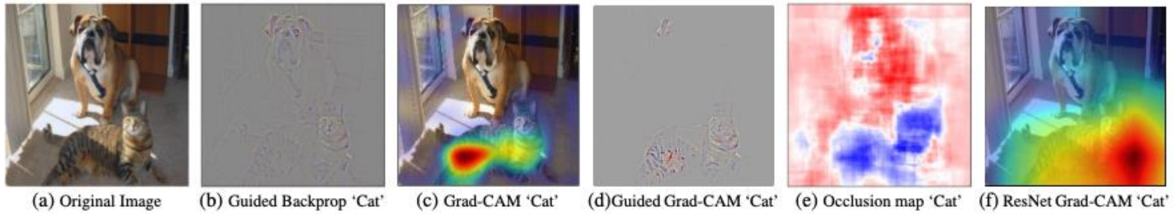
WHAT IS THIS TUTORIAL **NOT** ABOUT?

We will not have time to dive deep into:

1. "Traditional" explainable AI (XAI) methodologies



Example of *LIME* (taken from [1])



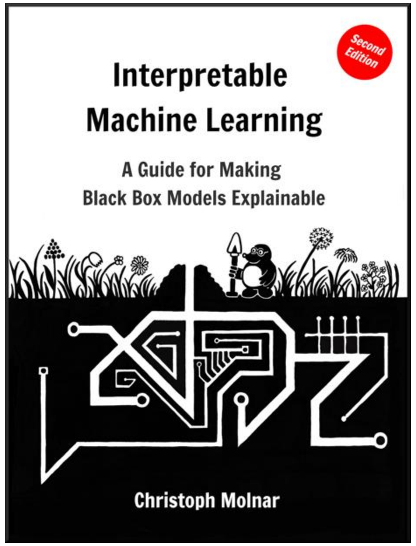
Example of *GradCAM* and other saliency methods (taken from [2])

[1] Ribeiro et al. "'Why should i trust you?' Explaining the predictions of any classifier." KDD (2016).
[2] Selvaraju, et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." ICCV (2017).

WHAT IS THIS TUTORIAL **NOT** ABOUT?

We will not have time to dive deep into:

1. "Traditional" explainable AI (XAI) methodologies



Interpretable
Machine Learning
A Guide for Making
Black Box Models Explainable
Second Edition
Christoph Molnar

Link to Book

Interpretable Machine Learning
Christoph Molnar

[1] Molnar, Christoph. Interpretable machine learning. Lulu. com, 2020.

WHAT IS THIS TUTORIAL **NOT** ABOUT?

We will not have time to dive deep into:

1. "Traditional" explainable AI (XAI) methodologies
2. Deep philosophical aspects of explaining models

The Mythos of Interpretability
Lipton et al. (2018) [1]

To Explain or to Predict?
Galit (2018) [2]

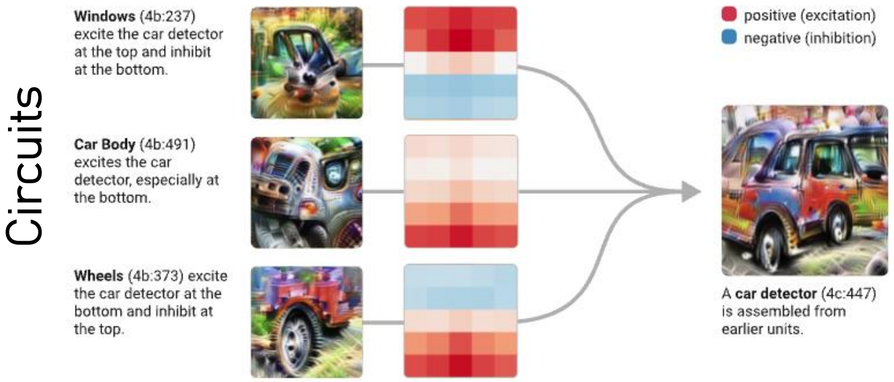
Explanation Theory
Bromberger (1992) [3]

[1] Lipton, Zachary C. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." Queue (2018).
[2] Galit Shmueli. "To Explain or to Predict?." Statist. Sci. 25 (3) 289 - 310, August 2010. https://doi.org/10.1214/10-STS330
[3] Bromberger, Sylvain. On what we know we don't know: Explanation, theory, linguistics, and how questions shape them. University of Chicago Press, 1992.

WHAT IS THIS TUTORIAL **NOT** ABOUT?

We will not have time to dive deep into:

1. "Traditional" explainable AI (XAI) methodologies
2. Deep philosophical aspects of explaining models
3. Connections with Mechanistic Interpretability



Taken from [1]

---

[1] Olah, Chris, et al. "Zoom in: An introduction to circuits." Distill 5.3 (2020): e00024-001.

WHAT IS THIS TUTORIAL **NOT** ABOUT?

We will **not** have time to dive deep into:

1. “Traditional” explainable AI (XAI) methodologies
2. Deep philosophical aspects of explaining models
3. Connections with Mechanistic Interpretability

Distill Circuits Thread

Anthropic Circuits Thread

[1] Cammarata, Nick, et al. "Thread: circuits." Distill 5.3 (2020): e24.
[2] Anthropic “Transformer Circuits Thread” found at https://transformer-circuits.pub/

TUTORIAL OUTLINE

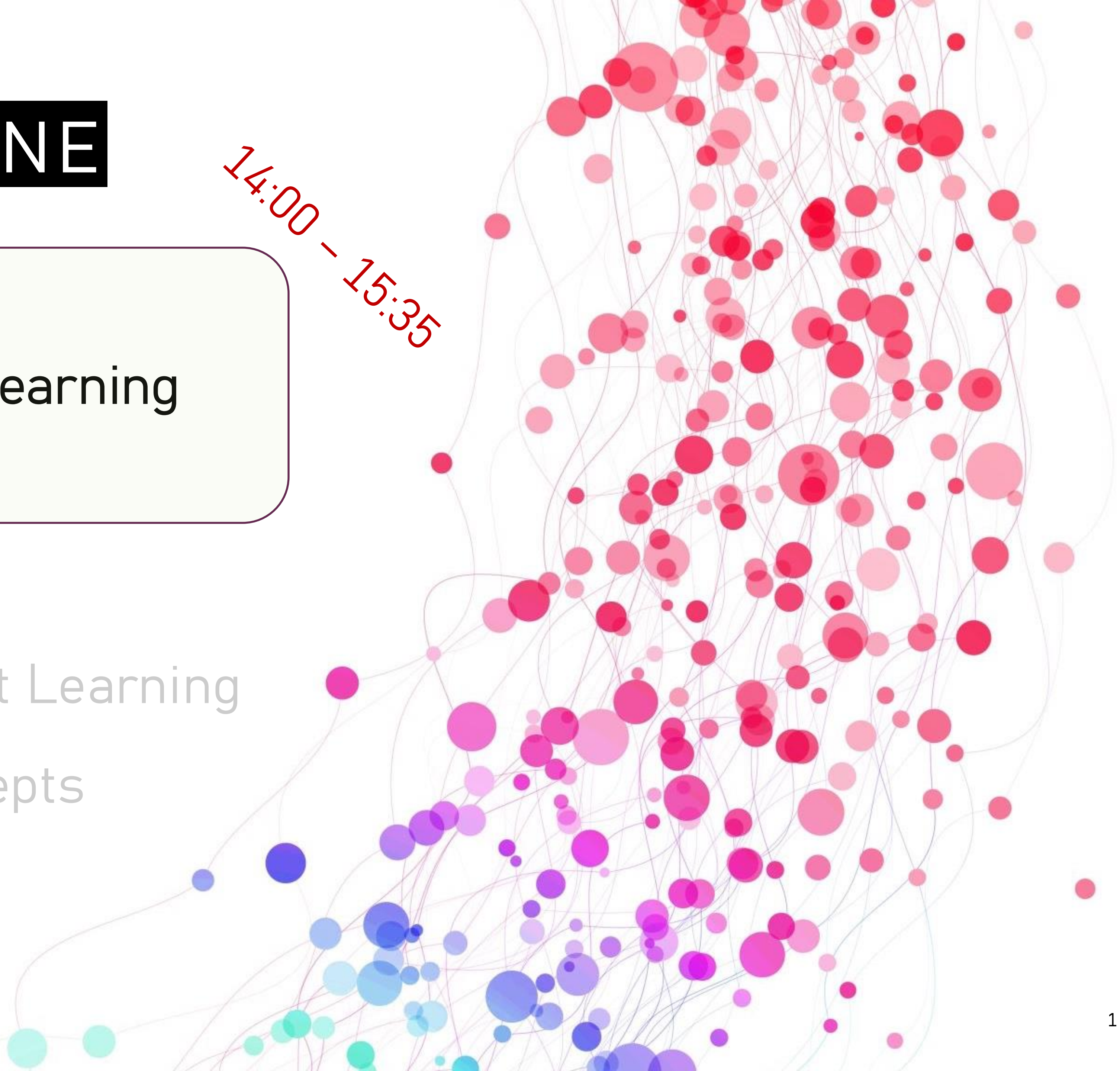
1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break
5. Unsupervised Concept Learning
6. Reasoning With Concepts
7. Future Directions
8. Q&A

TUTORIAL OUTLINE

14:00 – 15:35

Mateo

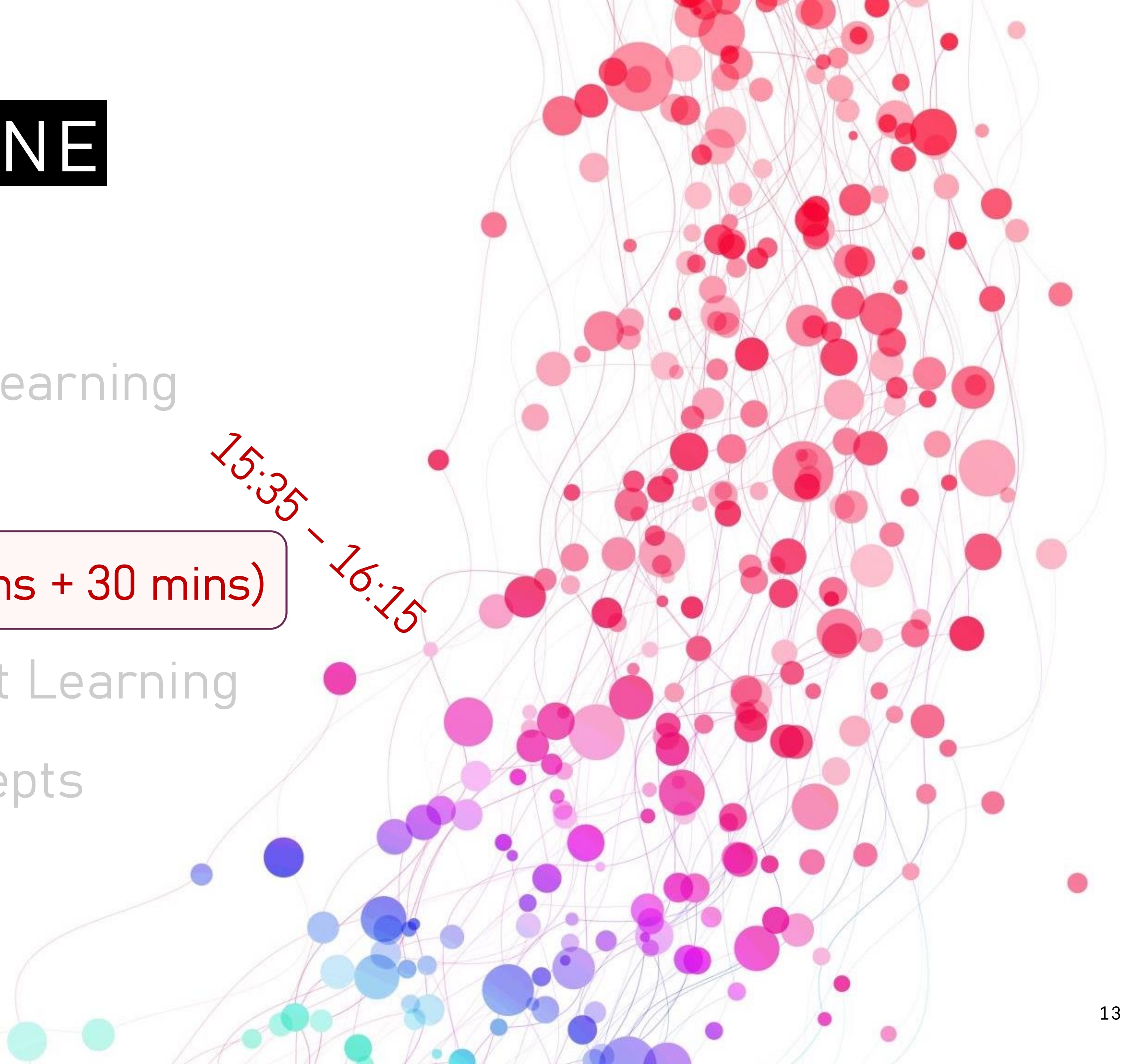
1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break
5. Unsupervised Concept Learning
6. Reasoning With Concepts
7. Future Directions
8. Q&A



TUTORIAL OUTLINE

1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. **Q&A + Break** (10 mins + 30 mins)
5. Unsupervised Concept Learning
6. Reasoning With Concepts
7. Future Directions
8. Q&A

15:35 – 16:15



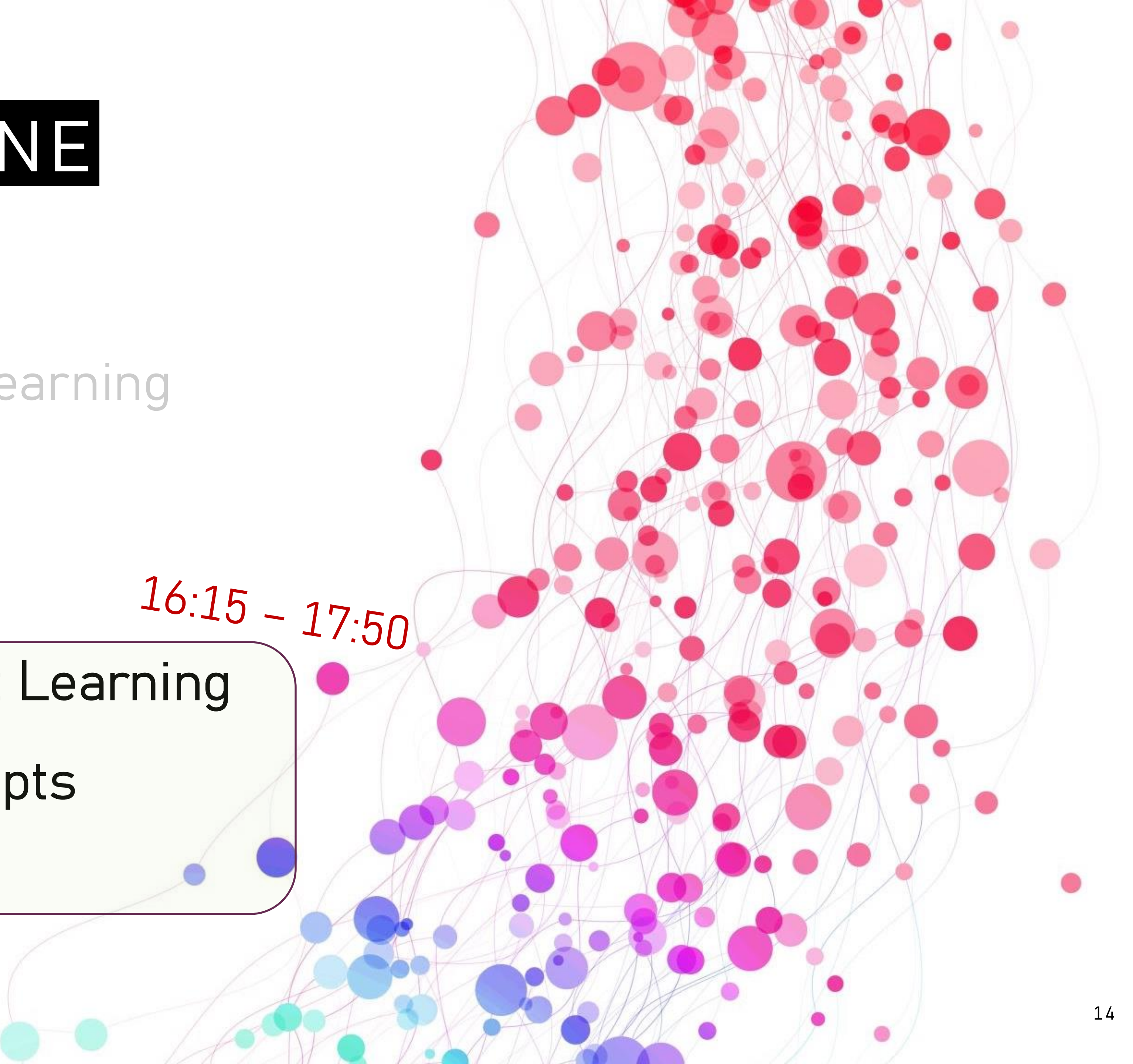
TUTORIAL OUTLINE

1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break

16:15 – 17:50

Pietro

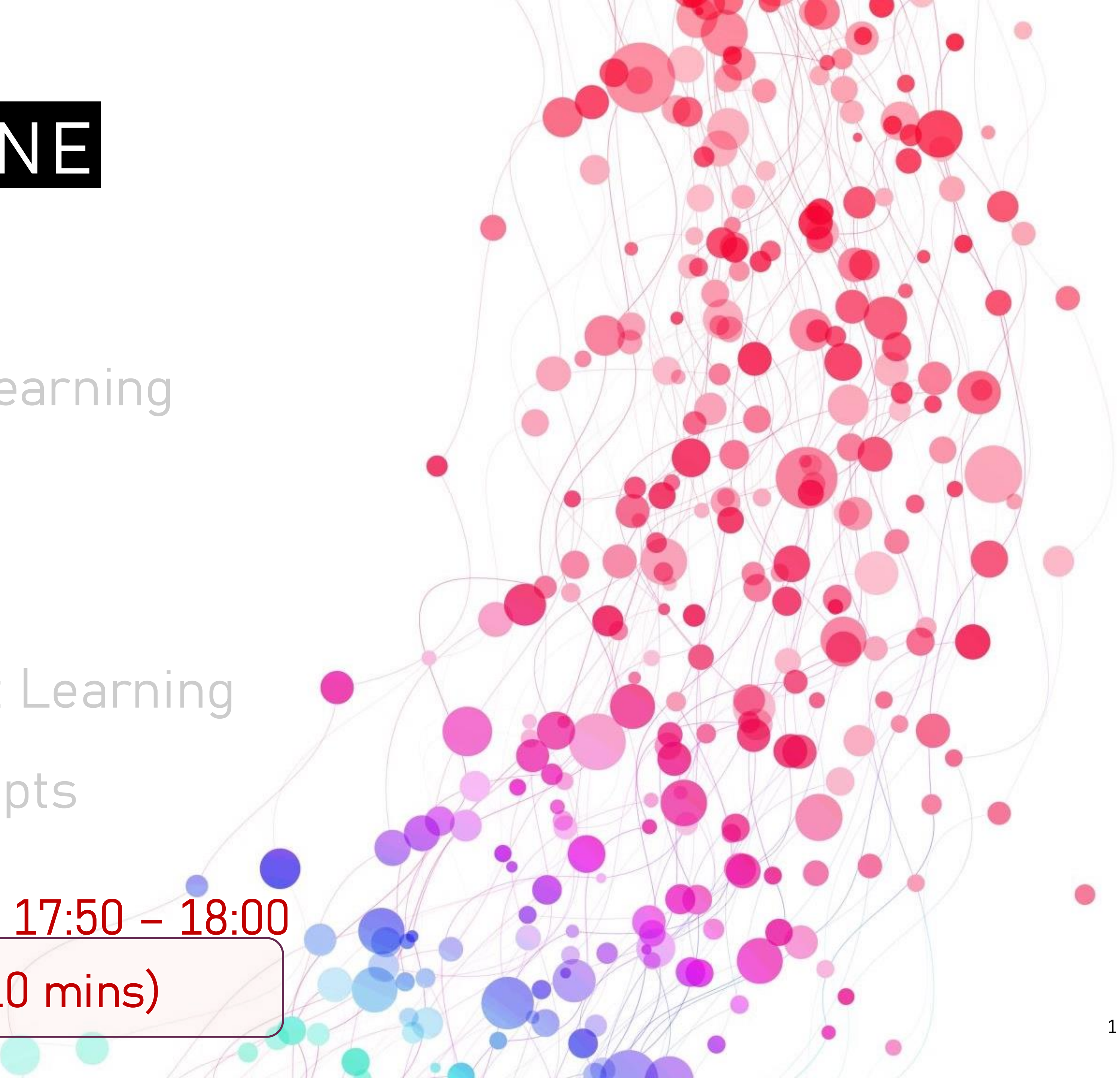
5. Unsupervised Concept Learning
6. Reasoning With Concepts
7. Future Directions
8. Q&A



TUTORIAL OUTLINE

1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break
5. Unsupervised Concept Learning
6. Reasoning With Concepts
7. Future Directions
8. Q&A (10 mins)

17:50 – 18:00



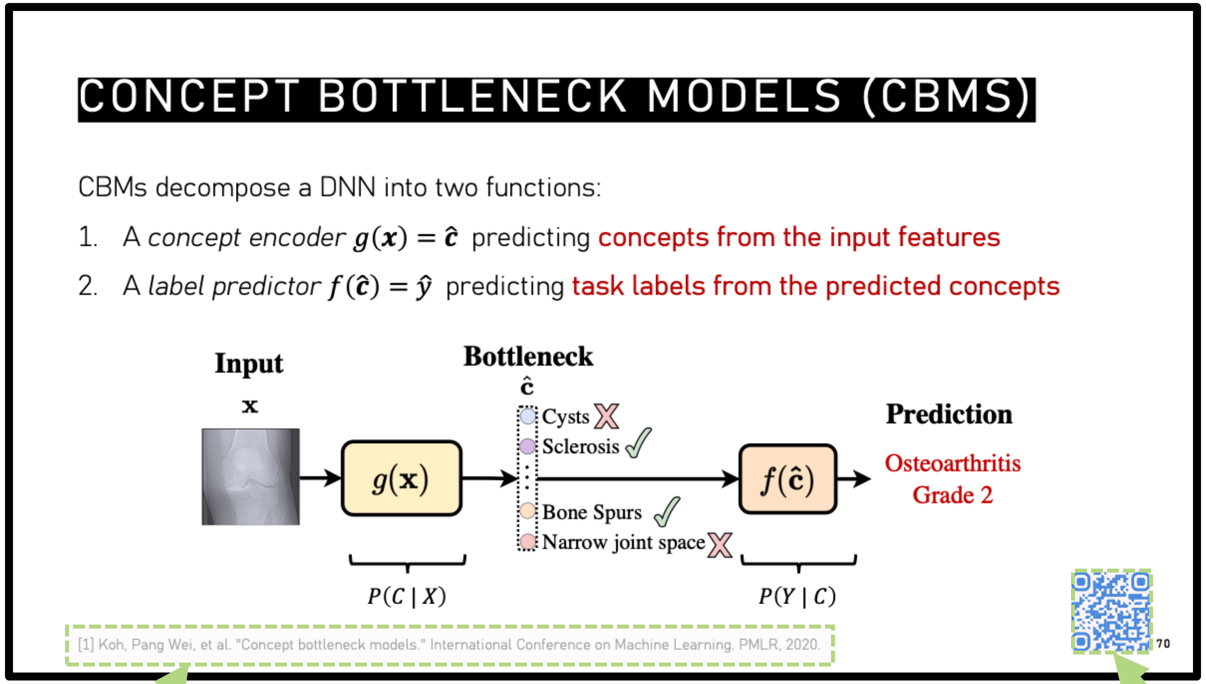
TUTORIAL WEBSITE AND MATERIALS

This tutorial's **slides**, **schedule**, and **resources** are in our website:

https://conceptlearning.github.io/

TUTORIAL WEBSITE AND MATERIALS

Throughout the tutorial, watch for QR codes to relevant references



CONCEPT BOTTLENECK MODELS (CBMS)

CBMs decompose a DNN into two functions:

1. A *concept encoder* $g(\boldsymbol{x}) = \hat{\boldsymbol{c}}$ predicting concepts from the input features
2. A *label predictor* $f(\hat{\boldsymbol{c}}) = \hat{\boldsymbol{y}}$ predicting task labels from the predicted concepts

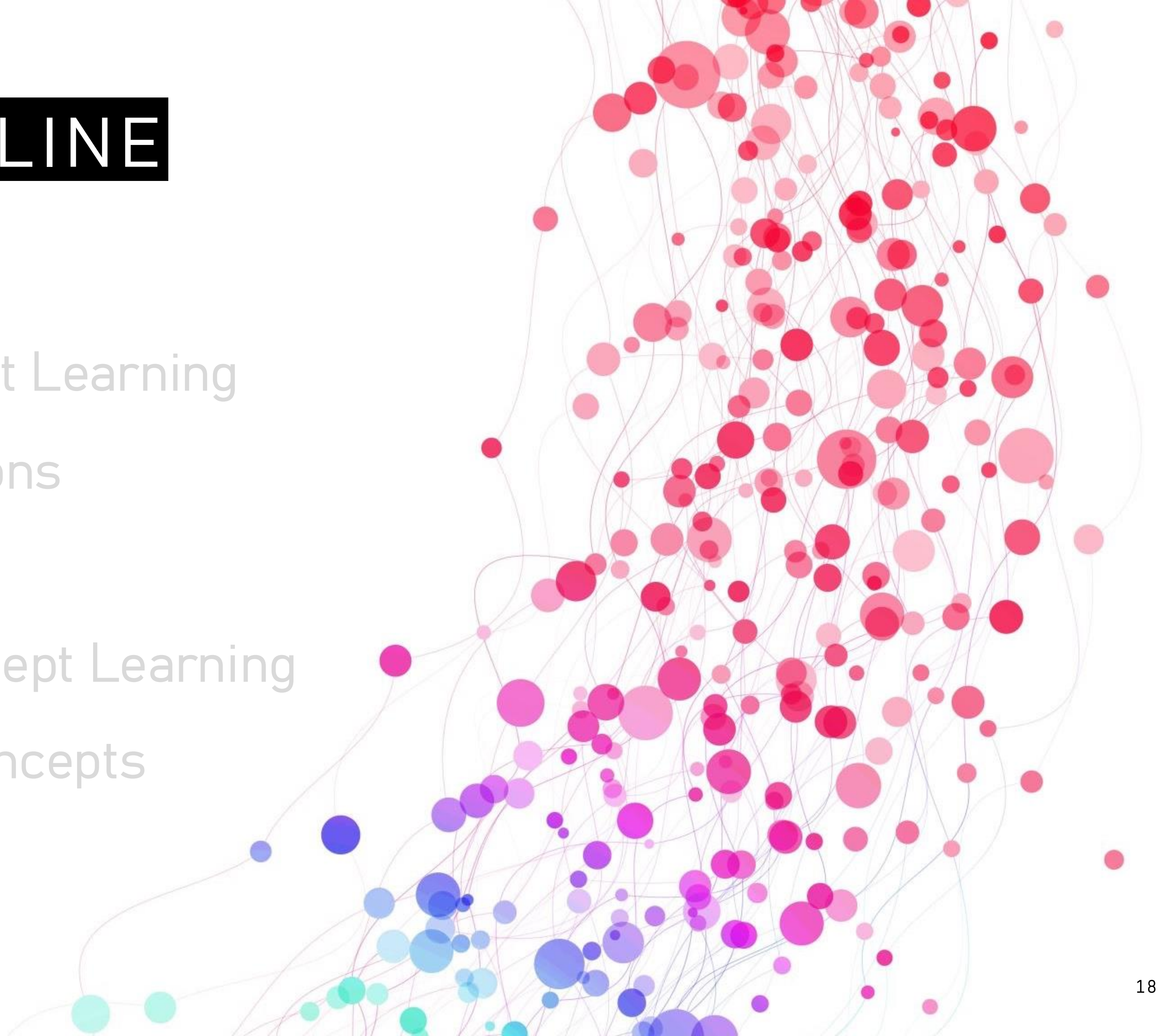
[1] Koh, Pang Wei, et al. "Concept bottleneck models." International Conference on Machine Learning. PMLR, 2020.

Citation + Hyperlink
(if you download slides)

QR code to paper/reference/extra material

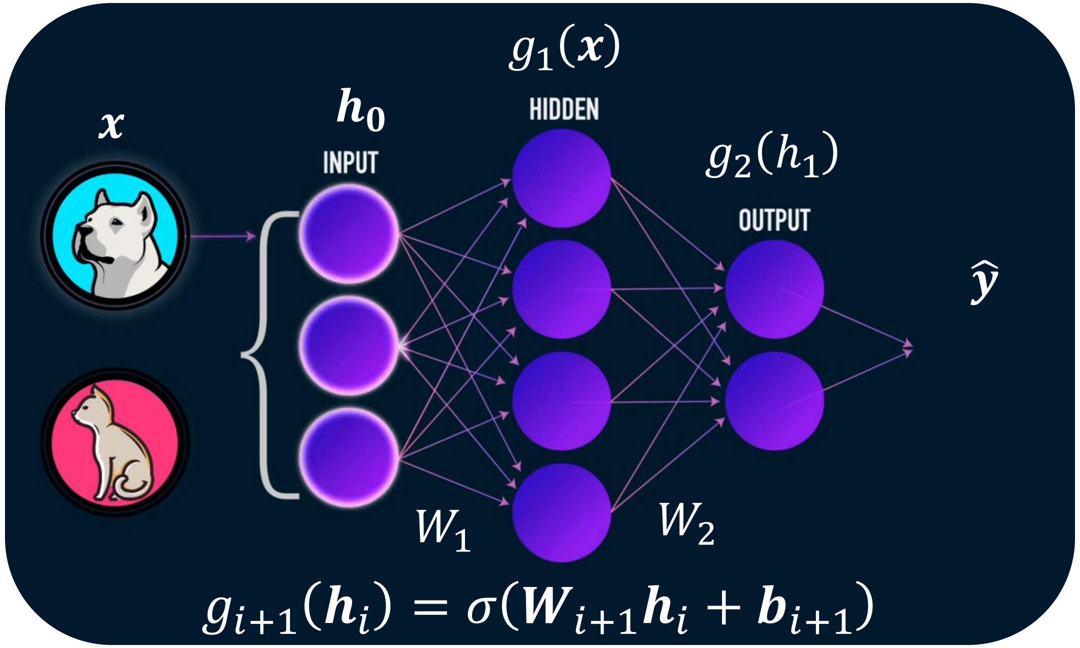
TUTORIAL OUTLINE

1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break
5. Unsupervised Concept Learning
6. Reasoning With Concepts
7. Future Directions
8. Q&A



A SWISS ARMY KNIFE FOR AI

Artificial Intelligence (AI) has experienced **a boom** in the last decade driven by so-called **Deep Neural Networks (DNNs)**

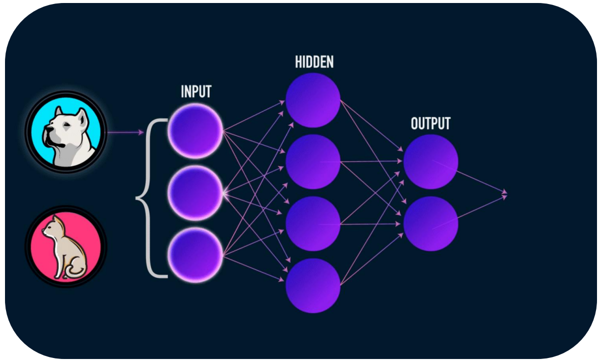


Goal: learn $\{W_1, b_1, \cdots W_m, b_m\}$ s.t. $(g_m \circ \cdots \circ g_2 \circ g_1)(\boldsymbol{x}) = \hat{y} \approx y$

[1] Image adapted from Deep Learning Courses at "https://deeplearningcourses.com/c/deep-reinforcement-learning-in-python"

THE POWER OF SCALE

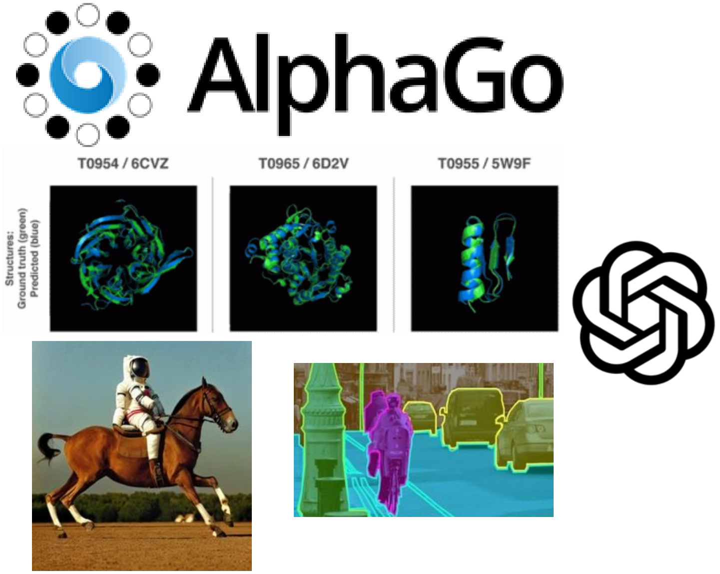
Scaling up DNNs can lead to expressive and generalisable models:



+

A LOT of data,
money, time,
and sweat

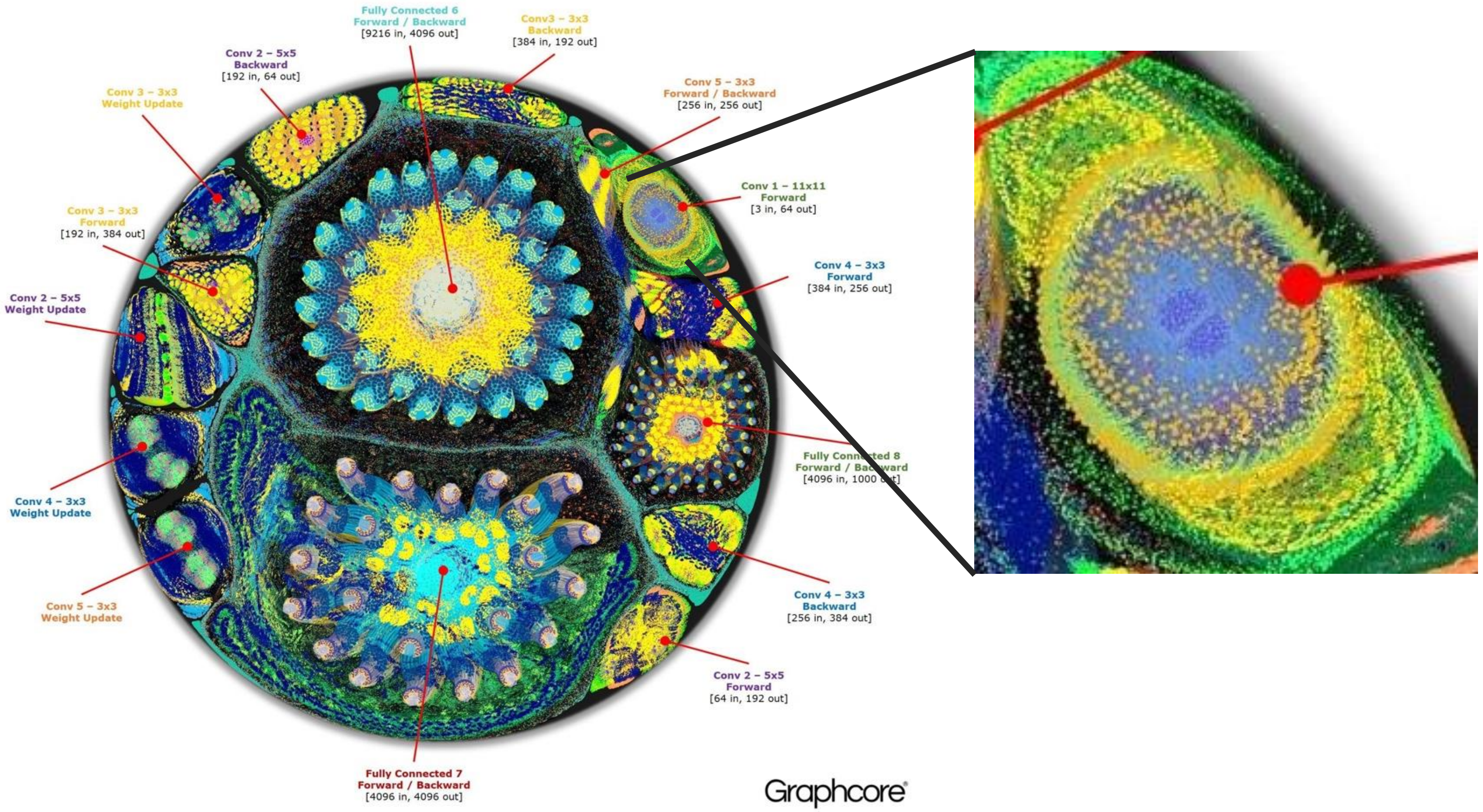
=



[1] Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." nature 529.7587 (2016): 484-489.
[2] OpenAI. "GPT-4 technical report." arXiv (2023).
[3] Jumper, John, et al. "Highly accurate protein structure prediction with AlphaFold." nature 596.7873 (2021): 583-589.
[4] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." CVPR (2022).

THE BLACK-BOX PROBLEM

Scale, however, leads to notoriously complex models!



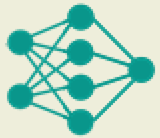
[1] Adapted from Graphcore, "Inside an AI 'Brain' – What does Machine Learning Look Like?" (2017)

THE BLACK-BOX PROBLEM

Scale, however, leads to notoriously complex models!

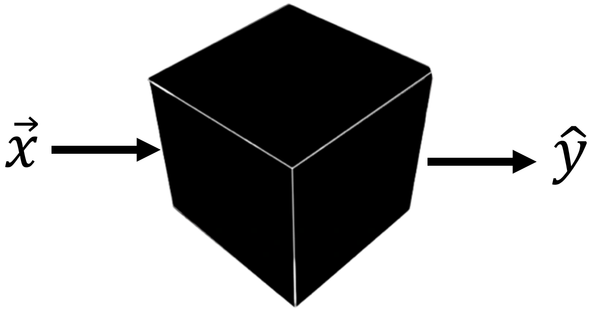
Deep Neural Networks

- Highly parametric
- Complex forward passes
- Continuous activations
- Sensitive to initial states and update rules



≈

Black box Functions



DNNs are "black-box" models

THE FLIP SIDE OF THE COIN

Blindly using black-box models can lead to all sorts of problems:

Wrongfully Accused by an Algorithm

In what may be the first known case of its kind, a faulty facial recognition match led to a Michigan man's arrest for a crime he did not commit.

Why Amazon's Automated Hiring Tool Discriminated Against Women

Predictive policing algorithms are racist. They need to be dismantled.

[1] Kashmir Hill. "Wrongfully Accused by an Algorithm." The New York Times (2020).
[2] Rachel Goodman. "Why Amazon's Automated Hiring Tool Discriminated Against Women." ACLU (2018).
[3] Will Douglas Heaven. "Predictive policing algorithms are racist. They need to be dismantled." MIT Technology Review (2020).

THE FLIP SIDE OF THE COIN

Blindly using black-box models can lead to all sorts of problems:

It's not all bad news ☺

Why Amazon's Automated Hiring Tool Discriminated Against Women

Predictive policing algorithms are racist. They need to be dismantled.

[1] Natasha Bernal. "IBM Watson AI criticised after giving 'unsafe' cancer treatment advice." The Telegraph (2018).
[2] Kashmir Hill. "Wrongfully Accused by an Algorithm." The New York Times (2020).
[3] Rachel Goodman. "Why Amazon's Automated Hiring Tool Discriminated Against Women." ACLU (2018).
[4] Will Douglas Heaven. "Predictive policing algorithms are racist. They need to be dismantled." MIT Technology Review (2020).

EXPLAINING DNNS

Recent advances in AI came with a rise in interest in making these models “interpretable”


CIO BLOG

Companies Grapple With AI’s Opaque Decision-Making Process

Uber, Xerox’s PARC, Capital One among organizations investigating how AI solves problems


How to Build Artificial Intelligence We Can Trust

Computer systems need to understand time, space and causality. Right now they don’t.


Opinion Artificial intelligence

Beware the rise of the black box algorithm


FORBES › INNOVATION › ENTERPRISE TECH

Building Trust In AI: The Case For Transparency


WHO calls for safe and ethical AI for health

16 May 2023 | Departmental update |Reading time: 2 min (507 words)


Why businesses need explainable AI—and how to deliver it

September 29, 2022 | Article

EXPLAINING DNNS

This interest has manifested itself at the regulatory/legal level

General Data Protection Regulations (GDPR, 2016):

- "The data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling,…" (Art. 22)
- The data subject has the right to "meaningful information about the logic involved" in the decision. (Art. 13 and 15)

EU AI Act (2024):

- "Any affected person subject to a decision which is taken by.. a high-risk AI system … shall have the right to obtain from the deployer clear and meaningful explanations (Art. 86)

GDRP

EU AI Act

[1] GDPR, EU. "Automated individual decision-making, including profiling." (2022).
[2] Act, EU Artificial Intelligence. "The EU Artificial Intelligence Act." (2024).

EXPLAINING DNNS

Researchers in Explainable Artificial Intelligence (XAI*) have developed a significant number of methods to explain DNNs

DARPA

Explainable Artificial Intelligence (XAI)

(DARPA 2016)

TAILOR

(EU Horizon Program)

*Not to be confused with a certain bird-related company

FIRST THINGS FIRST: TERMINOLOGY

Welcome to the Wild West of XAI terminology

Goebel et al. (2018)

Freiesleben et al. (2023)

Gilpin et al. (2018)

Vilone et al. (2021)

Confalonieri et al. (2020)

Rudin et al. (2019)

Barredo Arrieta et al. (2020)

FIRST THINGS FIRST: TERMINOLOGY

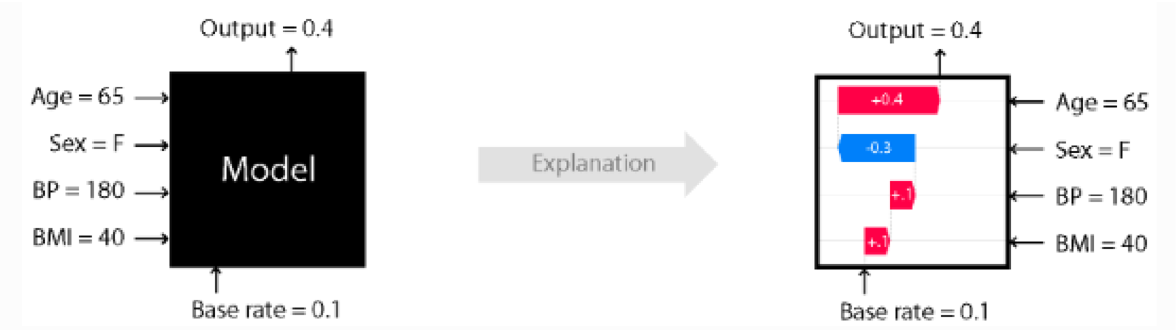
Here we will use some the following definitions by Gilpin et al. [1]:

- **Explainability (why)**: the ability to answer questions of the form "*why does this particular input lead to that particular output?*"
- **Interpretability (how)**: the ability to describe "the internals of a system in a way that is understandable to humans."

[1] Gilpin et al. "Explaining explanations: An overview of interpretability of machine learning." DSAA (2018).

FEATURE ATTRIBUTION

XAI methods have traditionally explained a model's prediction by estimating how important each input feature is for the output



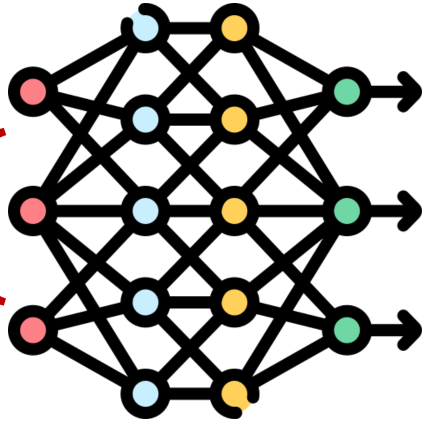
SHAP (Scott et al., 2017)

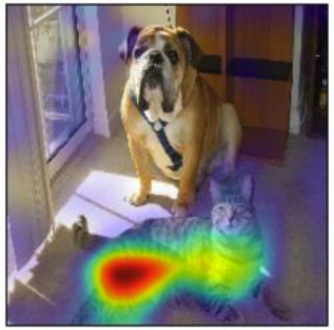
We call these feature importance or feature attribution methods

[1] Scott, M., and Lee Su-In. "A unified approach to interpreting model predictions." NeurIPS (2017).

SALIENCY: FEATURE ATTRIBUTION IN DNNS

DNN-specific attribution methods are called saliency methods

$\psi($

$f(\boldsymbol{x}\,;\theta)$
$,$

$\boldsymbol{x}$
$, \text{cat})$
$\hat{y}$
$=$


These are usually computed by measuring model sensitivity via its gradient $\frac{\partial f(\boldsymbol{x})_y}{\partial x_i}$

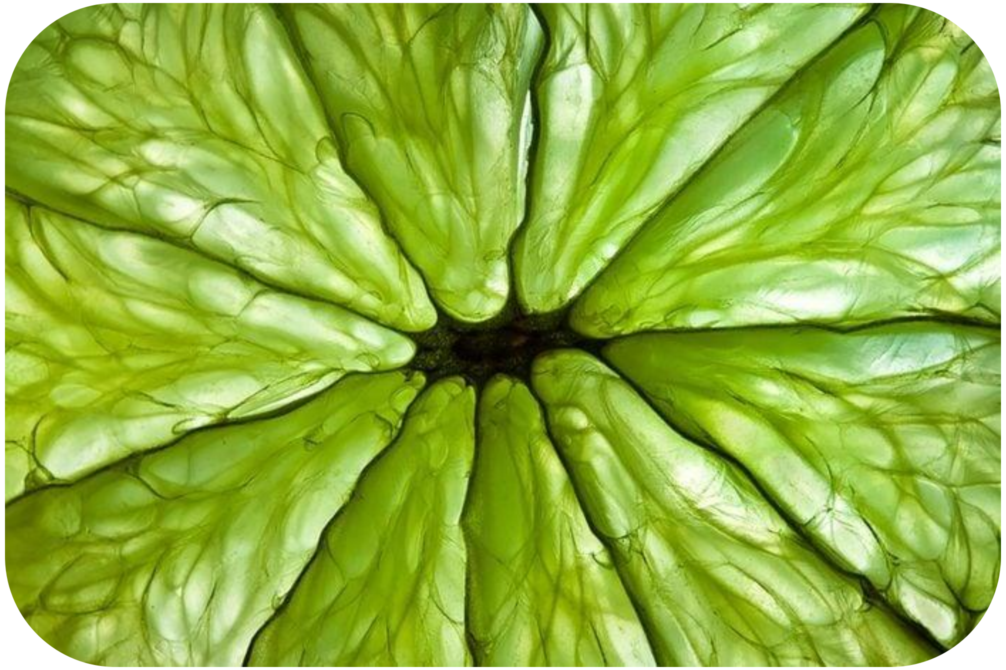
---

[1] Example taken from Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." ICCV (2017).

WHAT'S WRONG WITH FEATURE ATTRIBUTION?

WHAT'S WRONG WITH FEATURE ATTRIBUTION?

1. Low-level features like individual pixels are not always semantically meaningful:



Can you guess what this is?

[1] Andrey Armyagov/Shutterstock

WHAT'S WRONG WITH FEATURE ATTRIBUTION?

1. Low-level features like individual pixels are not always semantically meaningful:

Limes!

[1] ZaZa Studio/Shutterstock

WHAT'S WRONG WITH FEATURE ATTRIBUTION?

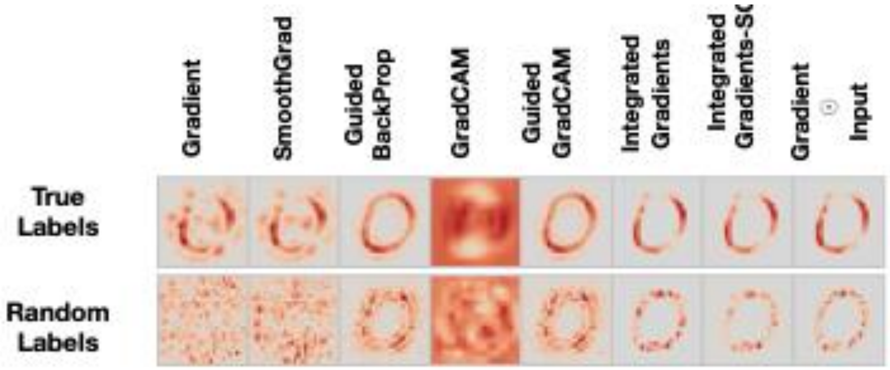
2. Saliency maps lack of actionability!

What does this really tell you about how the model made a prediction?

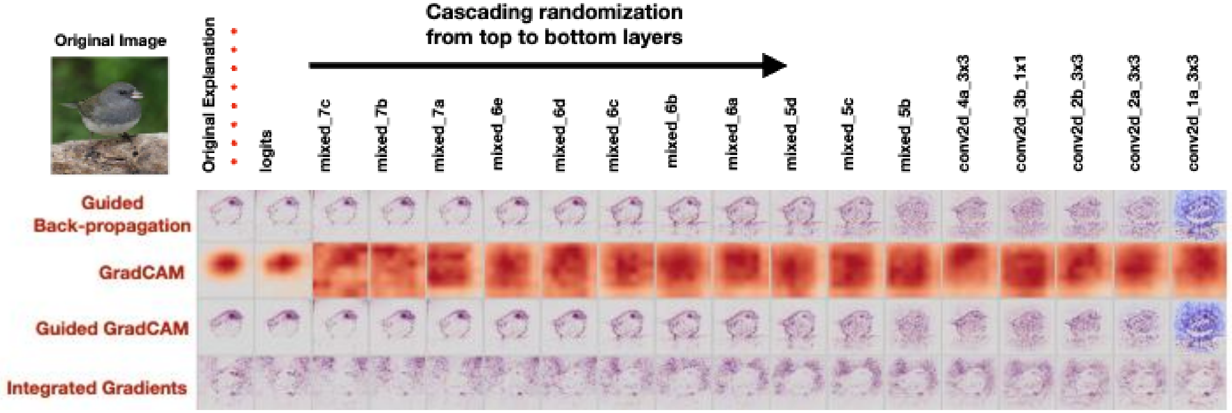
[1]Image adapted from Dombrowski et al. "Explanations can be manipulated and geometry is to blame." NeurIPS (2019).

WHAT'S WRONG WITH FEATURE ATTRIBUTION?

3. Several saliency methods fail very simple sanity checks



Random training labels do not always lead to random maps [1]



Random weights do not always lead to random maps [1]

[1] Adebayo, Julius, et al. "Sanity checks for saliency maps." NeurIPS (2018).

WHAT'S WRONG WITH FEATURE ATTRIBUTION?

4. Saliency methods are susceptible to adversarial attacks [1,2]

[1] Dombrowski et al. "Explanations can be manipulated and geometry is to blame." NeurIPS (2019).
[2] Also relevant Ghorbani et al. "Interpretation of neural networks is fragile." AAAI (2019).

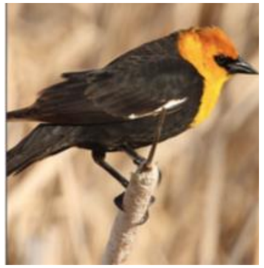
WHAT'S WRONG WITH FEATURE ATTRIBUTION?

How can we go around the limitations of feature attribution?

Here, we will focus on using so-called "*concepts*" to construct explanations

WHAT ARE CONCEPTS?

Concepts are high-level and semantically meaningful units of information

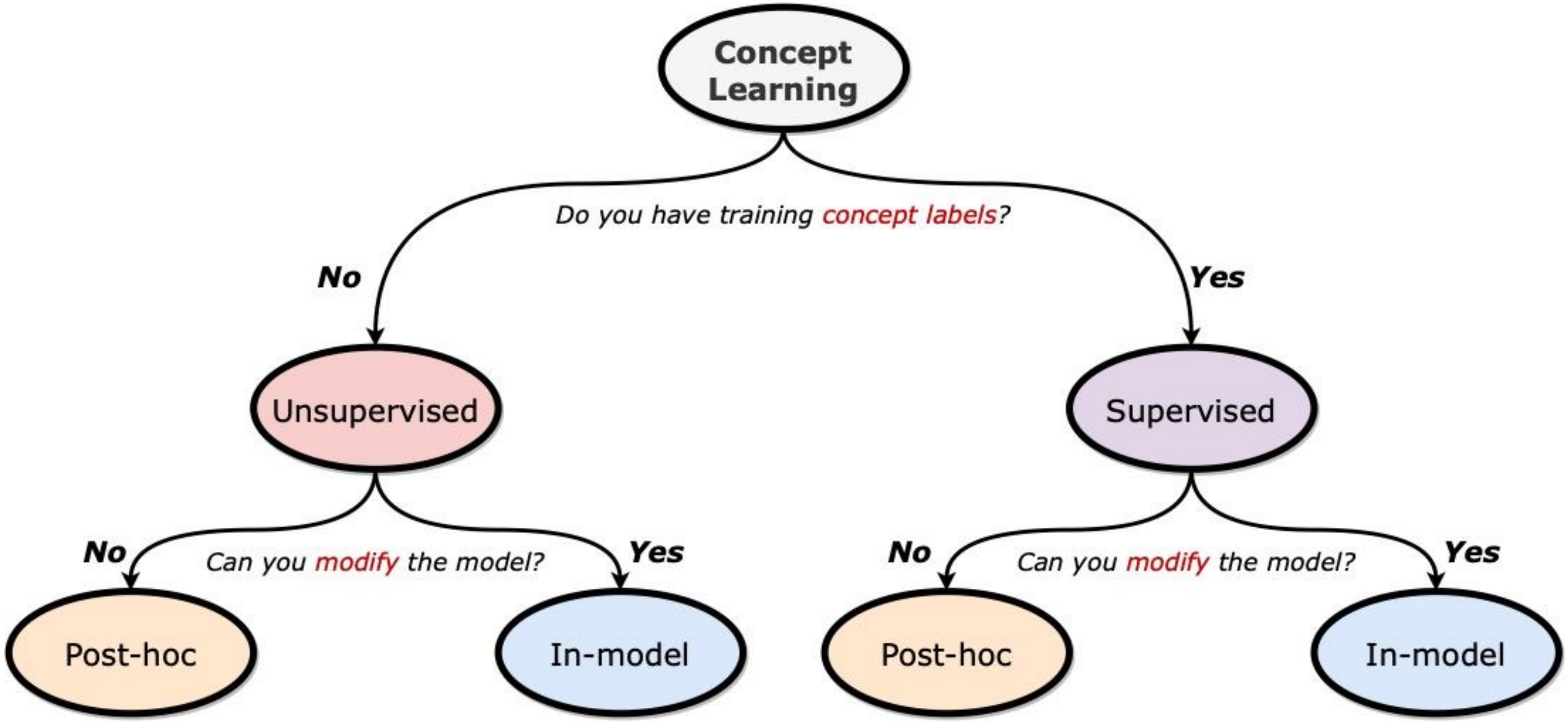


Task: bird species

Explanation of the prediction:
- wing color
- beak length
- tail shape

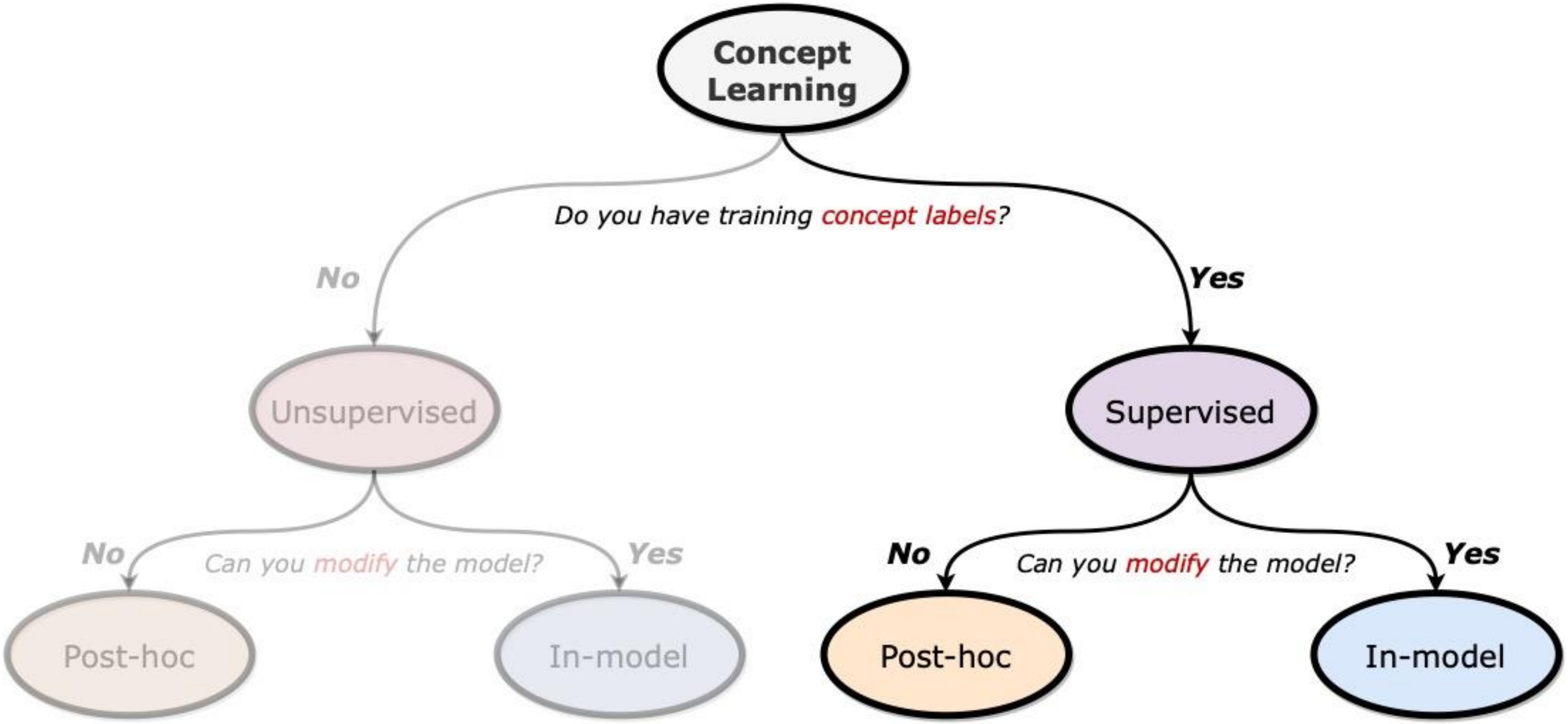
Concepts are terms or units of information used by domain experts to communicate or explain things to each another

A MAP OF CONCEPT LEARNING



[1] For a more complete taxonomy, see Poeta et al. "Concept-based explainable artificial intelligence: A survey." arXiv (2023).

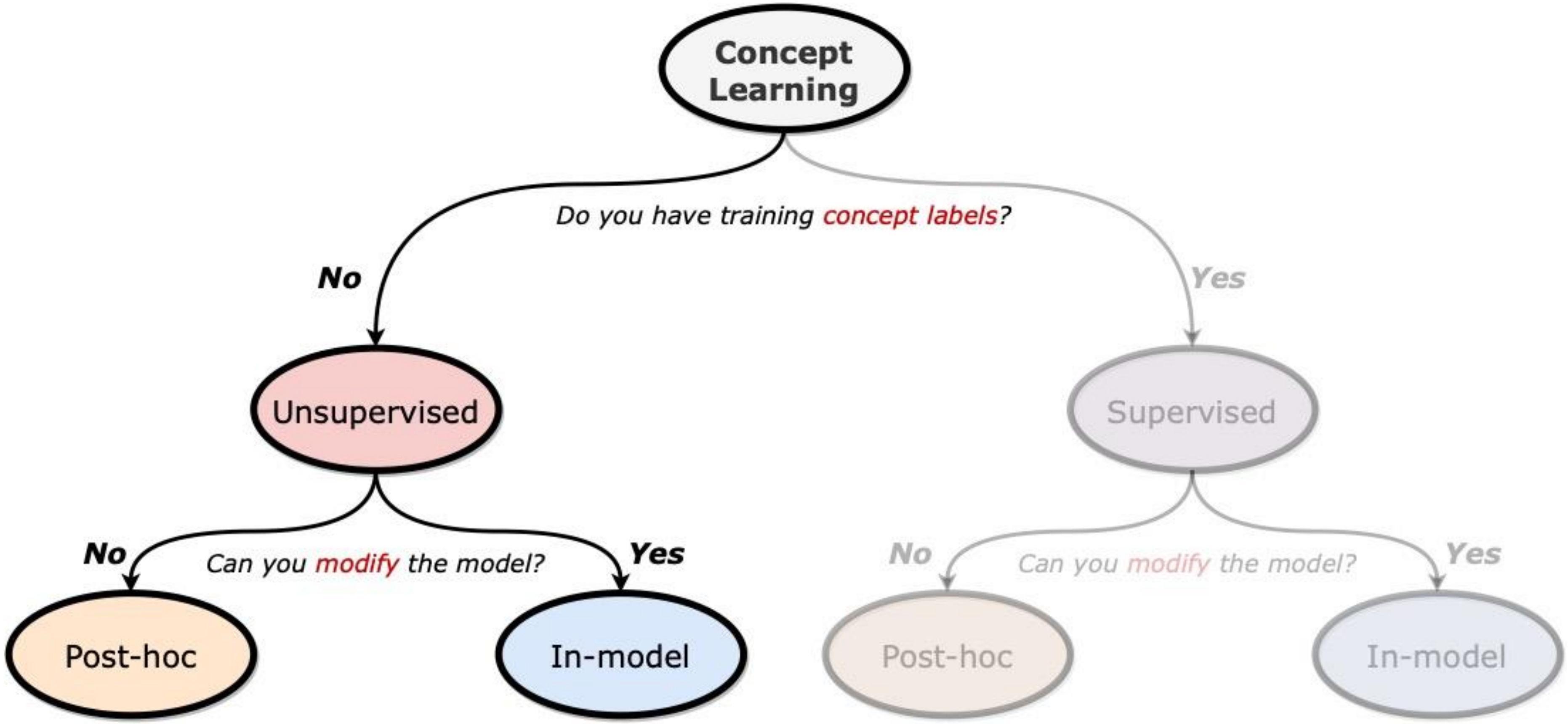
A MAP OF CONCEPT LEARNING



1/3

In the first third of this tutorial, we will discuss supervised concept learning

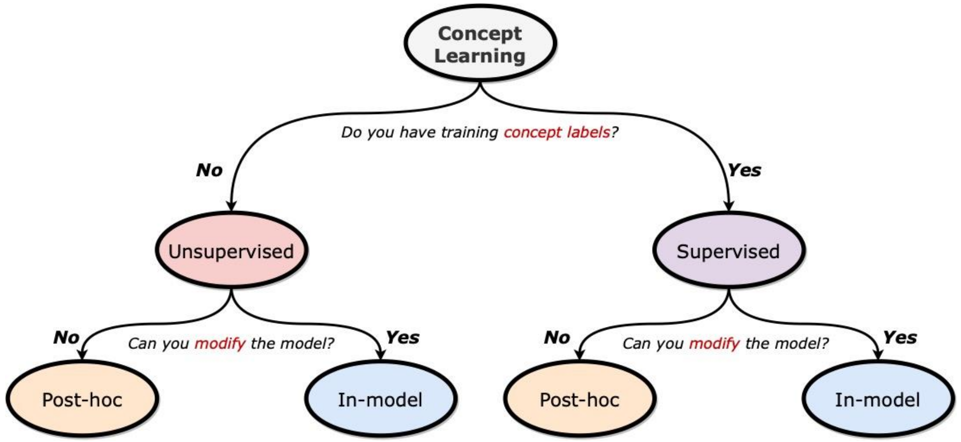
A MAP OF CONCEPT LEARNING



2/3

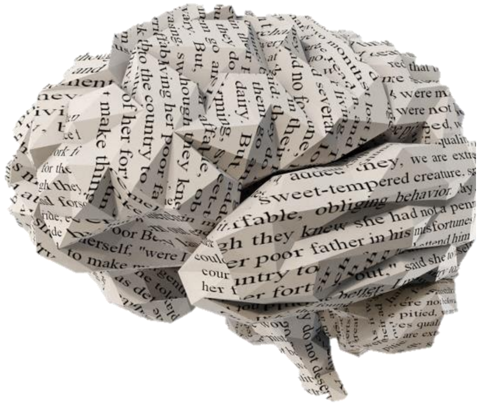
The second third discusses unsupervised concept learning approaches

A MAP OF CONCEPT LEARNING



Concepts

+



Reasoning

3/3

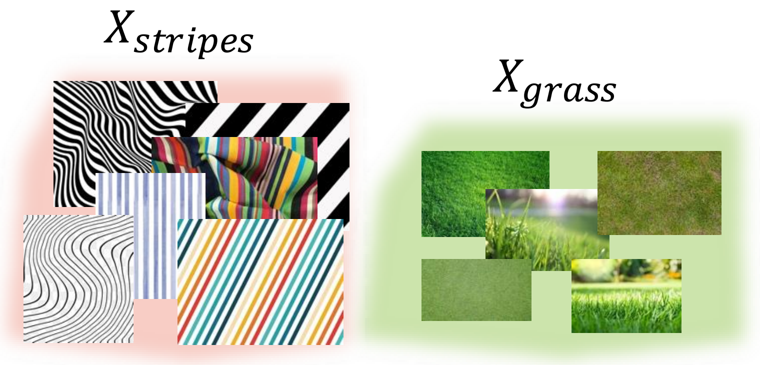
Finally, in the last third we discuss applications of CL to symbolic reasoning

TUTORIAL OUTLINE

1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break
5. Unsupervised Concept Learning
6. Reasoning With Concepts
7. Future Directions
8. Q&A

DIFFERENT LEVELS OF SUPERVISION

"Supervised" is a **loaded term**. In this tutorial's context, "**supervised**" means a method has **access to "concept" labels**



Sparse sets of images containing a concept

OR

$x^{(i)}$

$$\mathbf{c}^{(i)} = \begin{pmatrix}\text{lays eggs}\\ \text{has scales}\\ \text{has wings}\\ \text{eats only plants}\\ \text{eats meat}\\ \text{black wings}\\ \text{is colorful}\\ \text{has teeth}\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\1\\0\\0\end{pmatrix}$$

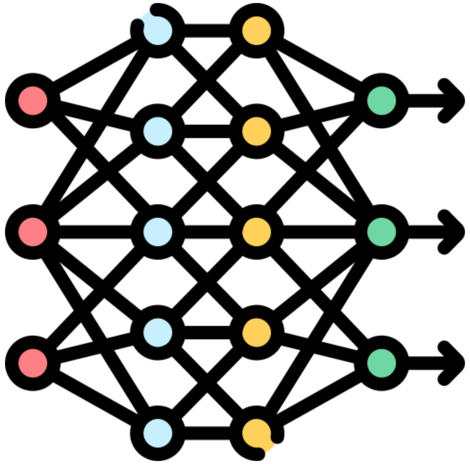
Dense binary vector annotations

These labels could come **besides** other downstream **task "labels"**

POST-HOC CONCEPT LEARNING

We will start by looking into supervised **post-hoc** concept learning:

A **trained** DNN $f(\boldsymbol{x}; \theta)$



Sparse concept annotations

$X_{stripes}$

X_{grass}

Concept-based **Explanations**

Local

How important is "*stripes*" for a prediction?

Global

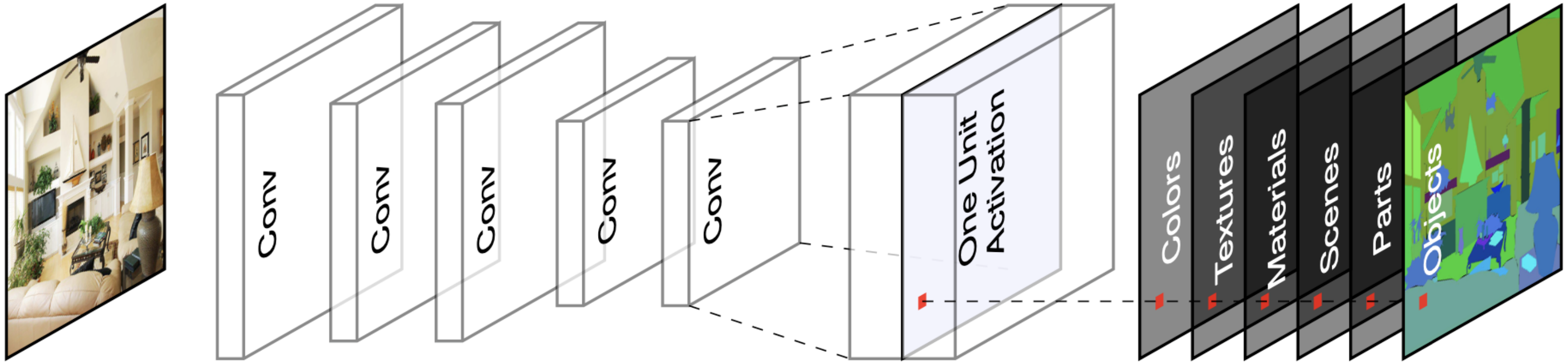
how important is "*grass*" for the class "*cow*"?

Given

We Want

WHY SHOULD WE EVEN ATTEMPT THIS?

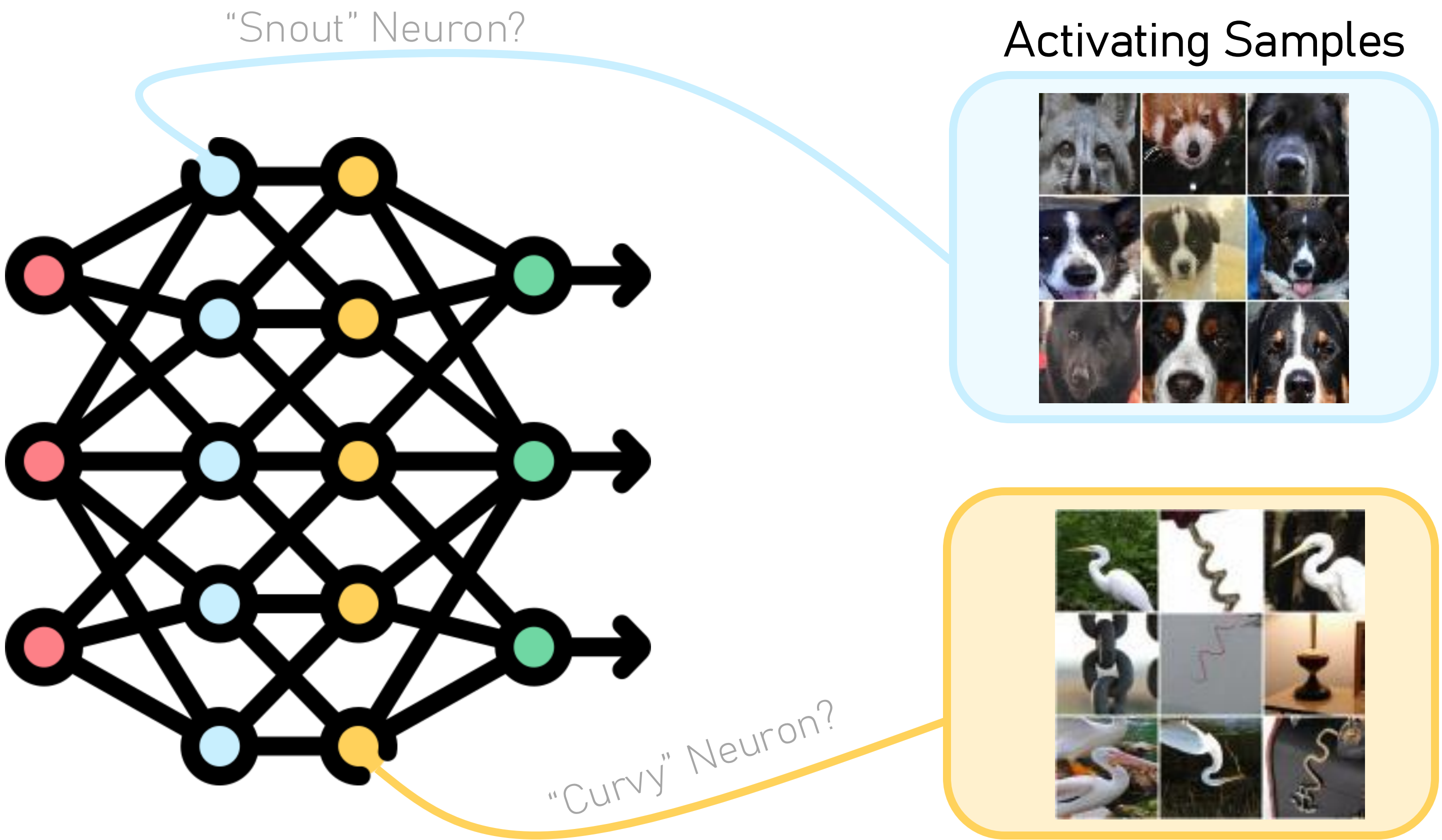
Evidence suggests DNNs may predict based on concepts



[1] Bau, David, et al. "Network dissection: Quantifying interpretability of deep visual representations." CVPR (2017).

WHY SHOULD WE EVEN ATTEMPT THIS?

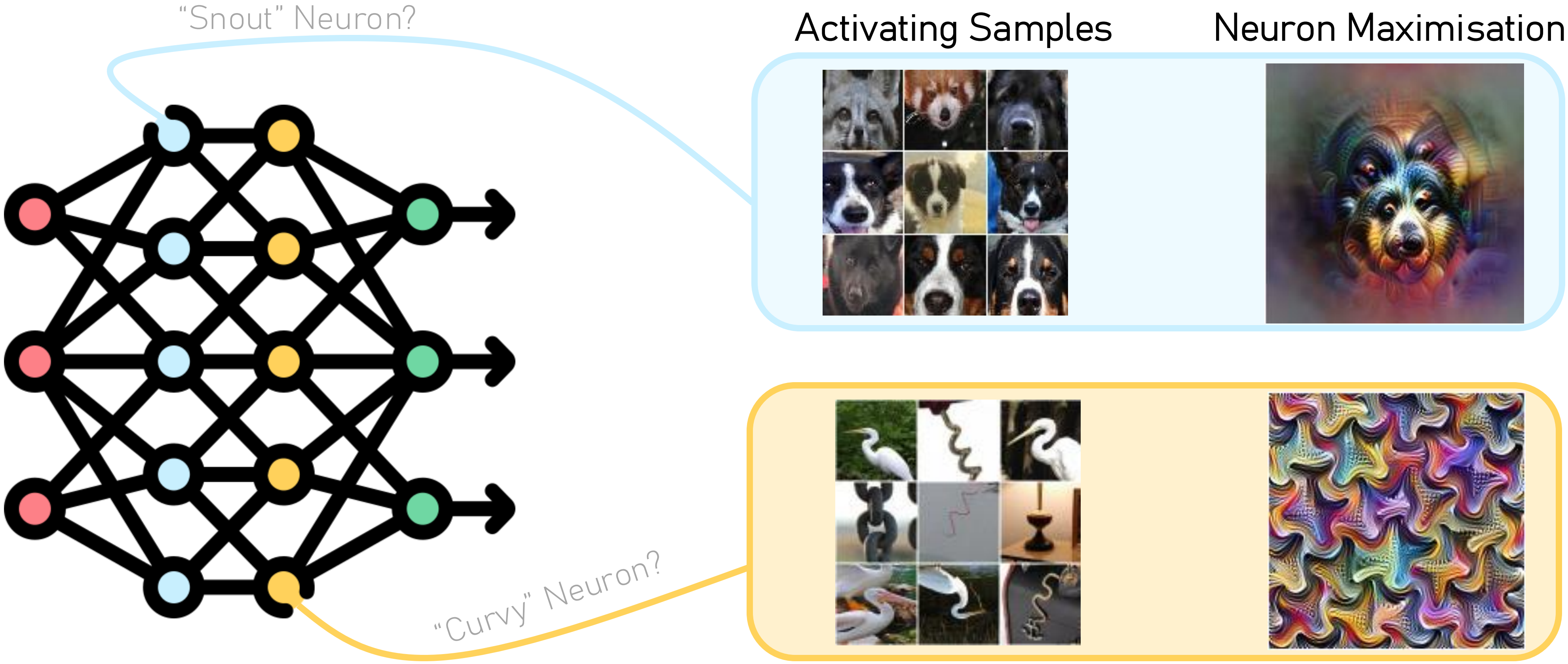
Evidence suggests DNNs **may** predict based on **concepts**



[1] Examples taken from Olah et al. "Feature visualization." Distill 2.11 (2017).

WHY SHOULD WE EVEN ATTEMPT THIS?

Evidence suggests DNNs may predict based on concepts



“Snout” Neuron?

Activating Samples

Neuron Maximisation

“Curvy” Neuron?

[1] Examples taken from Olah et al. "Feature visualization." Distill 2.11 (2017).

CONCEPTS ARE NOT ALWAYS LOCALISED

Concepts may not be always localised to specific neurons/maps but they may be distributed across the DNN's latent space

The same units appear to represent different concepts

This is sometimes called Polysemanticity (Olah et al., 2020)

[1] Fong et al. "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks." CVPR (2018).

CONCEPTS ARE NOT ALWAYS LOCALISED

Concepts may **not be always localised** to specific neurons/maps

Could we then try and capture ***directions in the latent space*** *that are* ***associated with known concepts****?*

The same units appear to represent different concepts

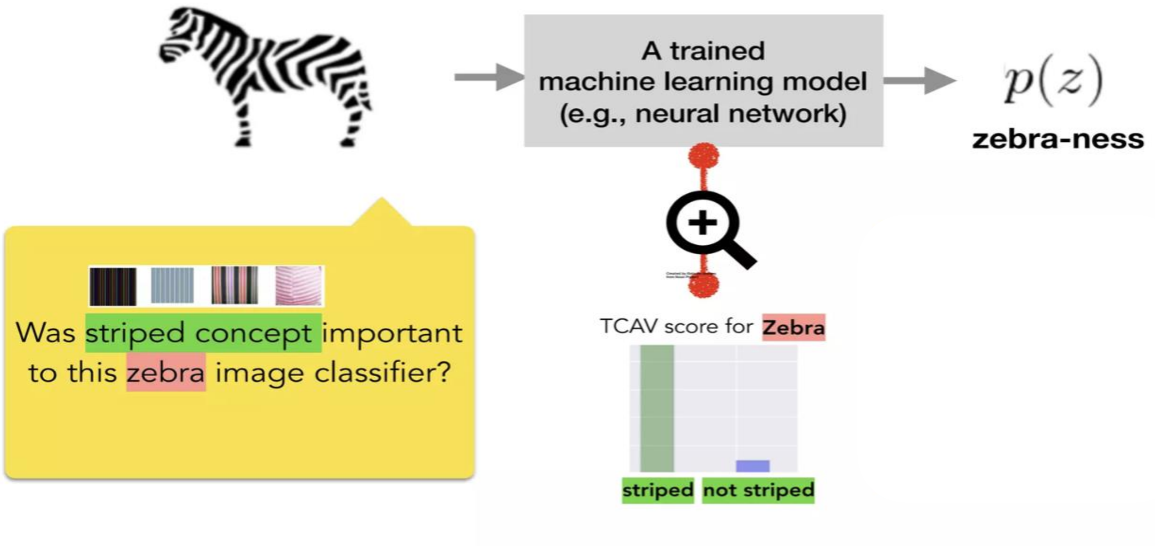
This is sometimes called **Polysemanticity** (Olah et al., 2020)

[1] Fong et al. "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks." CVPR (2018).

TESTING WITH CONCEPT ACTIVATION VECTORS

This is the idea behind T-CAV (Testing with concept activation vectors)

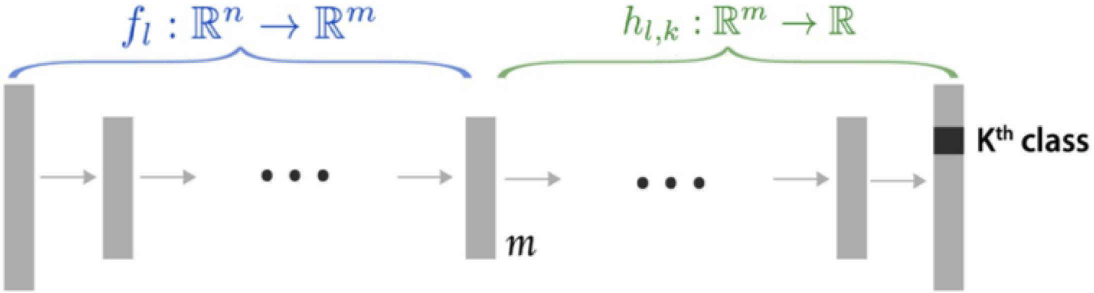
How sensitive is the prediction of *zebra* is to the presence of the concept of "*stripes*"?



[1] Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." ICML (2018).

PARTITIONING THE NETWORK

Step 1: Choose an intermediate layer $f_l\!: R^n \rightarrow R^m$ with m neurons



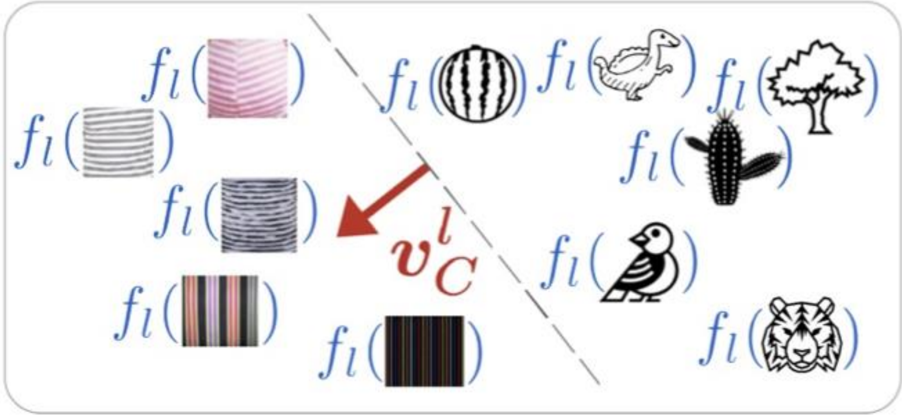
[1] Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." ICML (2018).

LEARNING CONCEPT ACTIVATION VECTORS

Step 2: Learn the *Concept Activations Vectors (CAVs)*

- Train a linear classifier to distinguish between the activations of concept's examples and random ones
- The CAV is the vector orthogonal to the classification boundary v_C^l

set of examples for a concept
and random examples



[1] Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." ICML (2018).

TESTING WITH CAVS (T-CAV)

Step 3: Given a sample $\boldsymbol{x}$, construct a local importance score $S_{C,k,l}(x)$ indicating how important concept $\boldsymbol{C}$ is for the $\boldsymbol{k}$–th output label.

We want $S_{C,k,l}(x)$ to capture "how much would the prediction of class k change if I "increase" concept C in sample $\boldsymbol{x}$?"

$$S_{C,k,l}(x) = \lim_{\epsilon \to 0} \frac{h_{l,k}\left(f_l(\boldsymbol{x}) + \epsilon \boldsymbol{v}_C^l\right) - h_{l,k}\left(f_l(\boldsymbol{x})\right)}{\epsilon}$$

(Read as: how much would the prediction of label k change if I take a small step in the direction of concept C?)

[1] Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." ICML (2018).

TESTING WITH CAVS (T-CAV)

Step 3: Given a sample $\boldsymbol{x}$, construct a local importance score $S_{C,k,l}(x)$ indicating how important concept $\boldsymbol{C}$ is for the $\boldsymbol{k}$–th output label.

We want $S_{C,k,l}(x)$ to capture "how much would the prediction of class k change if I "increase" concept C in sample $\boldsymbol{x}$?"

This is the same as a directional derivative!

[1] Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." ICML (2018).

TESTING WITH CAVS (T-CAV)

Step 3: Given a sample $\boldsymbol{x}$, construct a **local importance score** $S_{C,k,l}(x)$ indicating how important concept $\boldsymbol{C}$ is for the $\boldsymbol{k}$–th output label.

Intermediate representation of  at layer $\boldsymbol{l}$

$S_{C,k,l}(x) = \nabla h_{l,k}\big(f_l(x)\big) \,.\, v_C^l$ = $\nabla h_{l,k}\big(f_l($  $)\big) \,.\, v_C^l$

Output function

CAV for concept $\boldsymbol{C}$ (e.g., stripes)

The rate of change of output function as we move in the direction of a concept from data point 

Intuition: *"high directional derivative" = "large positive change in class label if we 'increase'* $\boldsymbol{C}$ *in input* $\boldsymbol{x}$*"*

[1] Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." ICML (2018).

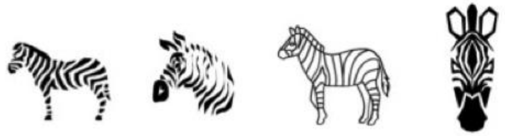
TESTING WITH CAVS (T-CAV)

Step 4: Get a **global importance score (T-CAV)** for each concept by **combining the local sensitivities** of samples in an **evaluation set**:

The **T-CAV score** is the fraction of samples with label $\boldsymbol{k}$ that are positively influenced by concept $\boldsymbol{C}$

$$\mathrm{TCAV}_{Q_{C,k,l}} = \frac{|\{x \in X_k : S_{C,k,l}(x) > 0\}|}{|X_k|}$$

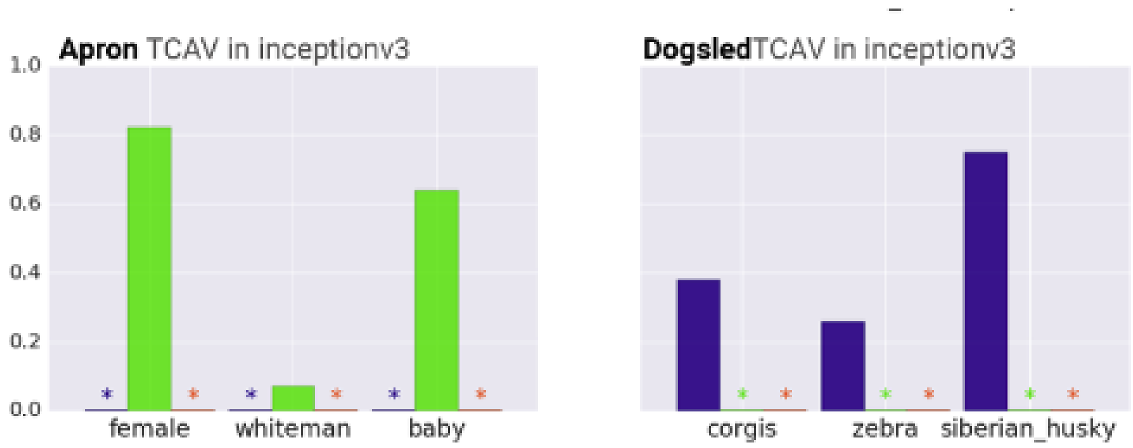
X_k: inputs with label k



[1] Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." ICML (2018).

EXAMPLE: DETECTING BIASES WITH T-CAV

You can use T-CAV scores to explore/identify model biases



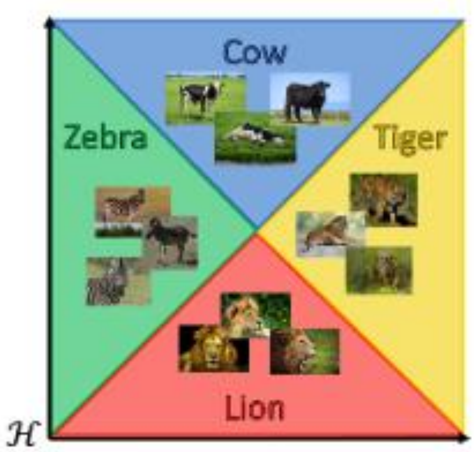
The concept of "female" was found to be significant for predicting the class "Apron"

The concept of "Siberian husky" was found to be significant for predicting the class "Dogsled"

[1] Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." ICML (2018).

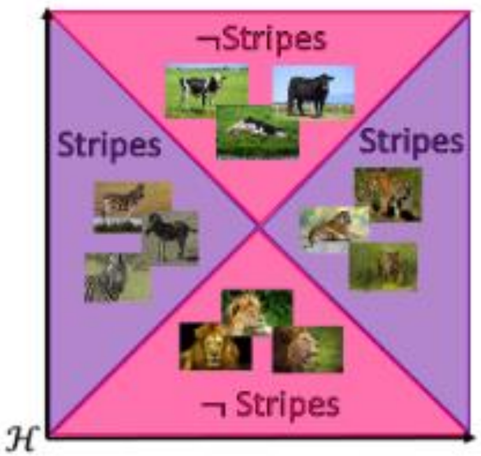
T-CAV LIMITATIONS

Assuming concepts are linearly separable is a strong and unrealistic assumption



Classes can be linearly separable

VS

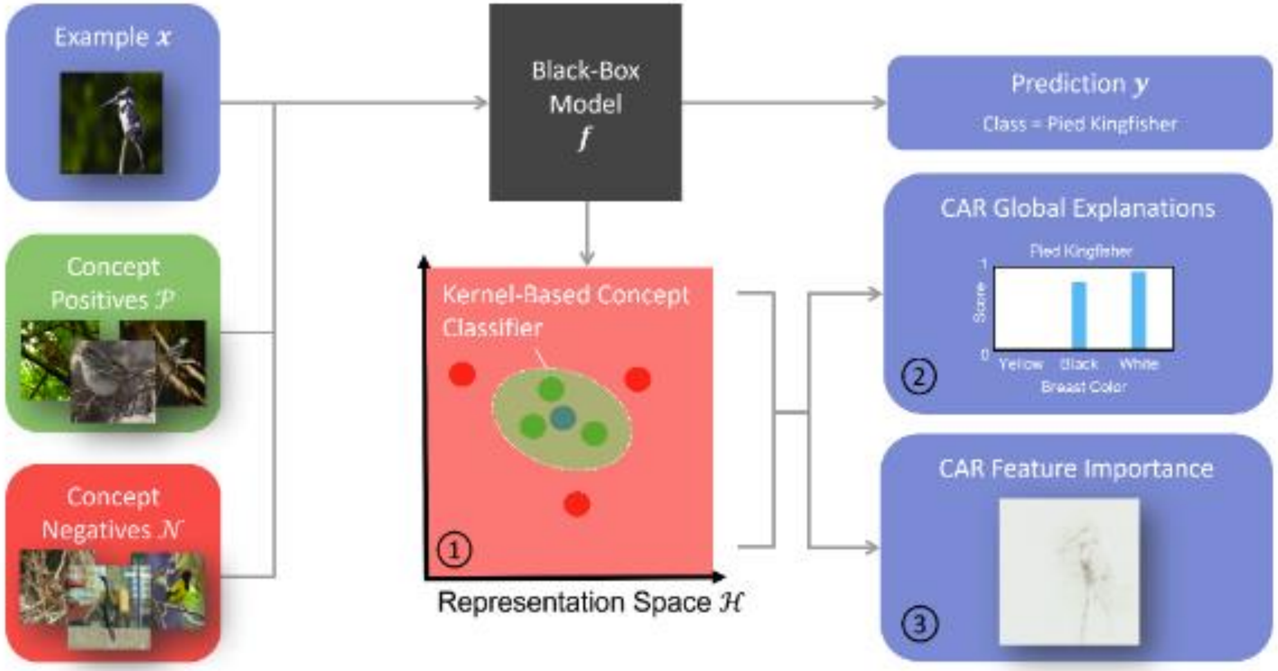


While concepts may not be separable

[1] Crabbé et al. "Concept activation regions: A generalized framework for concept-based explanations." NeurIPS (2022).

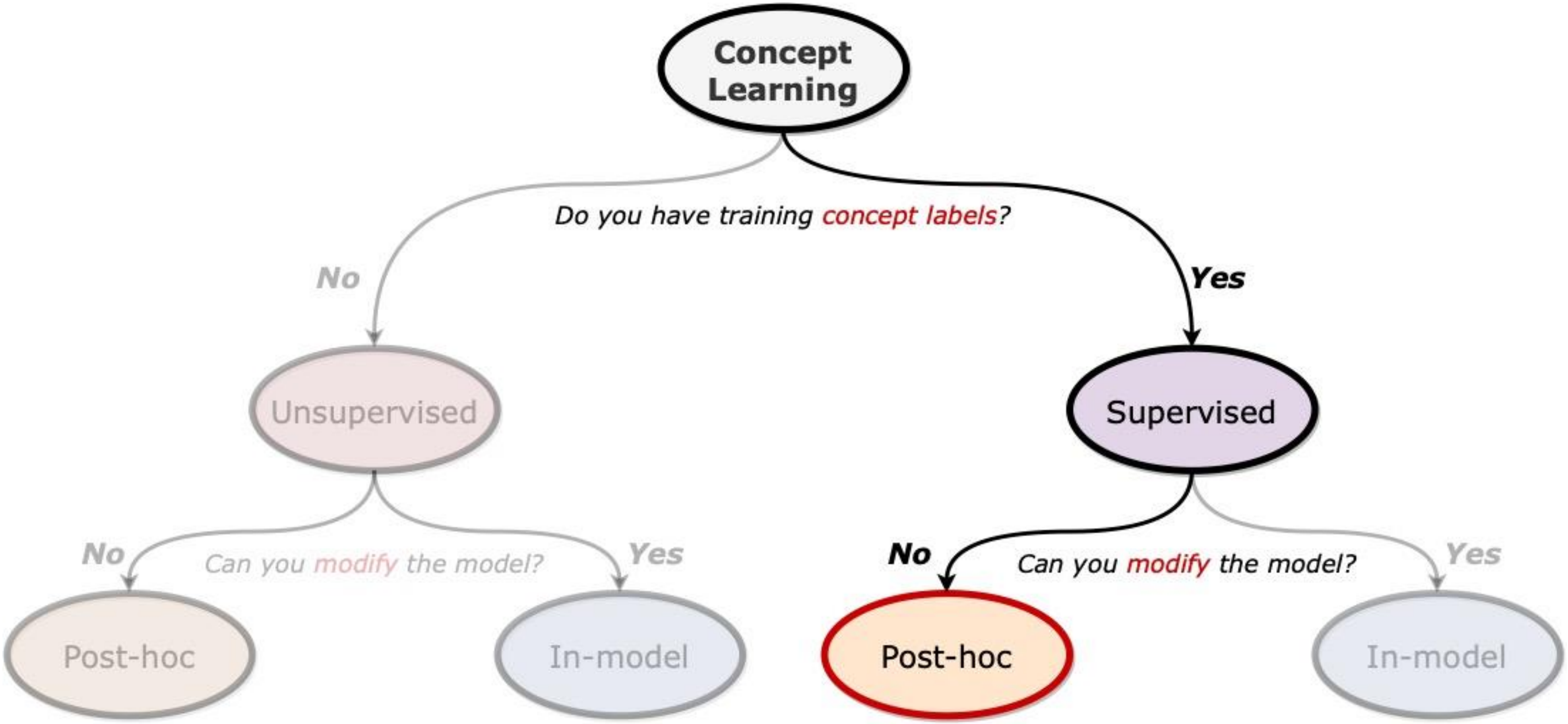
CONCEPT ACTIVATION REGIONS

This can be solved by using **kernel methods** to perform our concept probing **on higher-dimensional space where concepts may be separable**



[1] Crabbé et al. "Concept activation regions: A generalized framework for concept-based explanations." NeurIPS (2022).

THE POST-HOC STORY SO FAR



Concept Learning

Do you have training concept labels?

No

Yes

Unsupervised

Supervised

No

Can you modify the model?

Yes

No

Can you modify the model?

Yes

Post-hoc

In-model

Post-hoc

In-model

THE POST-HOC STORY SO FAR

Post-hoc methods have a clear set of important limitations:

1. They may fail to properly explain a model → potentially doubling the source of error!

THE POST-HOC STORY SO FAR

Post-hoc methods have a clear set of important limitations:

1. They may fail to properly explain a model → potentially doubling the source of error!

In fact, these methods often disagree with each other [1]

[1] Krishna et al. "The disagreement problem in explainable machine learning: A practitioner's perspective." TMLR (2022).

THE POST-HOC STORY SO FAR

Post-hoc methods have a clear set of important limitations:

2. Explanations are prone to confirmation bias [1]

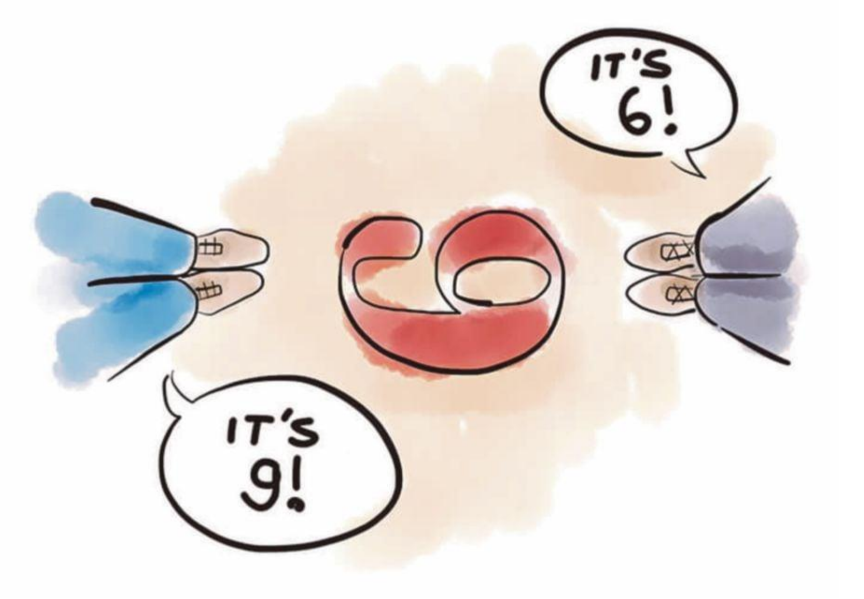
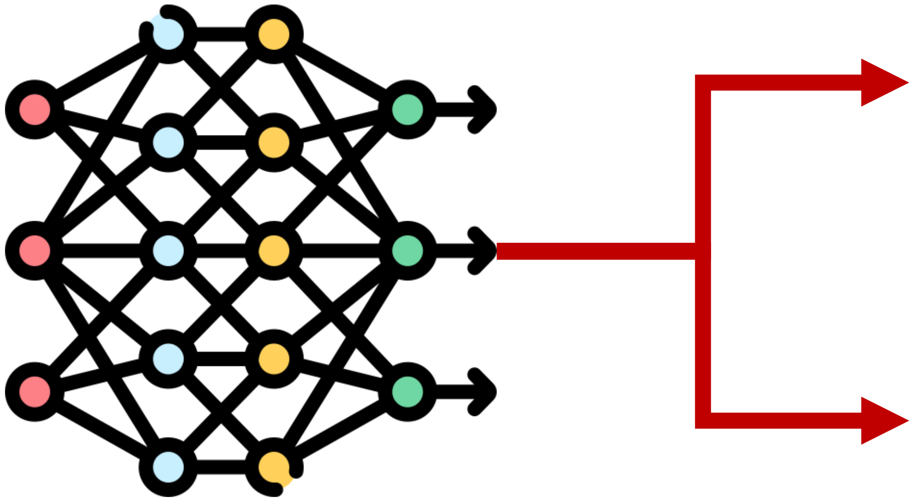


Image taken from "Confirmation Bias and the new Malaysia" by Datuk Steven Wong (New Straits Times)

[1] Bertrand et al. "How cognitive biases affect XAI-assisted decision-making: A systematic review." AIES (2022)

GOING IN-MODEL

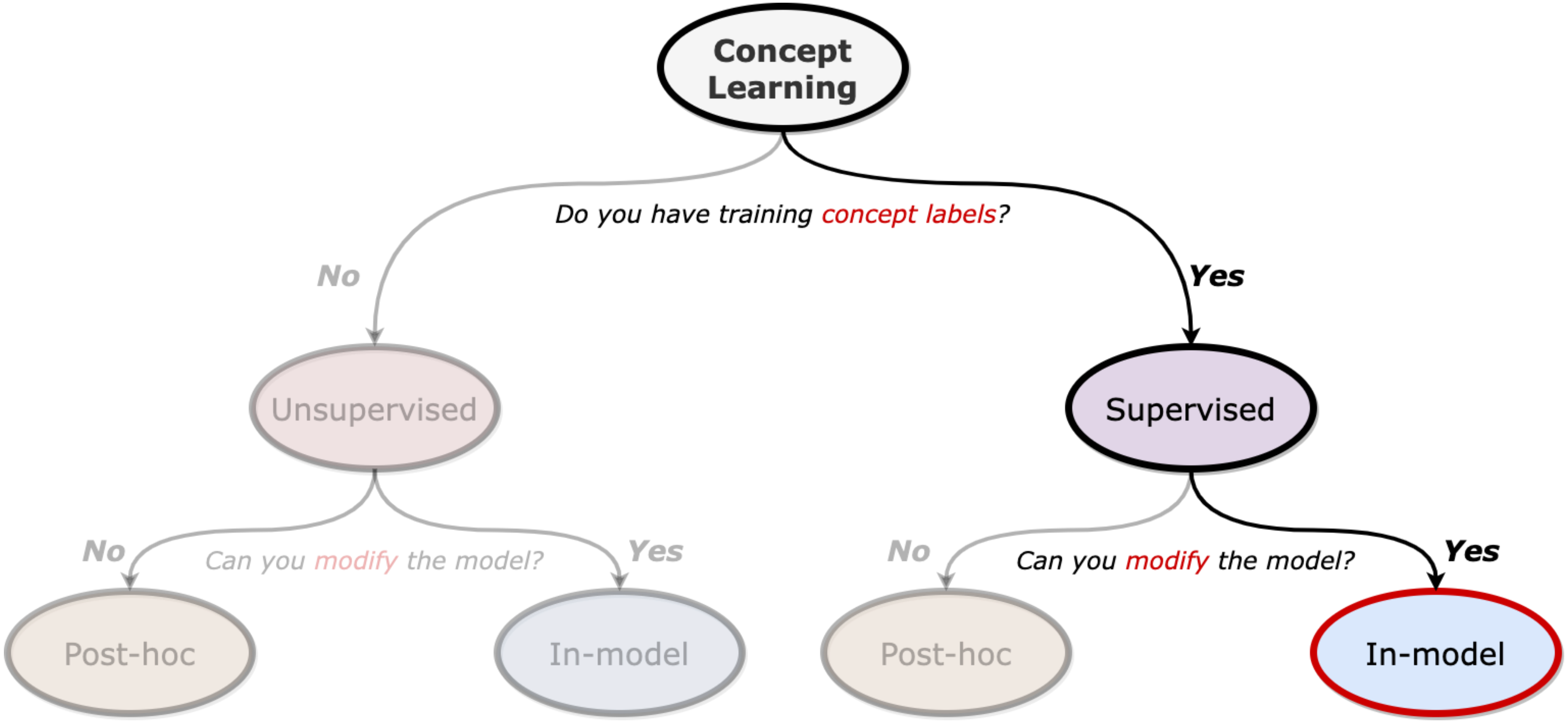
Rather than explaining an already trained model, let the model explain itself!



Prediction

Explanation

GOING IN-MODEL



Concept Learning

Do you have training concept labels?

No

Yes

Unsupervised

Supervised

No

Can you modify the model?

Yes

No

Can you modify the model?

Yes

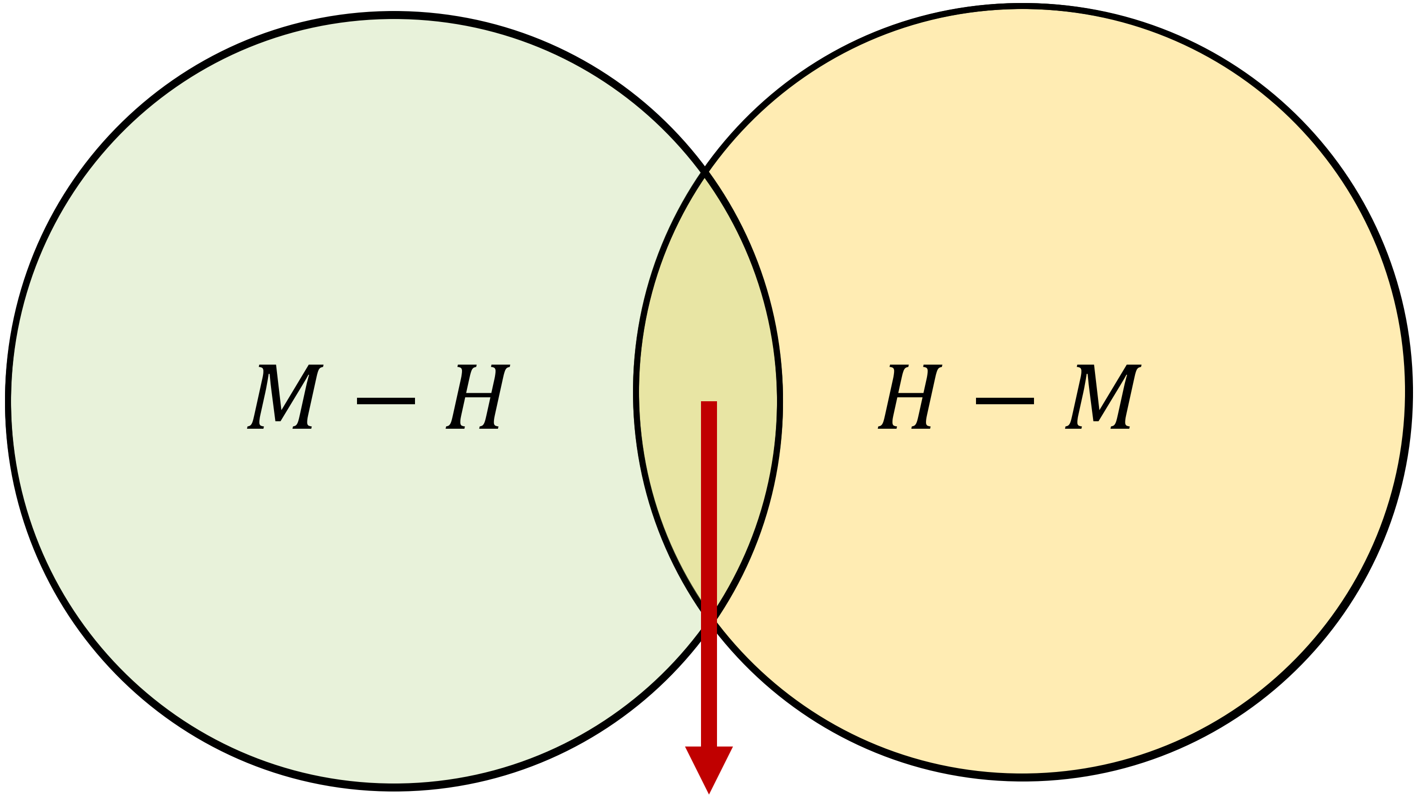
Post-hoc

In-model

Post-hoc

In-model

ALIGNING MACHINES AND HUMANS



M = Machine
Representational Space

H = Human
Representational Space

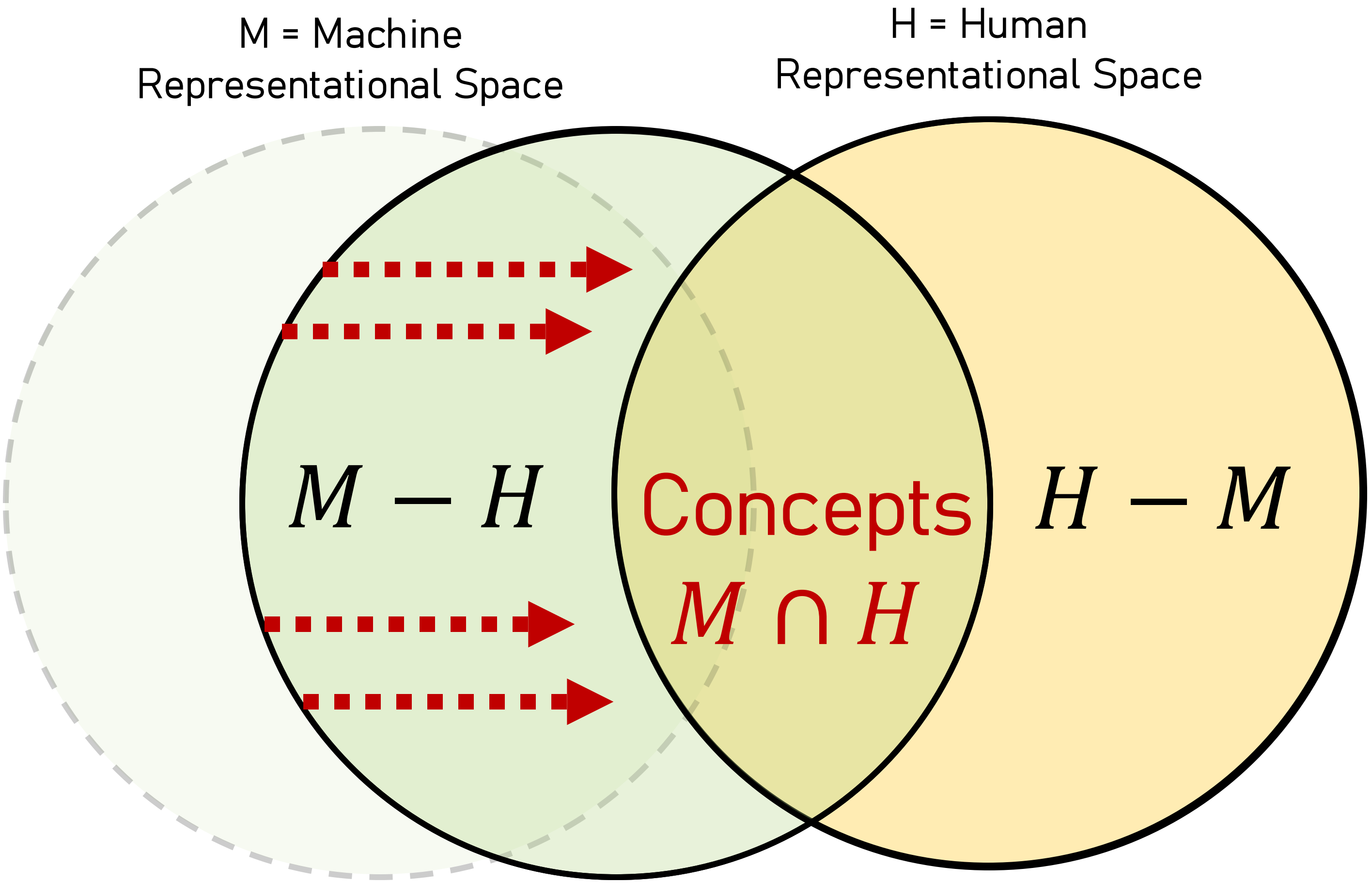
$M - H$

$H - M$

$M \cap H$ = Human-like Concepts

[1] Inspired by Schut et al. "Bridging the human-ai knowledge gap: Concept discovery and transfer in alphazero." arXiv (2023).

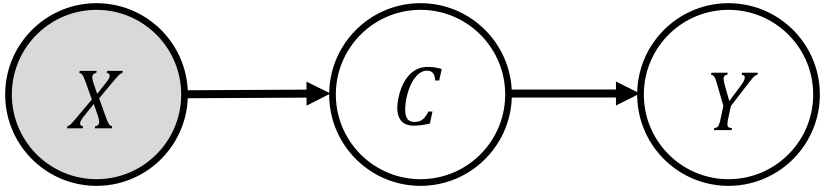
ALIGNING MACHINES AND HUMANS



[1] Inspired by Schut et al. "Bridging the human-ai knowledge gap: Concept discovery and transfer in alphazero." arXiv (2023).

CONCEPT-BASED REASONING

Concept-based reasoning can be framed as a Concept Bottleneck Model [1]



$$P(C, Y \mid X) = P(C \mid X) P(Y \mid C)$$

X = Sample Features

C = Human-interpretable "Concepts"

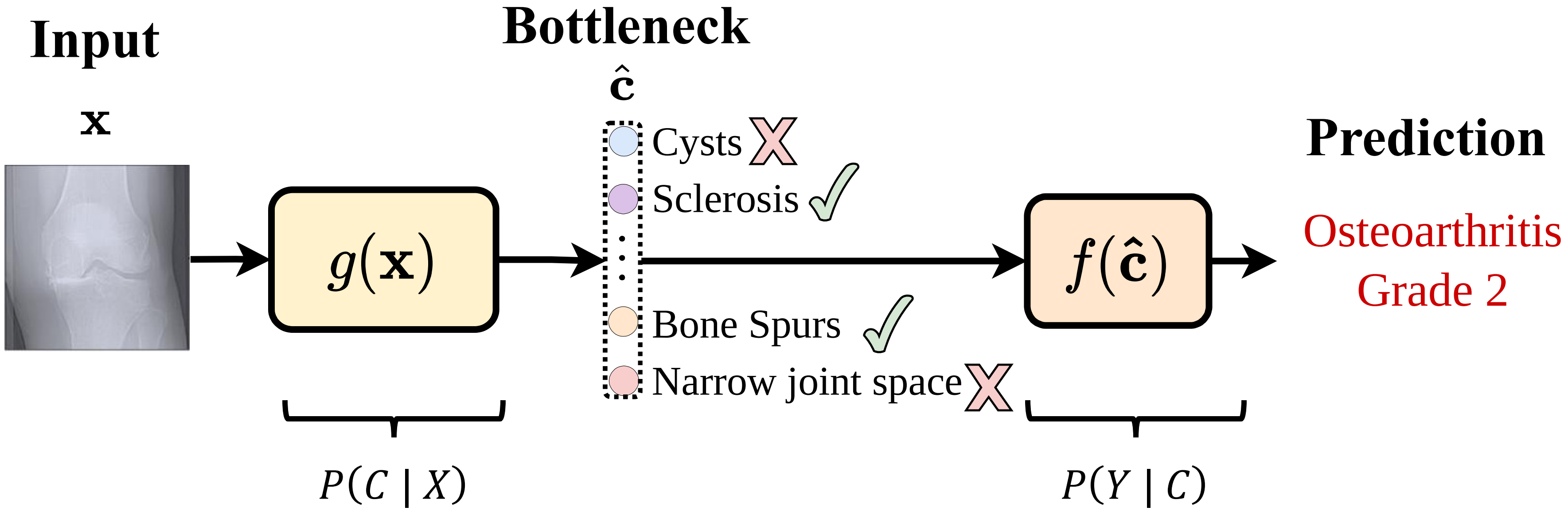
Y = Target Task Labels

[1] Koh et al. "Concept bottleneck models." International Conference on Machine Learning. PMLR, 2020.

CONCEPT BOTTLENECK MODELS (CBMS)

CBMs decompose a DNN into two functions:

1. A *concept encoder* $g(x) = \hat{c}$ predicting concepts from the input features
2. A *label predictor* $f(\hat{c}) = \hat{y}$ predicting task labels from the predicted concepts



[1] Koh et al. "Concept bottleneck models." International Conference on Machine Learning. PMLR, 2020.

TRAINING A CBM

Given a concept-annotated dataset $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$ we can train a CBM in three different forms:

(1) Independently

$$\begin{cases} \mathbb{E}_{(\boldsymbol{x},\,\boldsymbol{c})\sim\mathcal{D}}[\mathrm{BCE}(g(\boldsymbol{x}), \boldsymbol{c})] \\ \mathbb{E}_{(\boldsymbol{c},\,y)\sim\mathcal{D}}[\mathrm{CE}(f(\boldsymbol{c}), y)] \end{cases}$$

(2) Sequentially

(a) $\mathbb{E}_{(\boldsymbol{x},\,\boldsymbol{c})\sim\mathcal{D}}[\mathrm{BCE}(g(\boldsymbol{x}), \boldsymbol{c})]$

$\downarrow$ Freeze g

(b) $\mathbb{E}_{(\boldsymbol{x},\,y)\sim\mathcal{D}}\left[\mathrm{CE}\left(f\left(g(x)\right), y\right)\right]$

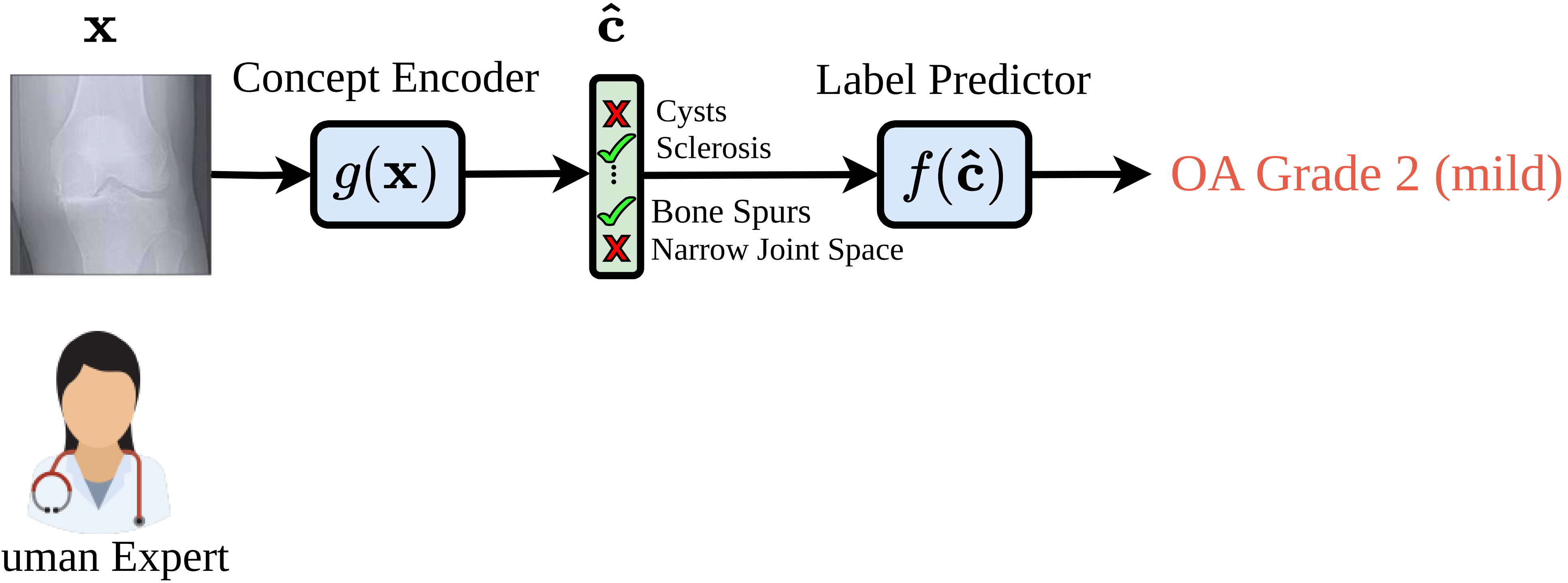
(3) Jointly

$$\mathbb{E}_{(\boldsymbol{x},\,\boldsymbol{c},\,y)\sim\mathcal{D}}\left[\mathrm{CE}\left(f\left(g(\boldsymbol{x})\right), y\right) + \lambda \cdot \mathrm{BCE}(g(\boldsymbol{x}), \boldsymbol{c})\right]$$

[1] Koh et al. "Concept bottleneck models." International Conference on Machine Learning. PMLR, 2020.

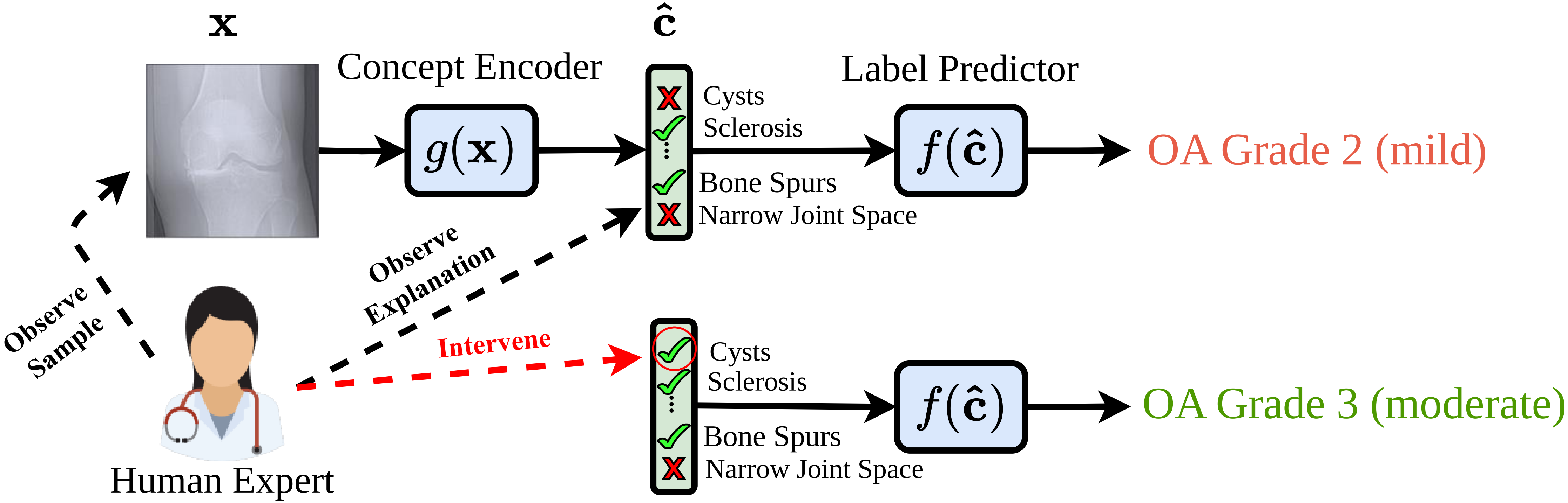
CONCEPT-LEVEL INTERVENTIONS

Concept-based reasoning enables powerful human-AI interactions



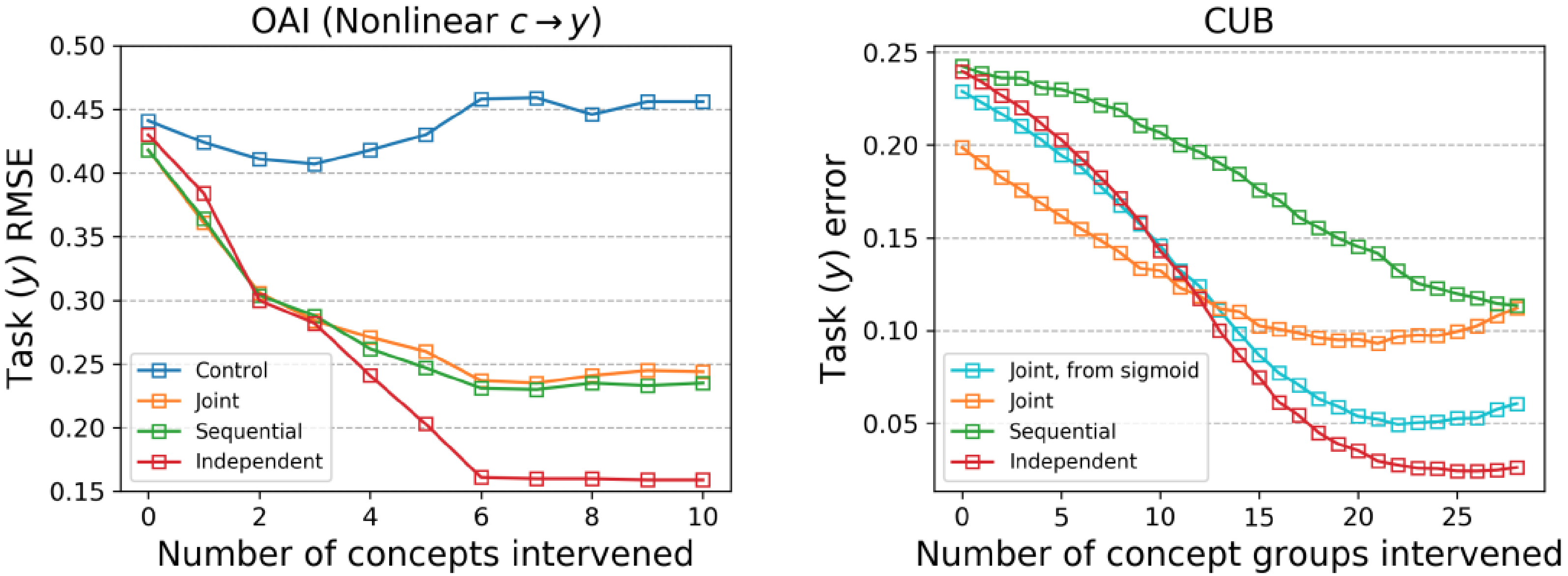
CONCEPT-LEVEL INTERVENTIONS

Concept-based reasoning enables powerful human-AI interactions



CONCEPT INTERVENTIONS

As we intervene on more concepts, CBM's test error goes down!



[1] Koh et al. "Concept bottleneck models." International Conference on Machine Learning. PMLR, 2020.

ARE CBMS ALL WE NEED?

CBMs are great in a lot of ways:

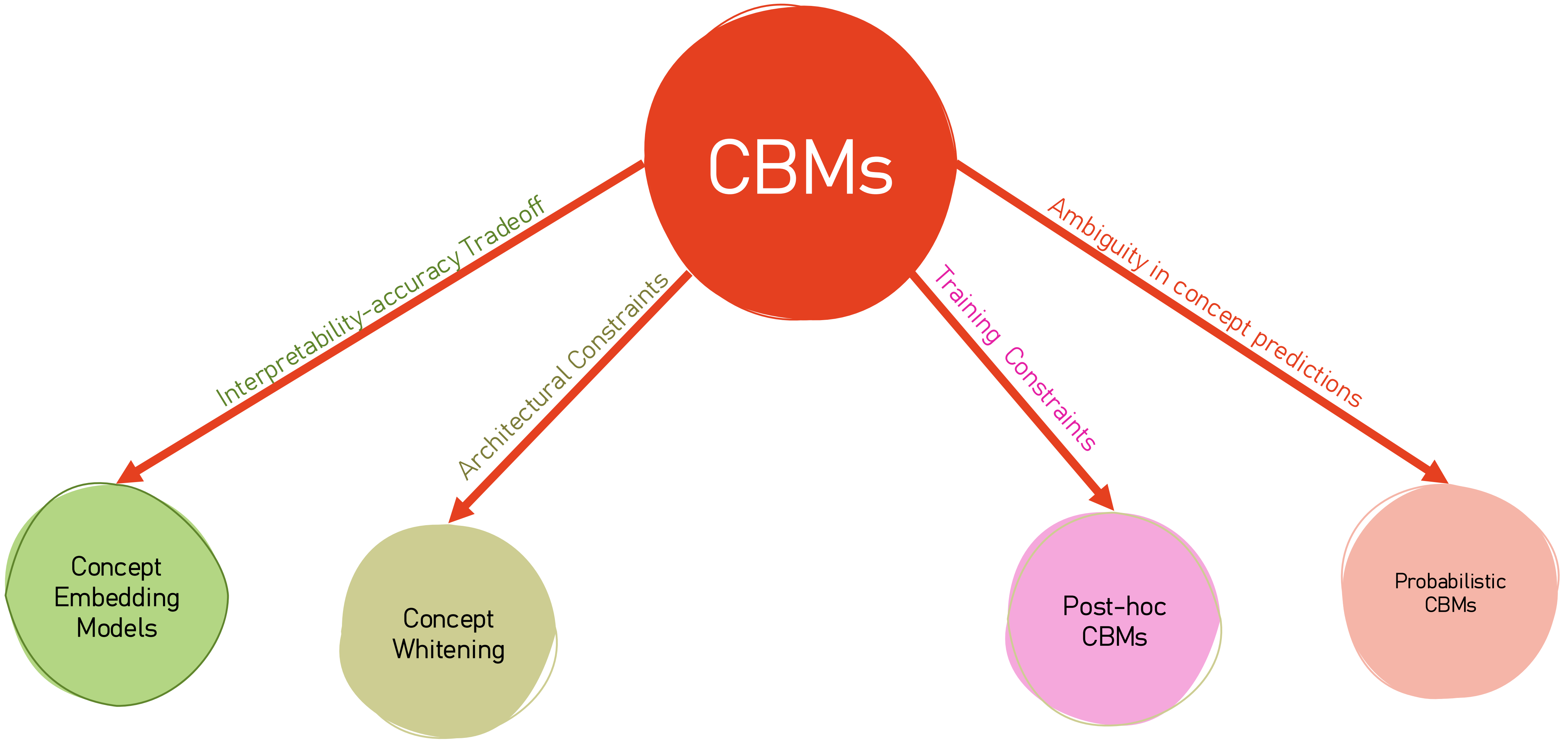
1. They are simple to understand and provide **high-level explanations**.
2. They enable **test-time interventions** that improve their accuracy.
3. They are very **stable**, expressive and **easy to train**.

So, are we done?

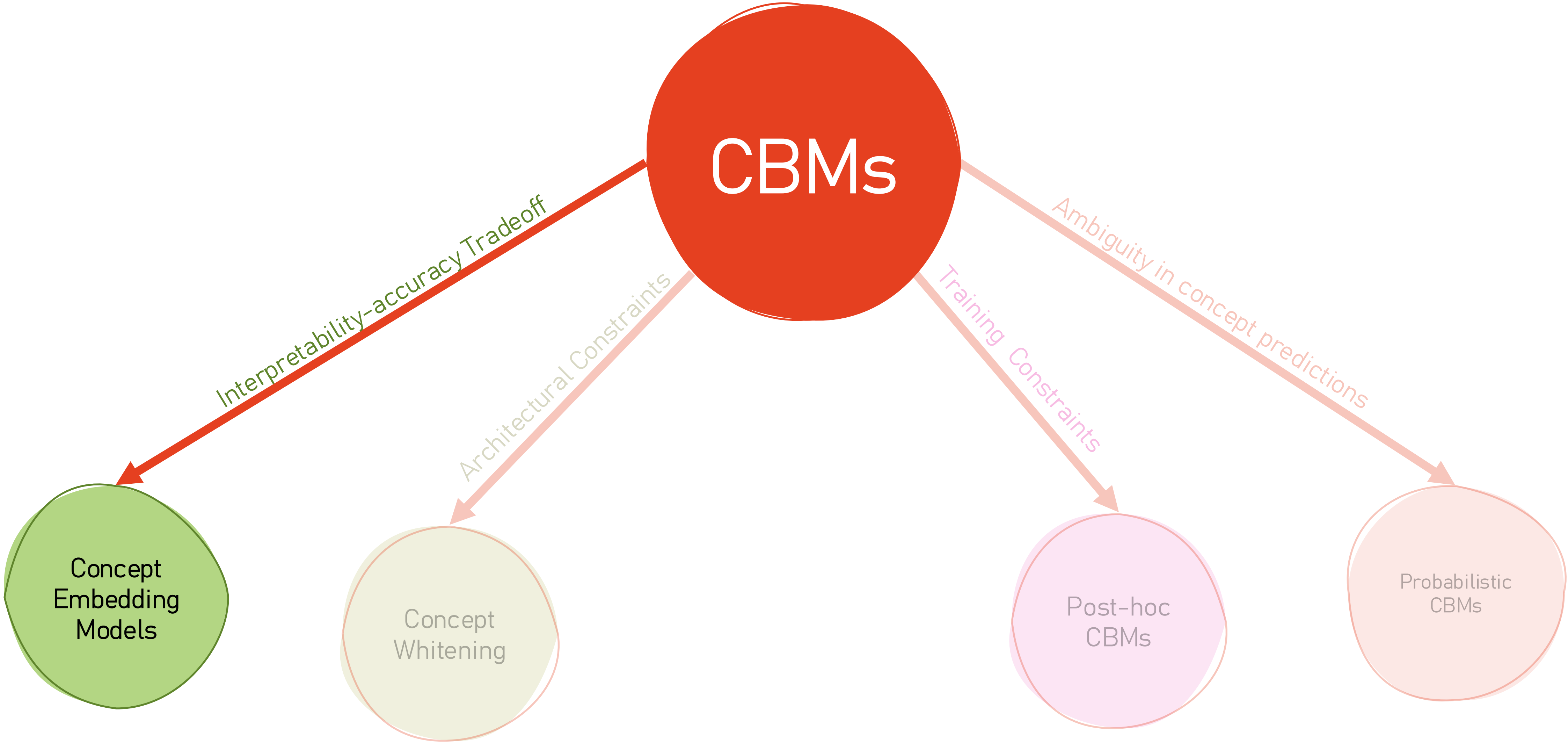
Short Answer: No

Long Answer:

INTRODUCING CBM'S FRIENDS



SPEED-DATING WITH CBM'S FRIENDS



CBMs

Interpretability-accuracy Tradeoff

Architectural Constraints

Training Constraints

Ambiguity in concept predictions

Concept Embedding Models

Concept Whitening

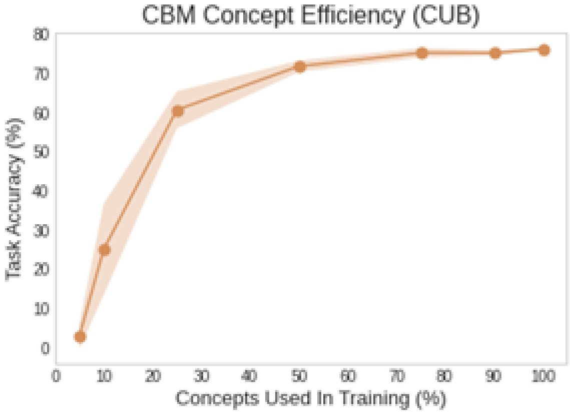
Post-hoc CBMs

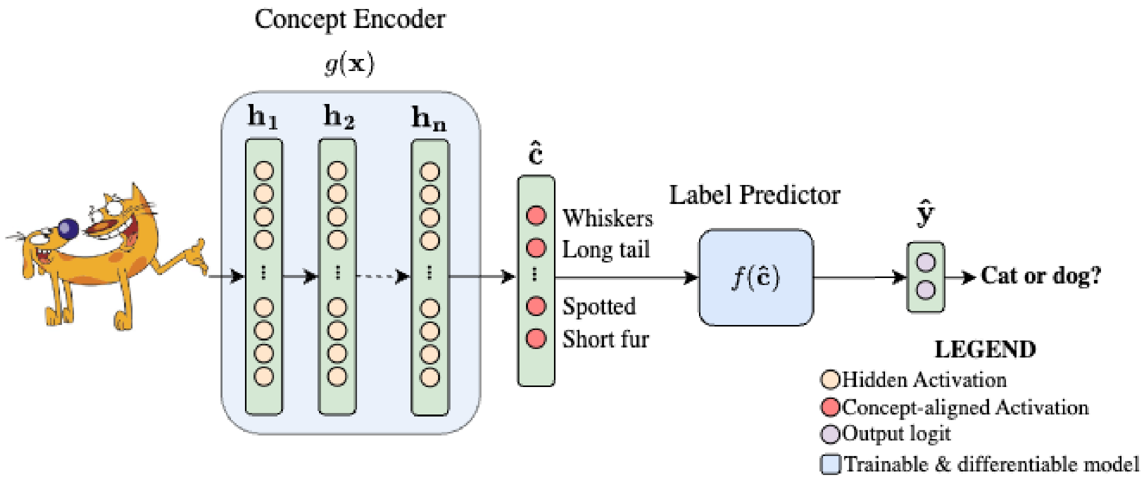
Probabilistic CBMs

CONCEPT EMBEDDING MODELS

Limitation Being Addressed

Provided concepts need to be "**complete**" or else we observe a **trade-off**!



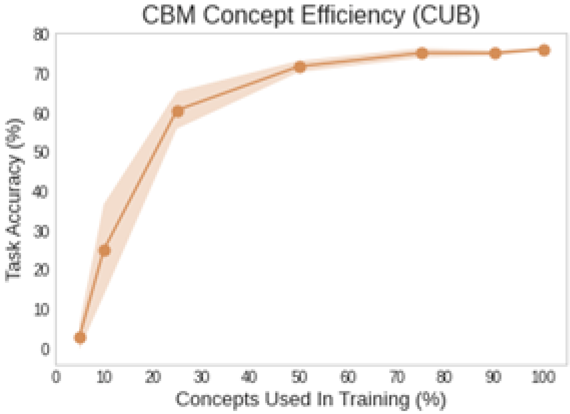


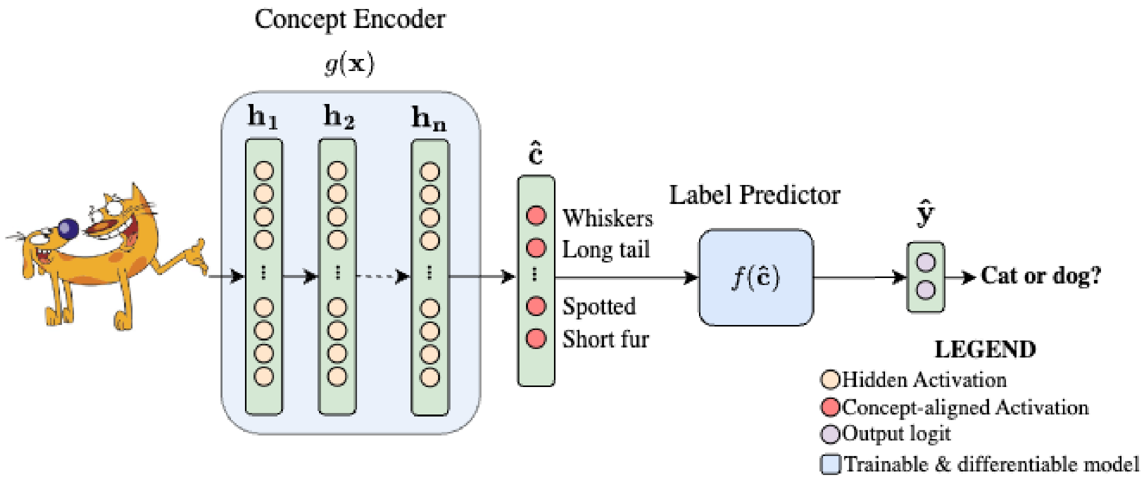
[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

Limitation Being Addressed

Provided concepts need to be “**complete**” or else we observe a **trade-off**!





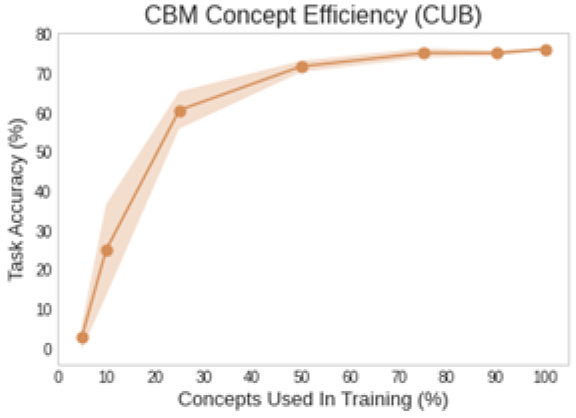
Why can’t we just add a bypass from the input to the output?

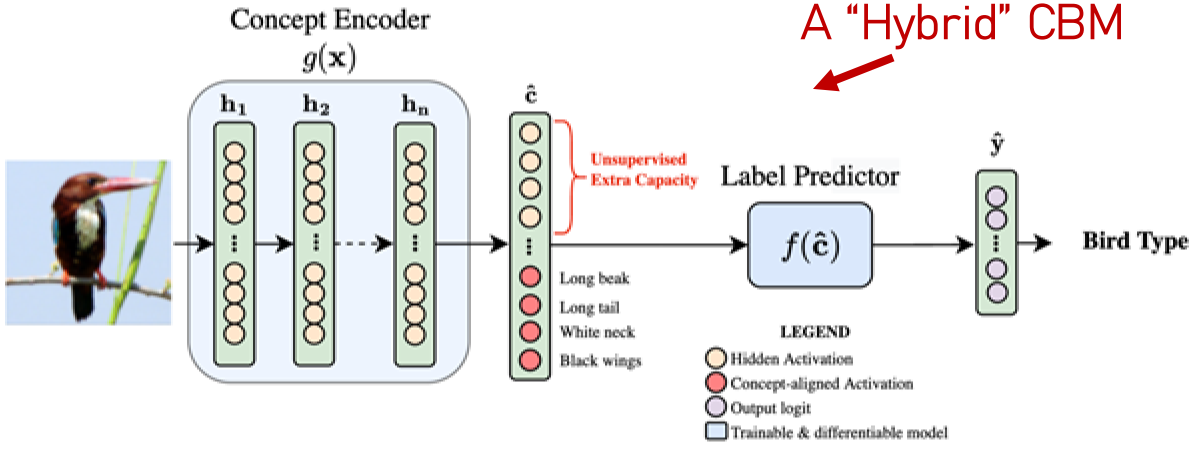
[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

Limitation Being Addressed

Provided concepts need to be "**complete**" or else we observe a **trade-off**!





A "Hybrid" CBM

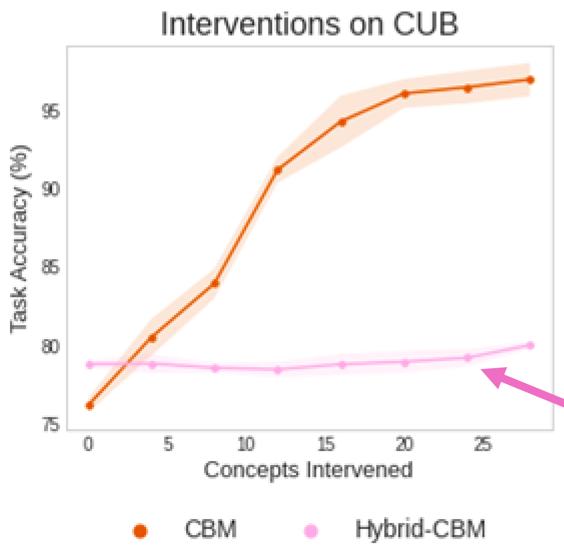
Why can't we just add a bypass from the input to the output?

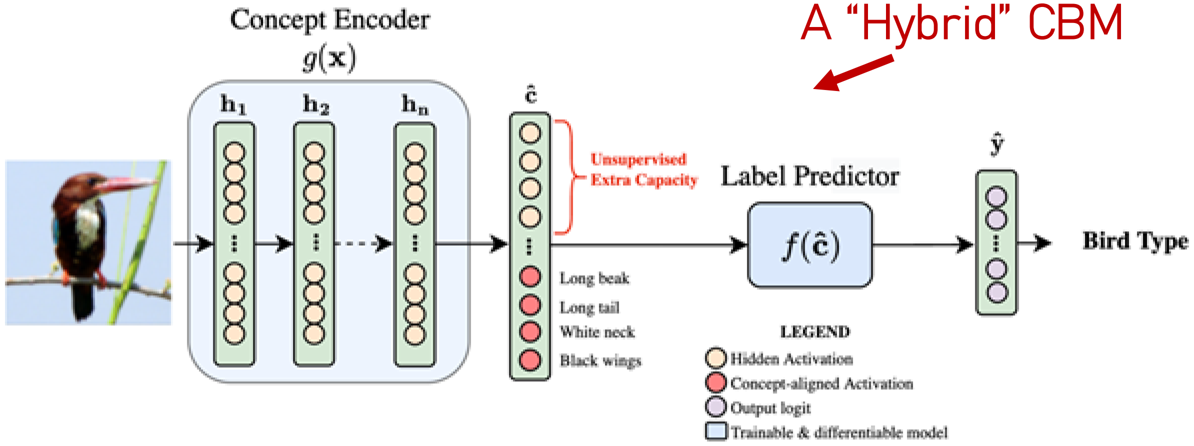
[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

Limitation Being Addressed

Provided concepts need to be "complete" or else we observe a trade-off!





A "Hybrid" CBM

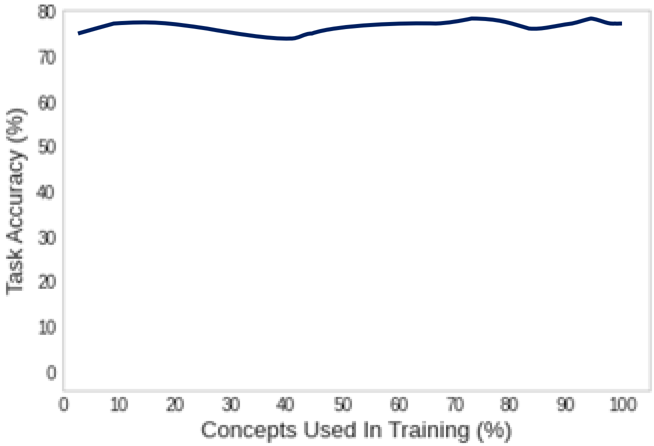
Interventions do not necessarily work with Hybrid CBMs!

[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

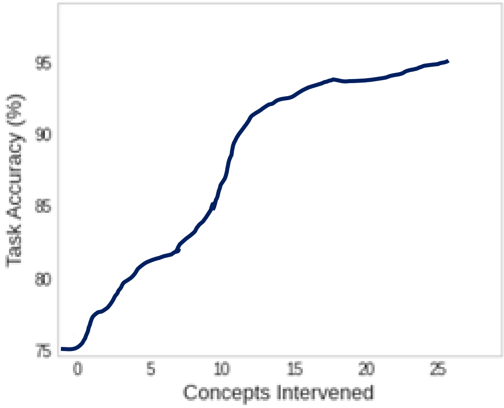
What we want

Similar performance regardless of the number of concepts used during training



"*Completeness Agnostic*"

Better performance as we intervene in more concepts



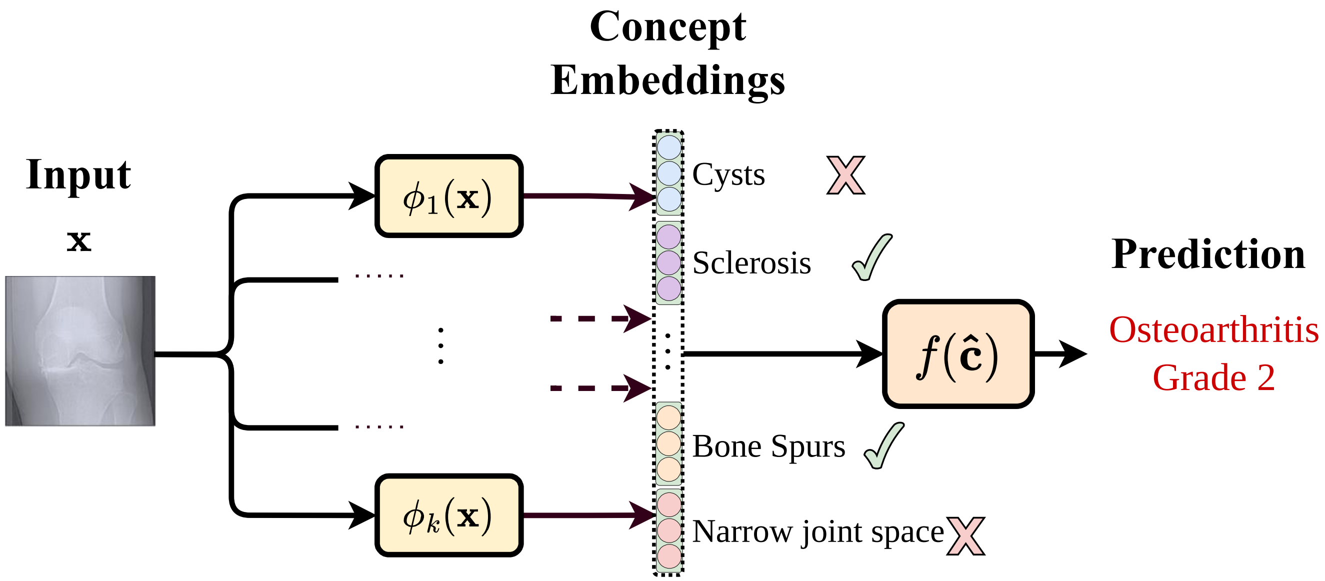
"*Intervenable*"

[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

Proposed Solution

We can achieve completeness agnosticism by extending the concept representations to higher-dimensions

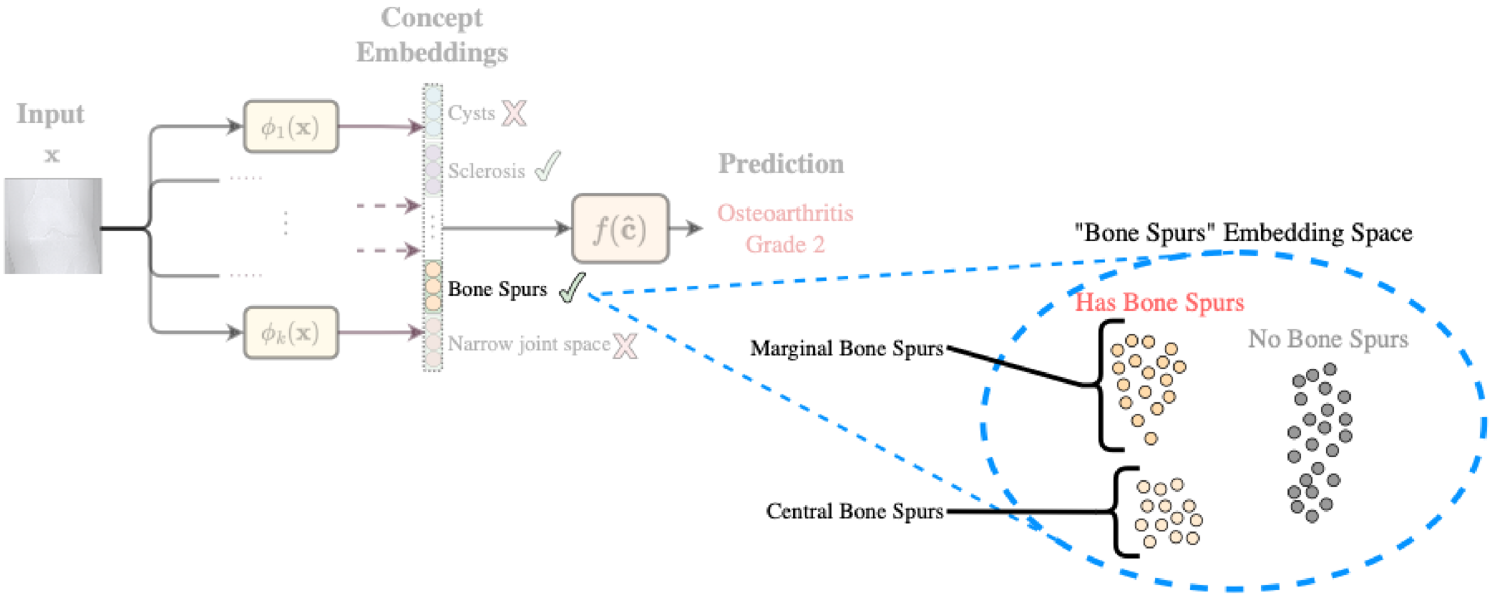


[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

Proposed Solution

We can achieve completeness agnosticism by extending the concept representations to higher-dimensions



[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

Proposed Solution

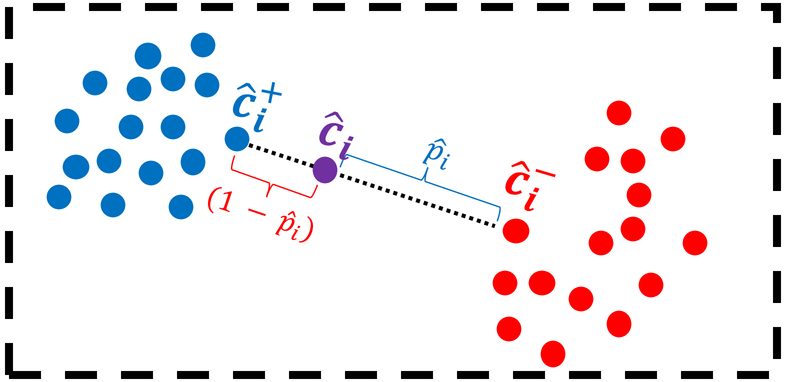
We can achieve intervenability by decomposing $\hat{\boldsymbol{c}}_{\boldsymbol{i}}$ as the mixture between two representations $\{\hat{\boldsymbol{c}}_{\boldsymbol{i}}^{+}, \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{-}\}$:

$$\boxed{\hat{\boldsymbol{c}}_{\boldsymbol{i}} \coloneqq \hat{p}_i \, \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{+} + (1 - \hat{p}_i) \, \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{-}}$$

🔵 “Positive” concept embeddings

🔴 “Negative” concept embeddings

Concept Embedding Space $\mathbb{R}^m$



[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

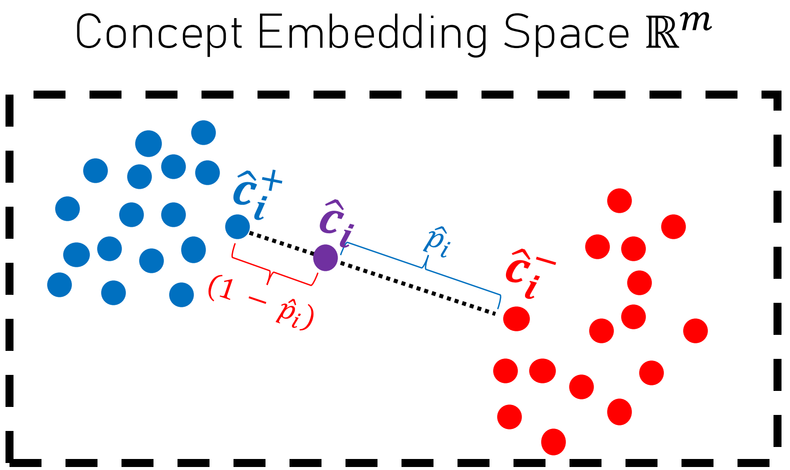
Proposed Solution

We can achieve intervenability by decomposing $\hat{\boldsymbol{c}}_{\boldsymbol{i}}$ as the mixture between two representations $\{\hat{\boldsymbol{c}}_{\boldsymbol{i}}^{+}, \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{-}\}$:

$$\boxed{\hat{\boldsymbol{c}}_{\boldsymbol{i}} \coloneqq \hat{p}_i \, \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{+} + (1 - \hat{p}_i) \, \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{-}}$$

🔵 “Positive” concept embeddings

🔴 “Negative” concept embeddings


Concept Embedding Space $\mathbb{R}^m$

Determining a concept’s activation given $\hat{\boldsymbol{c}}_{\boldsymbol{i}}$ then comes down to determining whether $\hat{\boldsymbol{c}}_{\boldsymbol{i}}$ comes from $P(\hat{\boldsymbol{c}}_{\boldsymbol{i}}^{+} \,|\, x)$ or $P(\hat{\boldsymbol{c}}_{\boldsymbol{i}}^{-} \,|\, x)$

[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

Proposed Solution

You can then intervene on a concept by fixing its representation to the embedding corresponding to the ground-truth concept label:

$$\hat{\boldsymbol{c}}_{\boldsymbol{i}} := \begin{cases} \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{+} \text{ if } c_i = 1 \\ \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{-}, \text{otherwise} \end{cases}$$

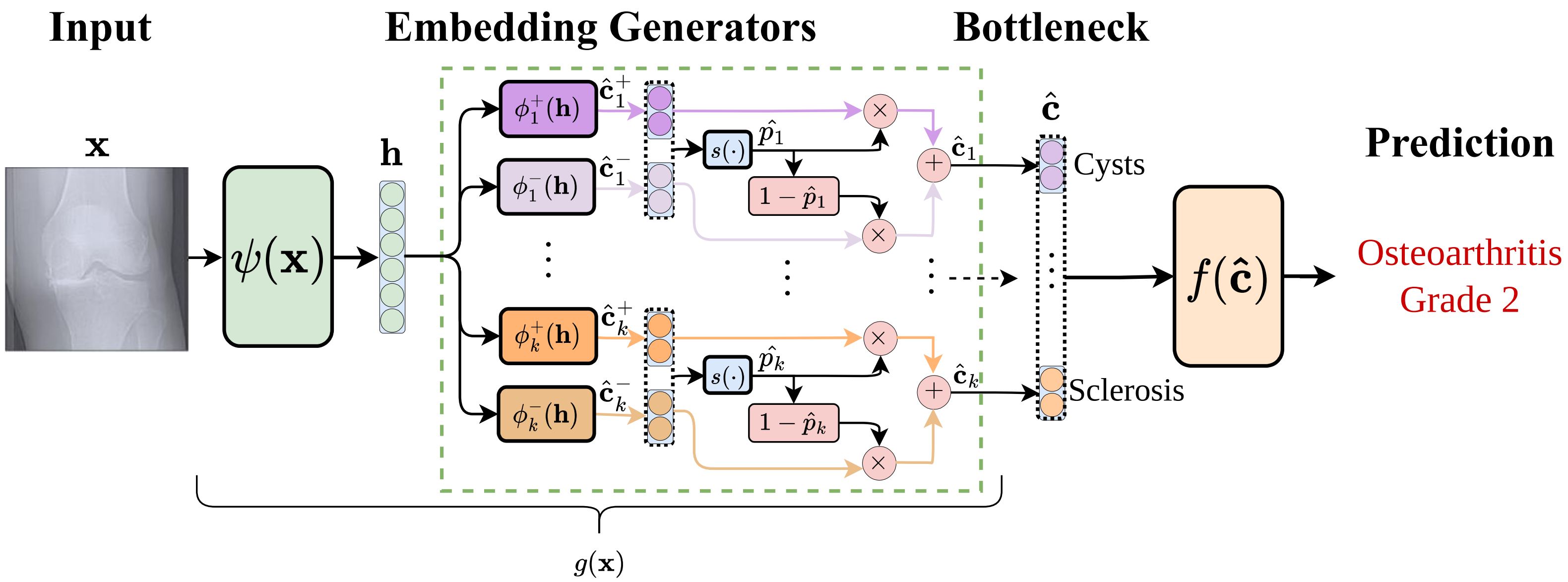
We can randomly do these interventions at training time to learn more useful representations!

[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

Proposed Solution

Learn two functions $(\phi_i^+,\ \phi_i^-)$ mapping $\boldsymbol{x}$ to a *positive* $\hat{\boldsymbol{c}}_i^+$ and a *negative* embedding $\hat{\boldsymbol{c}}_i^-$



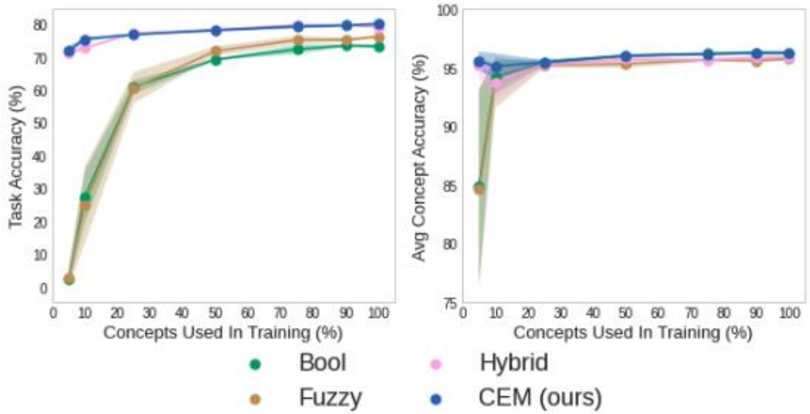
$$P(c_i = 1 \mid \boldsymbol{x}) = s(\hat{\boldsymbol{c}}_i^+, \hat{\boldsymbol{c}}_i^-) = \sigma(S\,[\hat{\boldsymbol{c}}_i^+, \hat{\boldsymbol{c}}_i^-]^T\,)$$

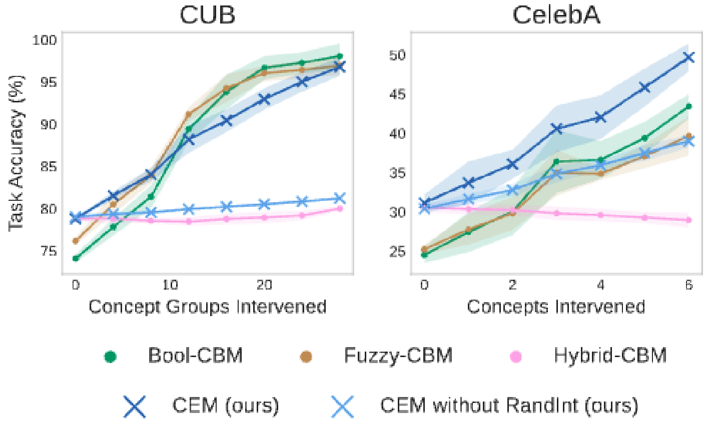
[1] Espinosa Zarlenga, Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

Proposed Solution

This gives you models that are completeness agnostic and intervenable*



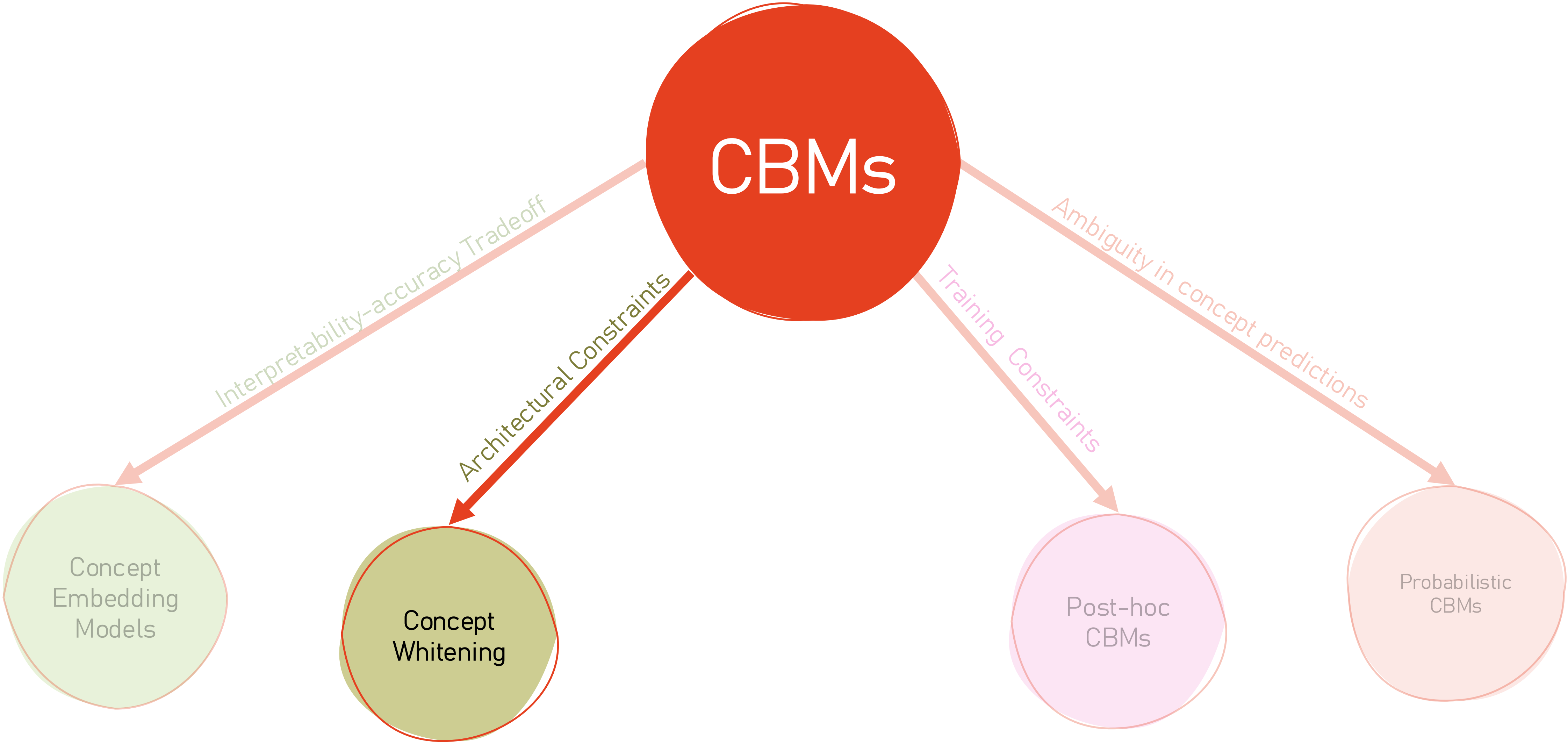


RandInt is important!

*This is particularly true when, during training, you randomly intervene** on a concept with probability p_{int}.

** These sorts of training-time interventions are useful here only because by using embeddings, we can backpropagate gradients to the concept encoder even when a concept is intervened on (a CBM wouldn't).

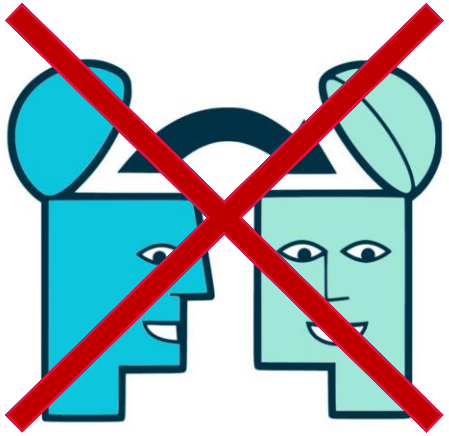
SPEED-DATING WITH CBM'S FRIENDS



CONCEPT WHITENING (CW)

Limitation Being Addressed

Training a CBM has impractical architectural constraints and requires all training samples to be concept annotated!



This limits our ability to exploit powerful pre-trained models

[1] Chen et al. "Concept whitening for interpretable image recognition." Nature Machine Intelligence 2.12 (2020).

INTERPRETABLE-BY-DESIGN NEURAL LAYER

Proposed Solution

Design an interpretable-by-design layer which we can use to replace an equivalent component in a pre-trained model and quickly fine-tune it to make it interpretable

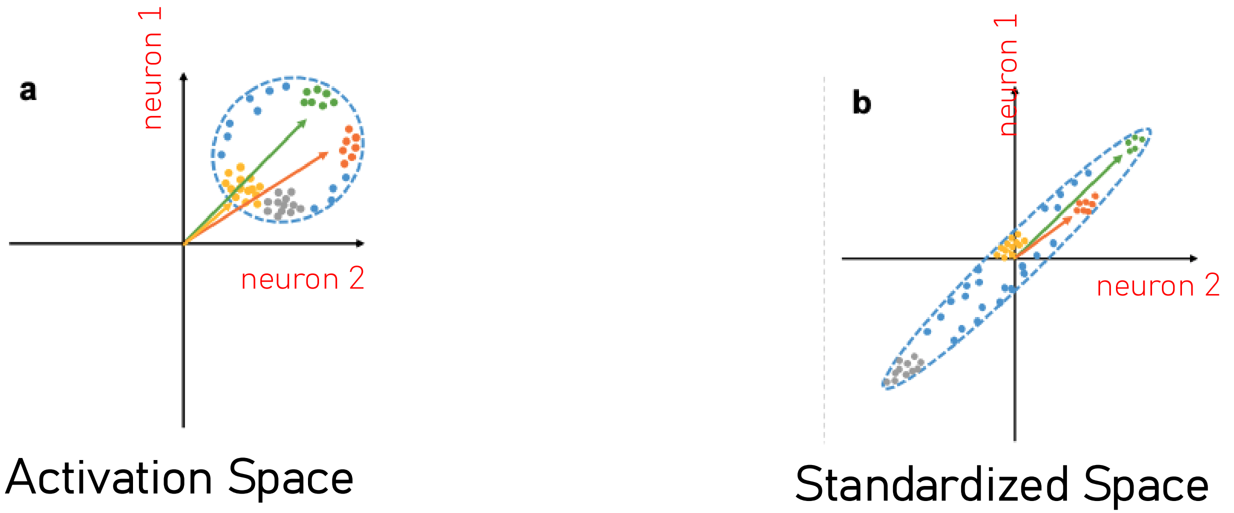
We will target the commonly used Batch Normalization (BN) layer

[1] Chen et al. "Concept whitening for interpretable image recognition." Nature Machine Intelligence 2.12 (2020).

WHITENING FOR DISENTANGLING CONCEPTS

Intuition

Normalization can somewhat help disentangle concepts in a DNN’s latent space



Activation Space

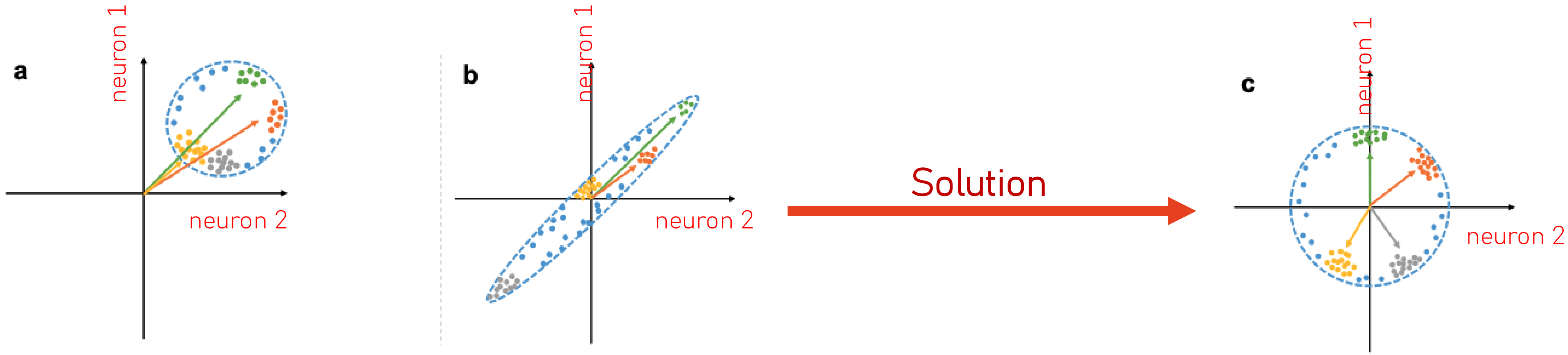
Standardized Space

[1] Chen et al. "Concept whitening for interpretable image recognition." Nature Machine Intelligence 2.12 (2020).

WHITENING FOR DISENTANGLING CONCEPTS

Intuition

Whitening a latent space can allow us to properly separate concepts in the latent space



Activation Space

Standardized Space

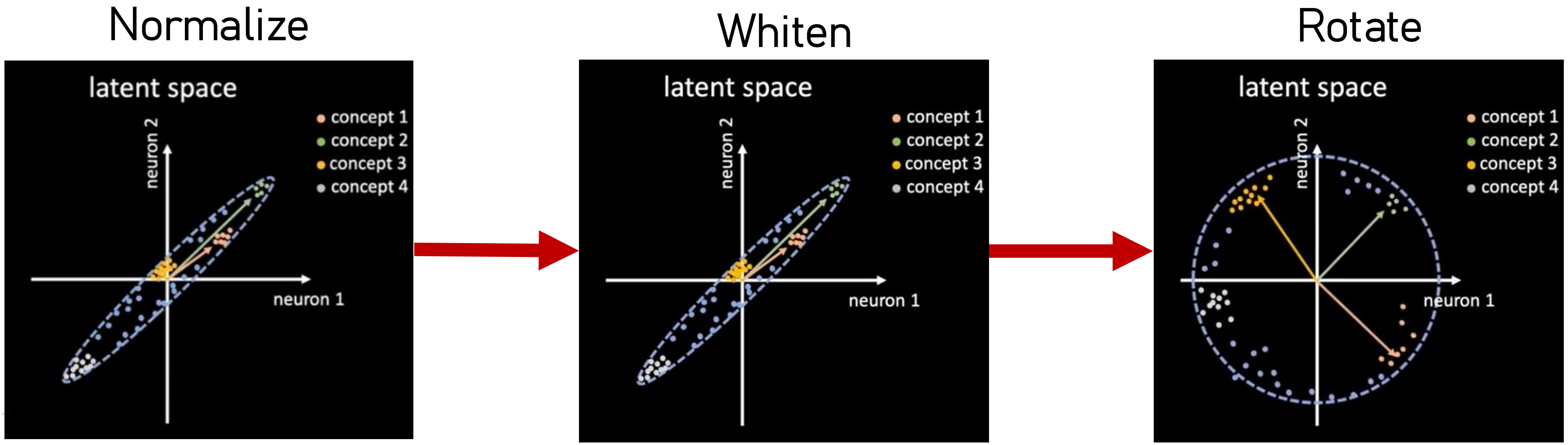
Whitened Space

[1] Chen et al. "Concept whitening for interpretable image recognition." Nature Machine Intelligence 2.12 (2020).

ROTATING TO ENSURE CONCEPT ALIGNMENT

Approach

More importantly, once an input is whitened, we can apply a rotation to align a specific concept to a specific axis!



Normalize

Whiten

Rotate

[1] Chen et al. "Concept whitening for interpretable image recognition." Nature Machine Intelligence 2.12 (2020).

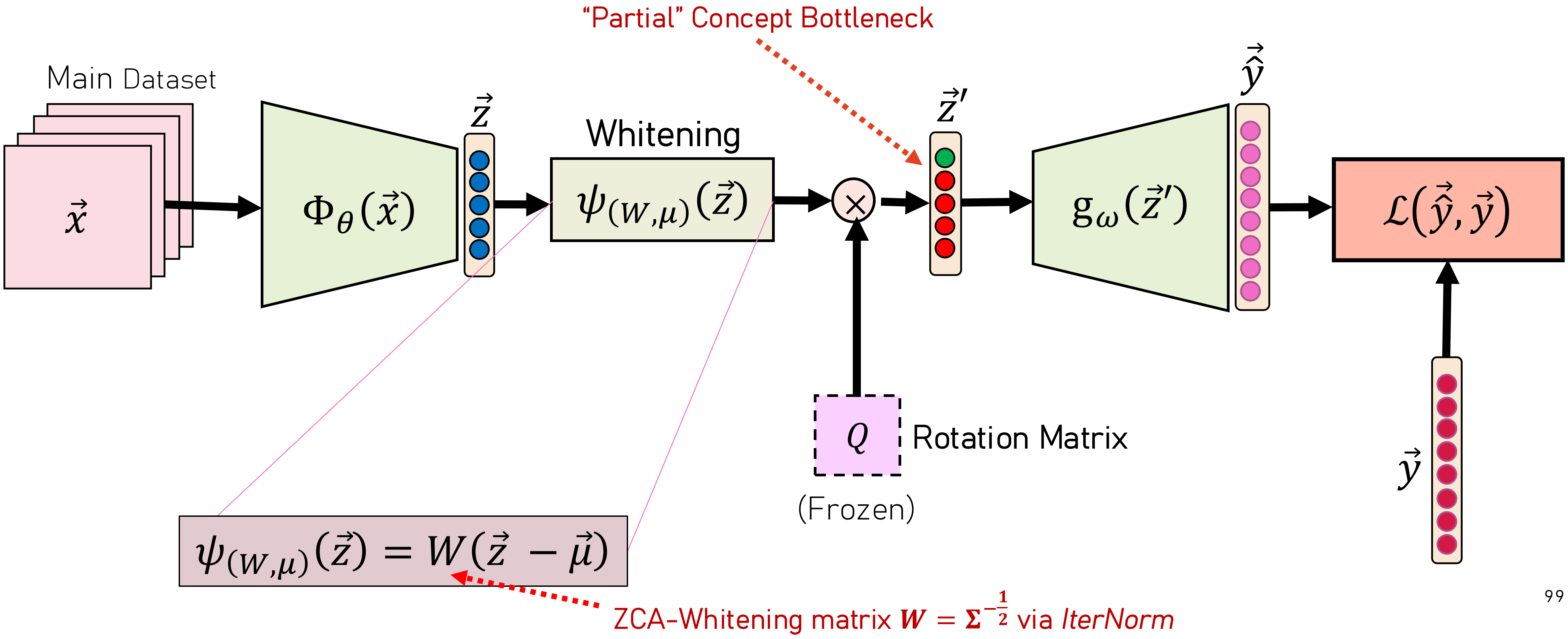
CONCEPT WHITENING (CW)

Approach

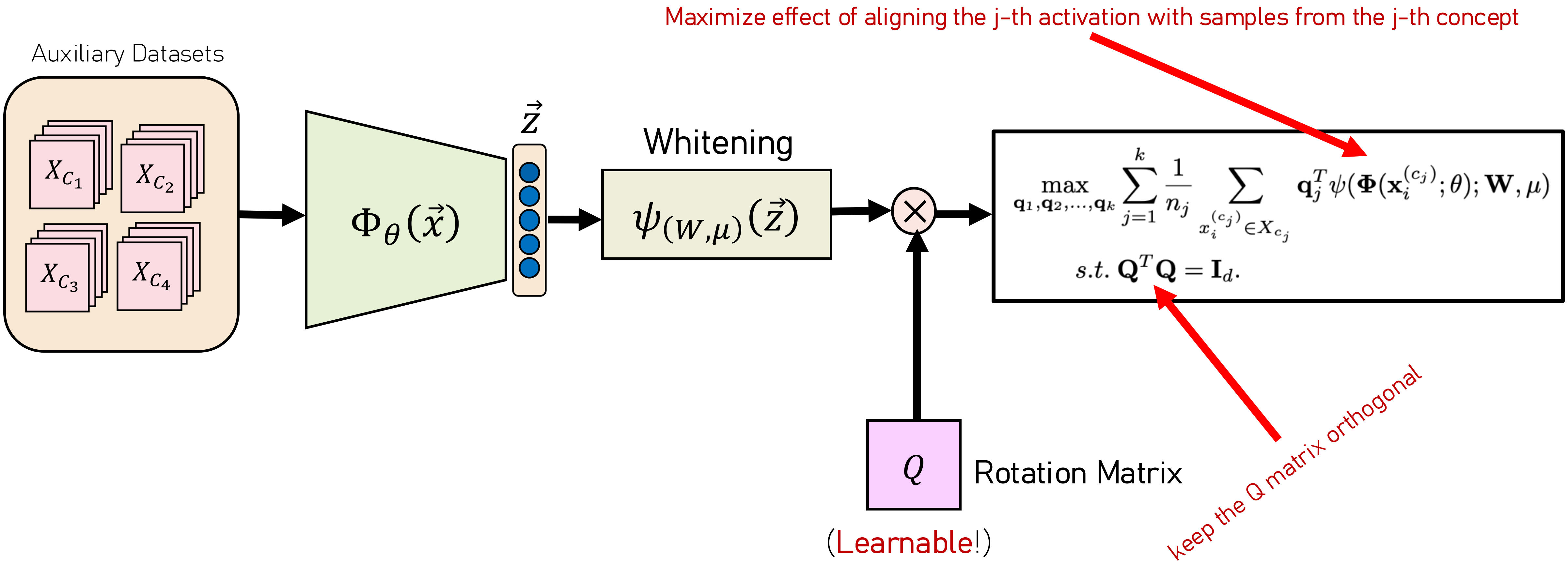
Given a fine-tuning training set $\mathcal{D}_t = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_i$ and $\boldsymbol{k}$ concept sets $\mathcal{D}_c = \{X_{C_1}, X_{C_2}, \cdots, X_{C_k}\}$, we will learn a rotation matrix $Q \in \mathbb{R}^{m \times m}$ by iterating between:

1. Task training step → make sure the downstream task prediction is accurate
2. Concept alignment step → make sure each concept is aligned to a latent activation

TRAINING CW: TASK TRAINING STEP



TRAINING CW: CONCEPT ALIGNMENT STEP



Maximize effect of aligning the j-th activation with samples from the j-th concept

$$\max_{\mathbf{q}_1,\mathbf{q}_2,\ldots,\mathbf{q}_k} \sum_{j=1}^{k} \frac{1}{n_j} \sum_{x_i^{(c_j)} \in X_{c_j}} \mathbf{q}_j^T \psi(\mathbf{\Phi}(\mathbf{x}_i^{(c_j)};\theta);\mathbf{W},\mu)$$

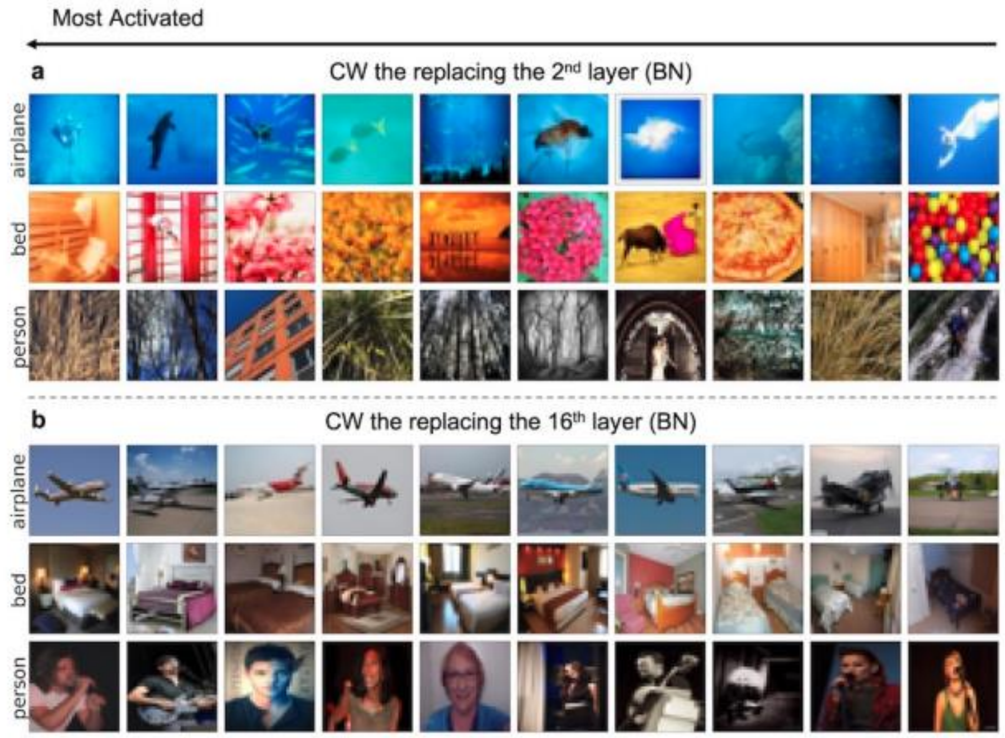
$$s.t.\ \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_d.$$

keep the Q matrix orthogonal

(Learnable!)

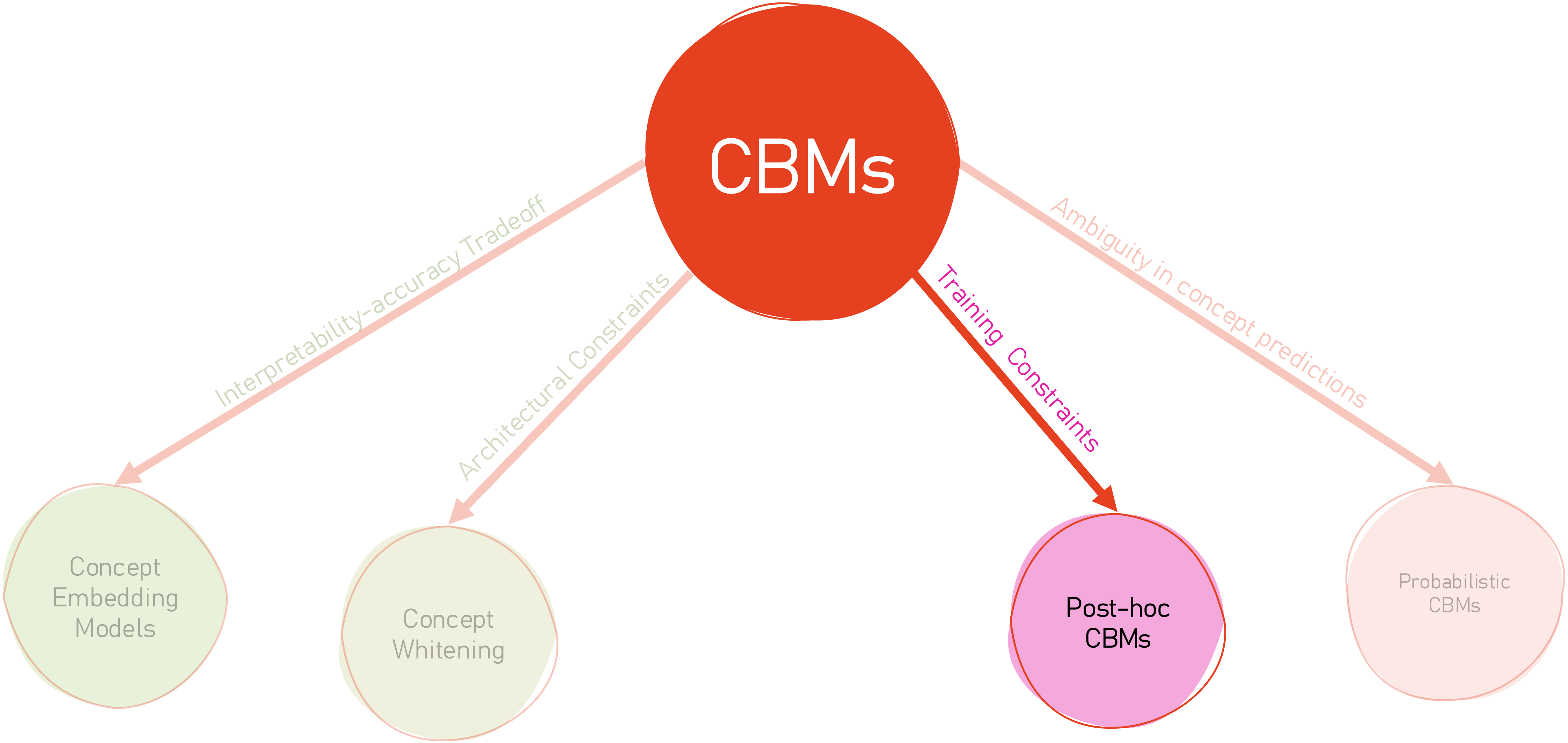
FOUND PROTOTYPES

CW supports the hypothesis that DNNs learn more complex concepts in later layers:



[1] Chen et al. "Concept whitening for interpretable image recognition." Nature Machine Intelligence 2.12 (2020).

SPEED-DATING WITH CBM'S FRIENDS



POST-HOC CBMS

Limitation Being Addressed

CW is great but is has some key limitations:

1. It requires a batch norm layer in the pretrained model (read: architecture-specific)
2. It requires fine-tuning of all the of the model's weights (read: could be expensive)

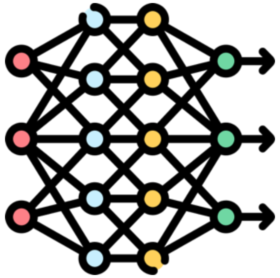
[1] Yuksekgonul et al. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

Proposed Solution

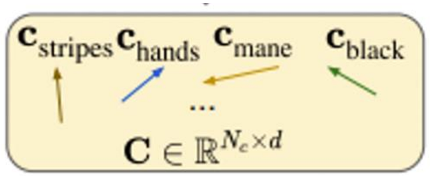
Given a bank of concept activation vectors in a pre-trained model, we **learn an interpretable mapping projected concept scores** and **a downstream task** of interest

*A **trained** DNN*



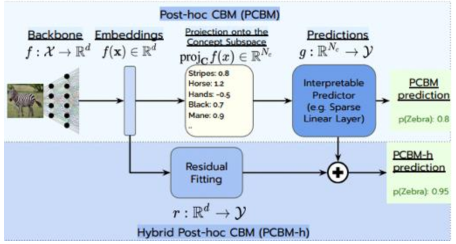
+

Concept Activation Vector Bank



=

Post-hoc CBMs!

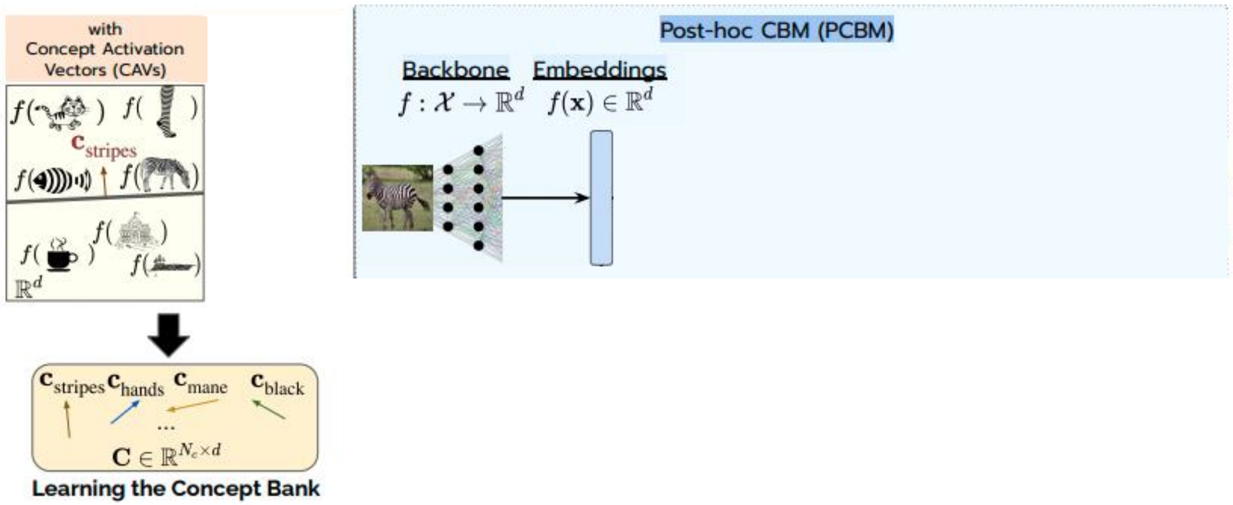


[1] Yuksekgonul et al. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

Proposed Solution

Goal: learn an interpretable model mapping concept similarity scores to task labels



Make the final prediction with an interpretable predictor

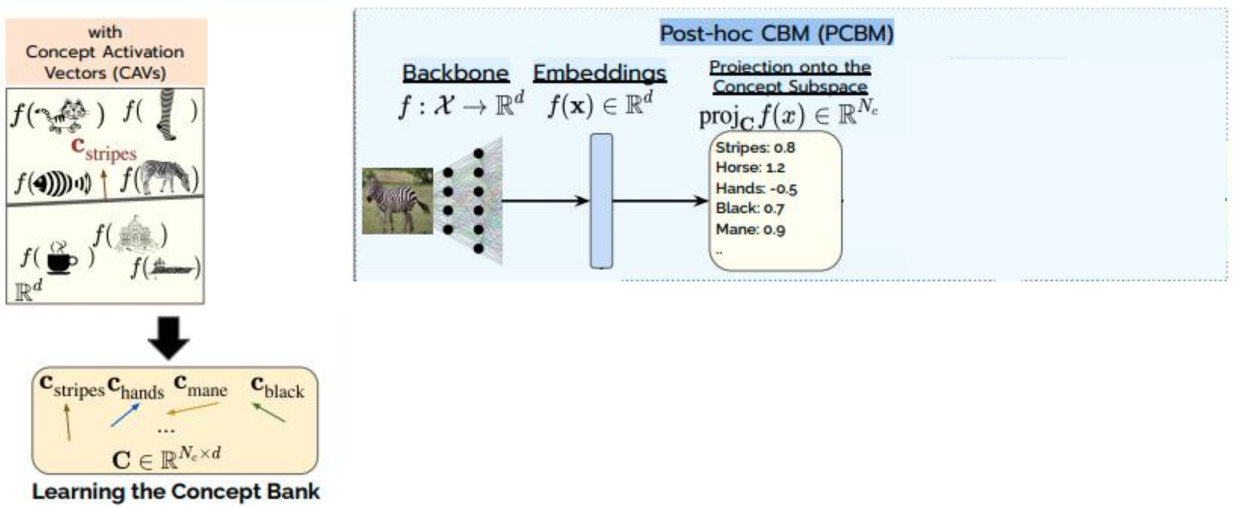
Step 1: learn CAVs from the frozen latent space of the pre-trained DNN

[1] Yuksekgonul et al. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

Proposed Solution

Goal: learn an interpretable model mapping concept similarity scores to task labels



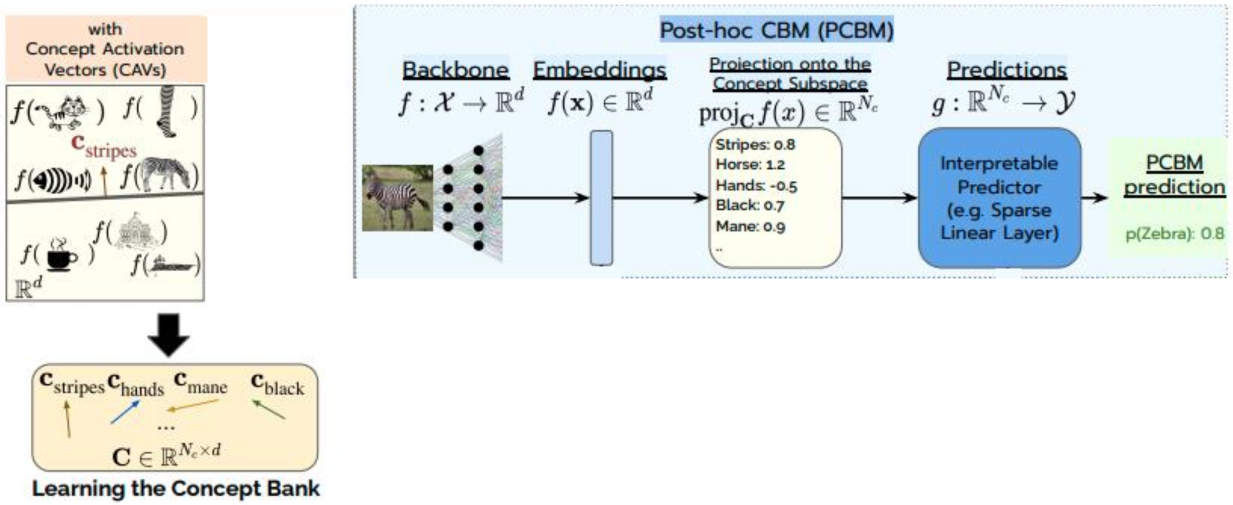
Step 2: project all training samples to the concept activation space using the cavs

[1] Yuksekgonul et al. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

Proposed Solution

Goal: learn an interpretable model mapping concept similarity scores to task labels



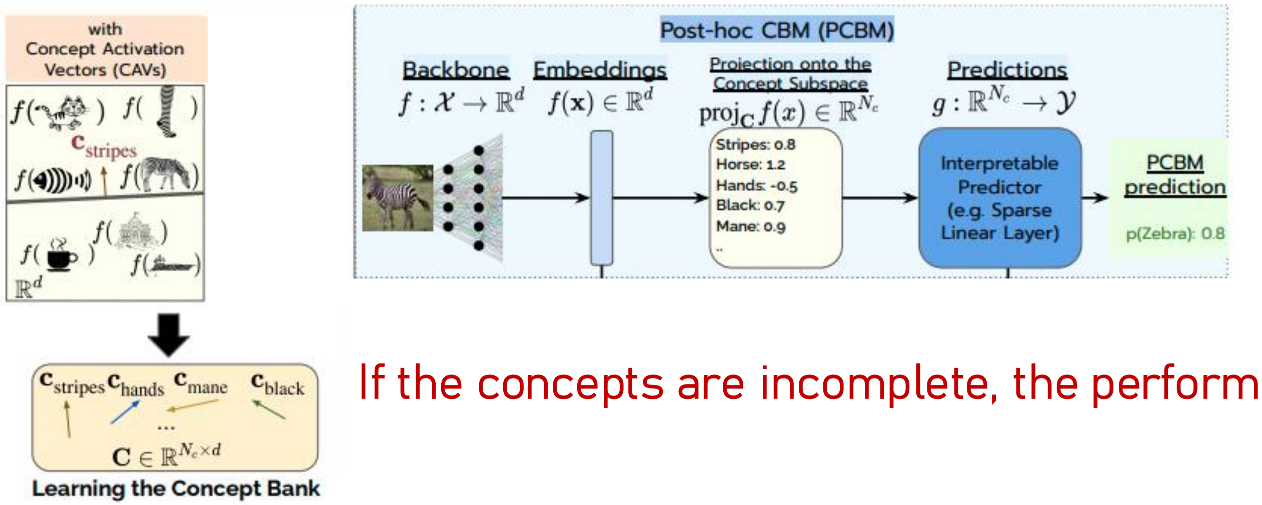
Step 3: learn an interpretable predictor from the concept scores to the task labels

[1] Yuksekgonul et al. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

Proposed Solution

Goal: learn an interpretable model mapping concept similarity scores to task labels



If the concepts are incomplete, the performance will drop significantly!

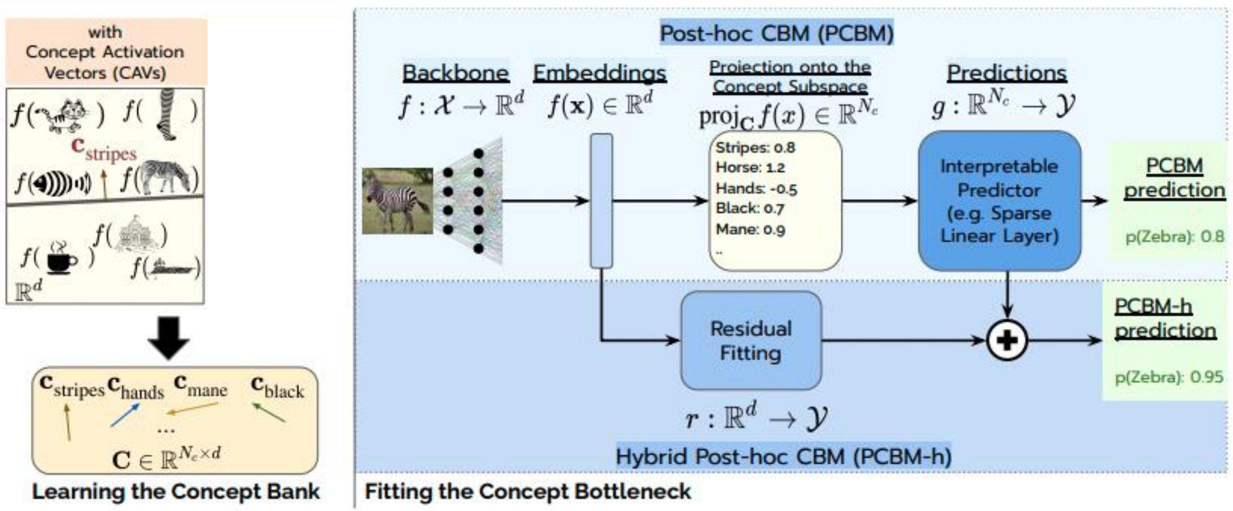
Step 3: learn an interpretable predictor from the concept scores to the task labels

[1] Yuksekgonul et al. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

Proposed Solution

Goal: learn an interpretable model mapping concept similarity scores to task labels

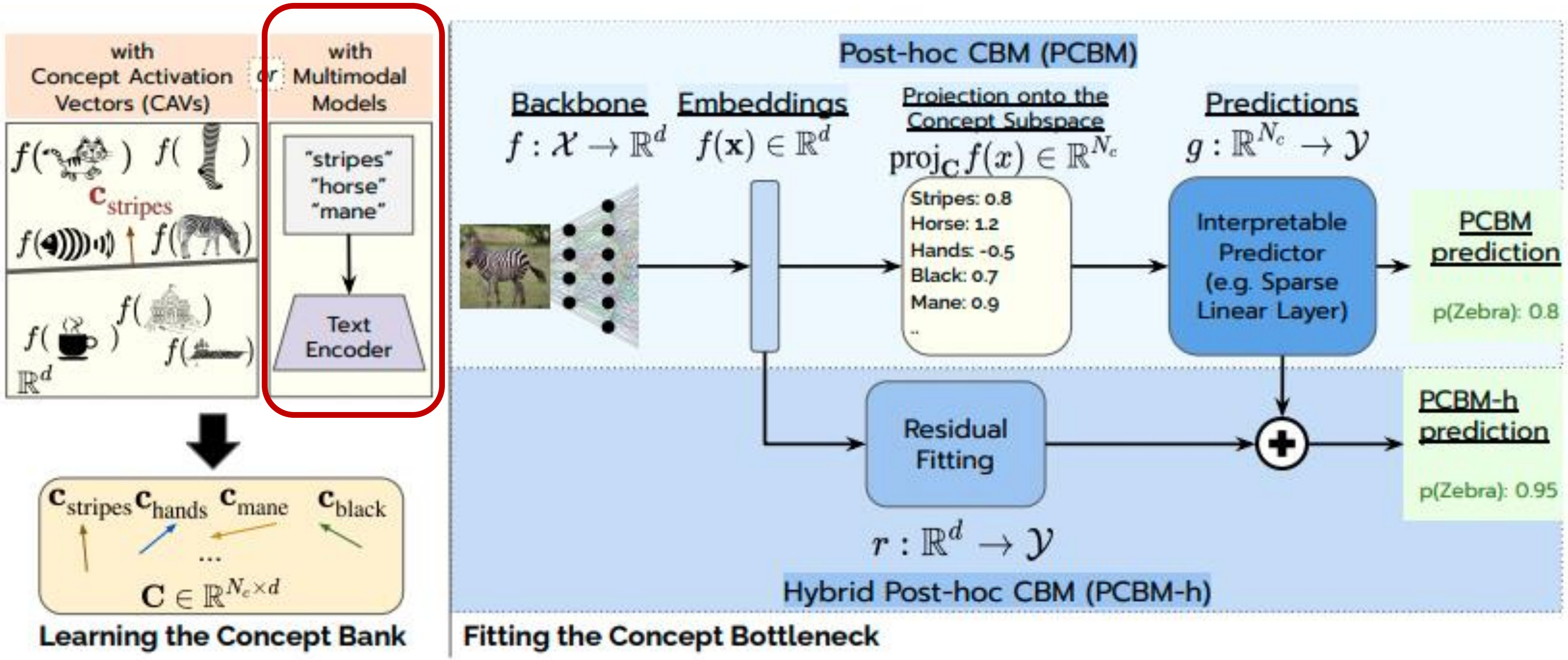


Step 4 (optional): fit a residual model if the concepts are incomplete

[1] Yuksekgonul et al. "Post-hoc concept bottleneck models." ICLR (2023).

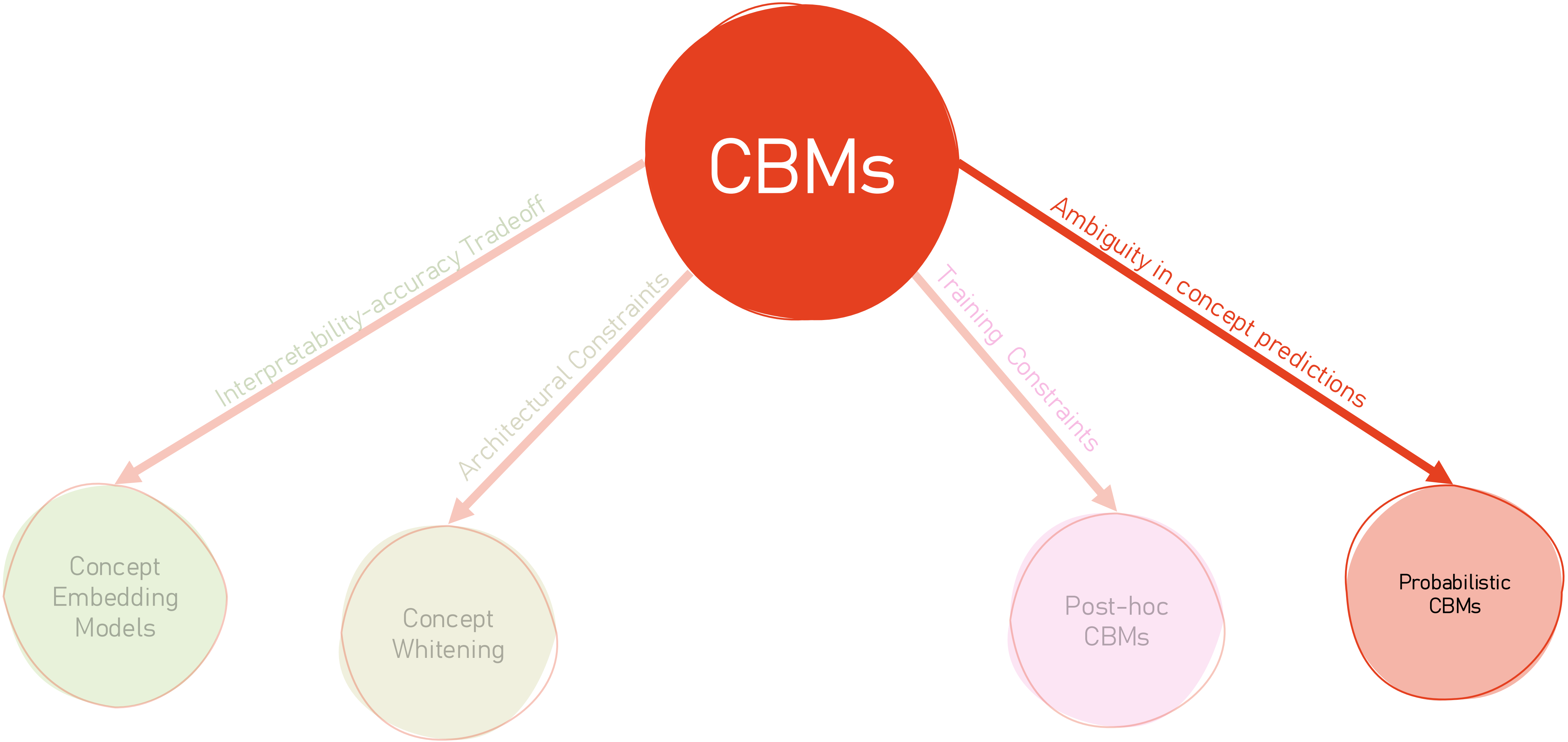
POST-HOC CBMS WITHOUT CONCEPT SETS

Post-hoc CBMs can be learnt **without concept sets** if we have access to **language-based concepts** together with a **multimodal model**



[1] Yuksekgonul et al. "Post-hoc concept bottleneck models." ICLR (2023).

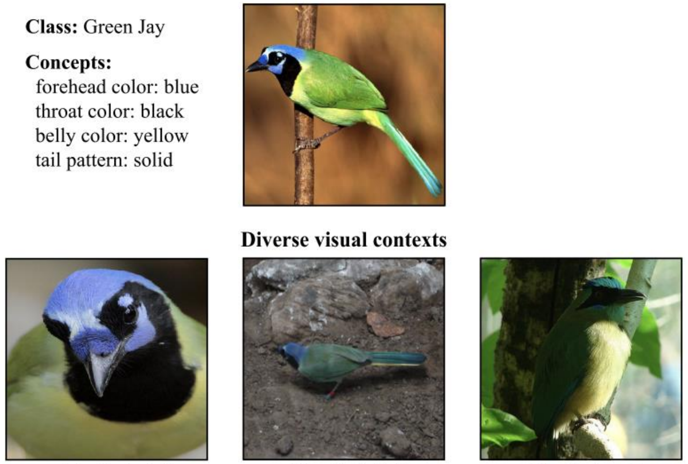
SPEED-DATING WITH CBM'S FRIENDS



PROBABILISTIC CBMS

Limitation being addressed

CBMs must predict concepts for all samples even they are ambiguous



ambiguity in tail ambiguity in belly ambiguity in color

The cross-entropy loss does not encourage the concept predictor to be uncertain

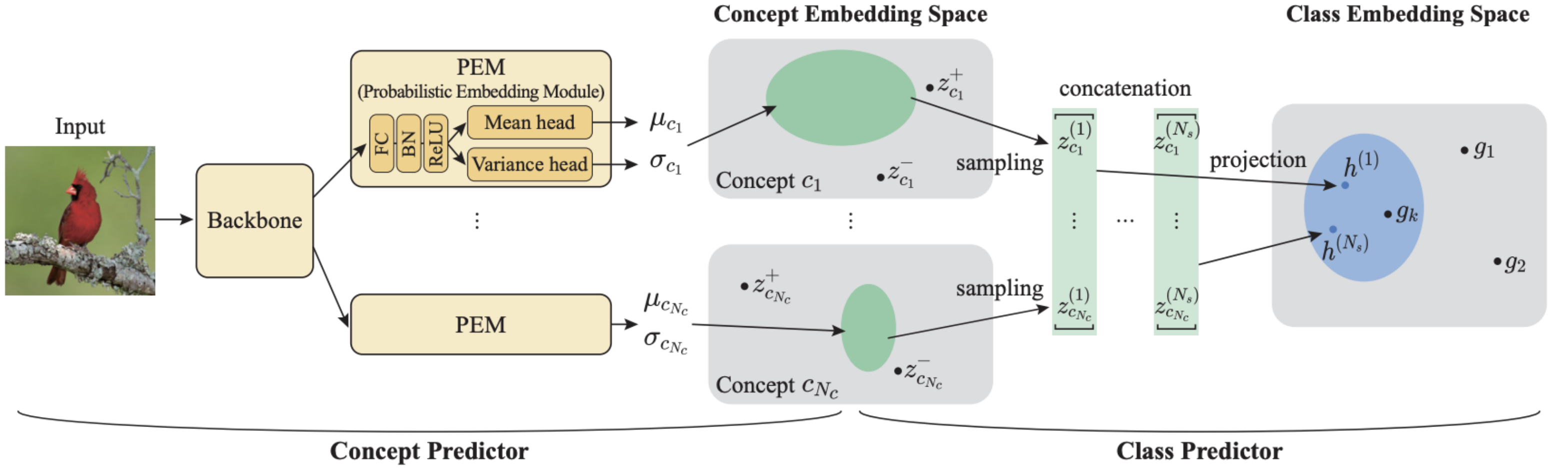
[1] Kim et al. "Probabilistic Concept Bottleneck Models." ICML (2023).

PROBABILISTIC CBMS

Proposed Solution

Use probabilistic embeddings that enable uncertainty estimation of each concept!

Learn a distribution over concept embeddings and use its variance to estimate uncertainty

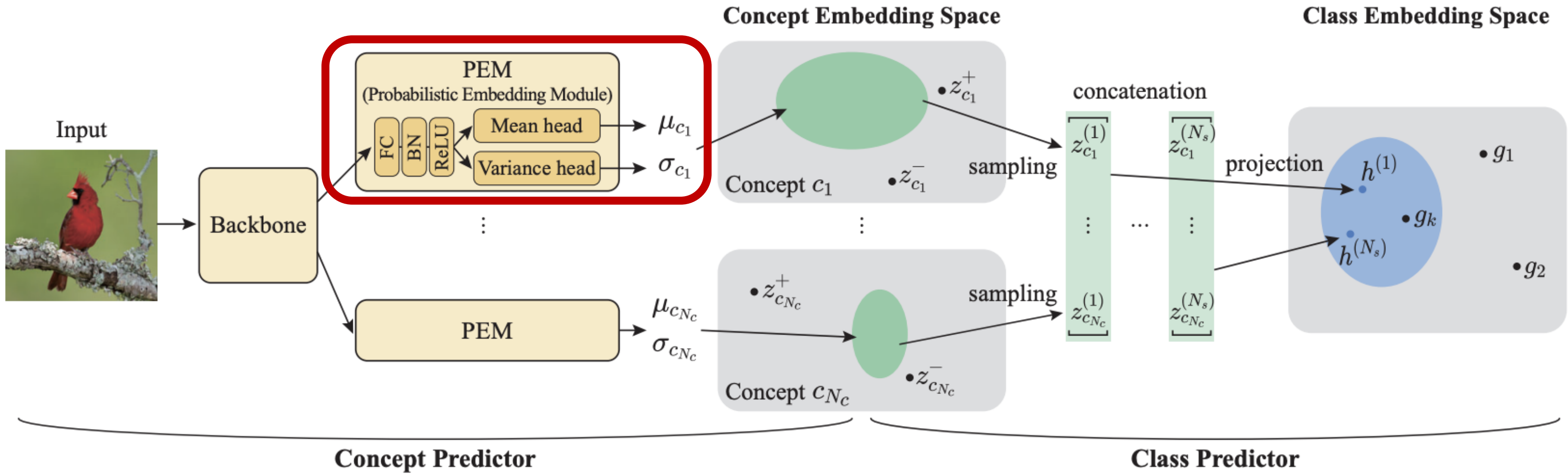


[1] Kim et al. "Probabilistic Concept Bottleneck Models." ICML (2023).

PROBABILISTIC CBMS

Proposed Solution

Each Probabilistic Embedding Module (PEM) generates a mean μ_{c_i} and a variance σ_{c_i} for the concept embedding



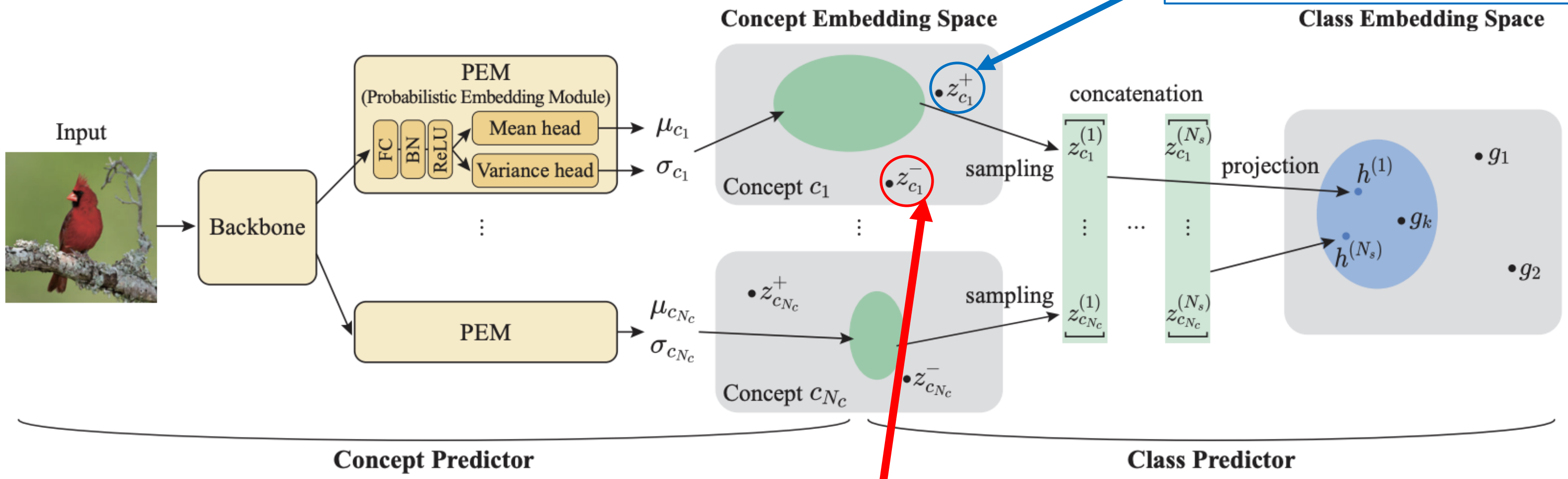
$$p(z_c|x) \sim \mathcal{N}(\mu_c, \mathrm{diag}(\sigma_c))$$

[1] Kim et al. "Probabilistic Concept Bottleneck Models." ICML (2023).

PROBABILISTIC CBMS

Proposed Solution

We learn a set of fixed anchor embeddings representing the concept when it is on vs off



“Anchor” embedding representing the concept when it is “on”

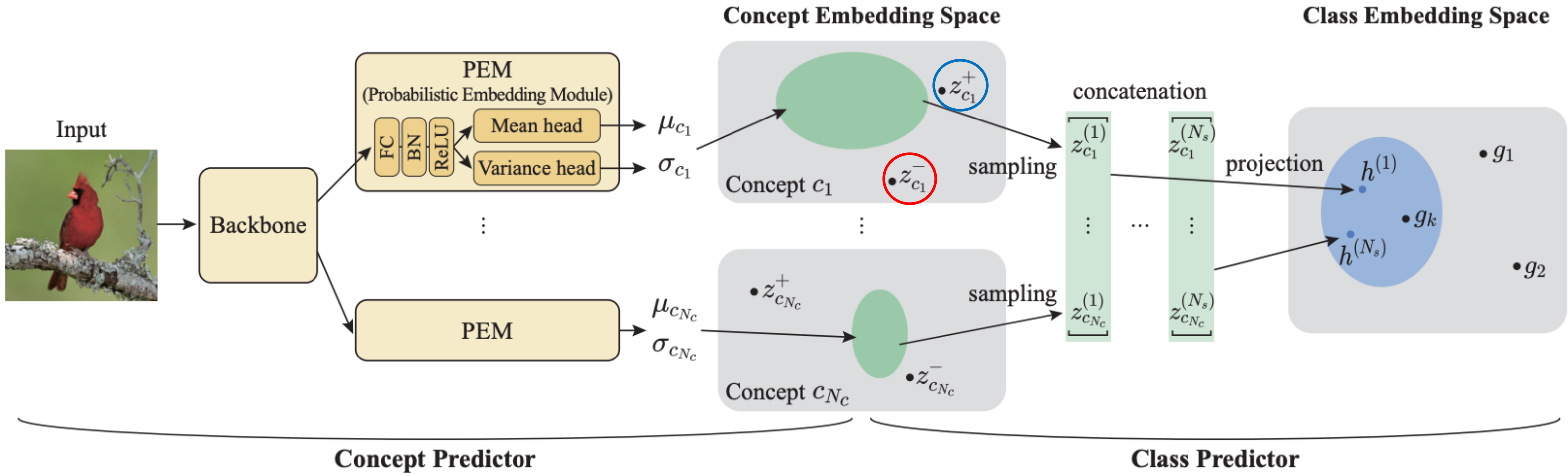
“Anchor” embedding representing the concept when it is “off”

[1] Kim et al. "Probabilistic Concept Bottleneck Models." ICML (2023).

PROBABILISTIC CBMS

Proposed Solution

The distance from the sampled embedding to each anchor can be used to predict a concept



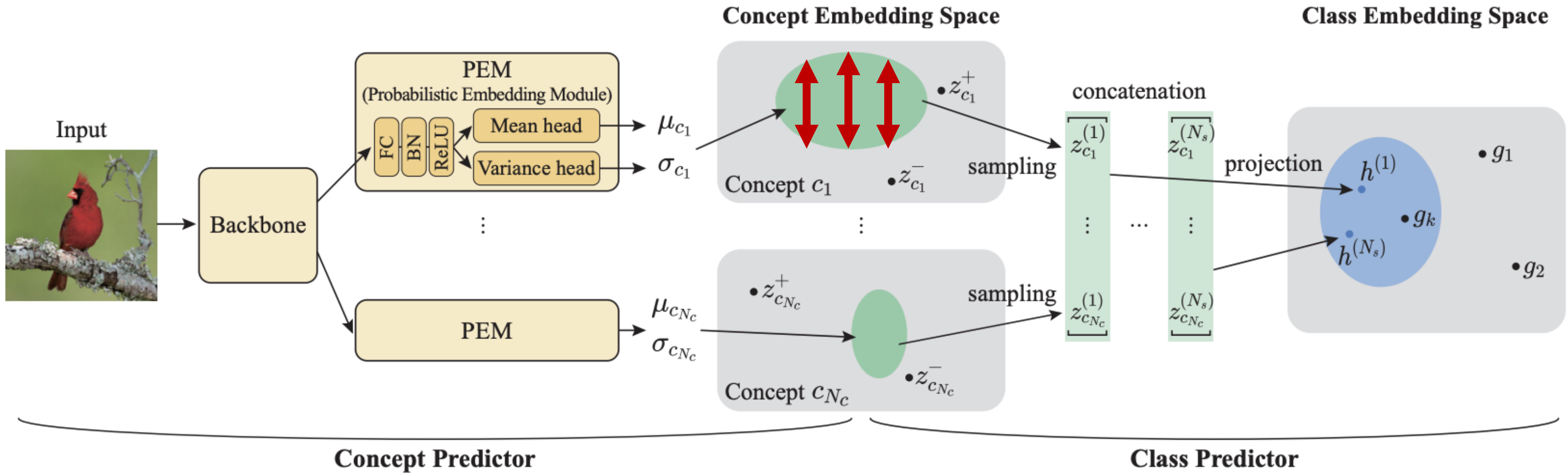
$$p(c = 1 \mid z_c) = \sigma\left(a\left(\|z_c - z_c^-\|_2 - \|z_c - z_c^+\|_2\right)\right)$$

[1] Kim et al. "Probabilistic Concept Bottleneck Models." ICML (2023).

PROBABILISTIC CBMS

Proposed Solution

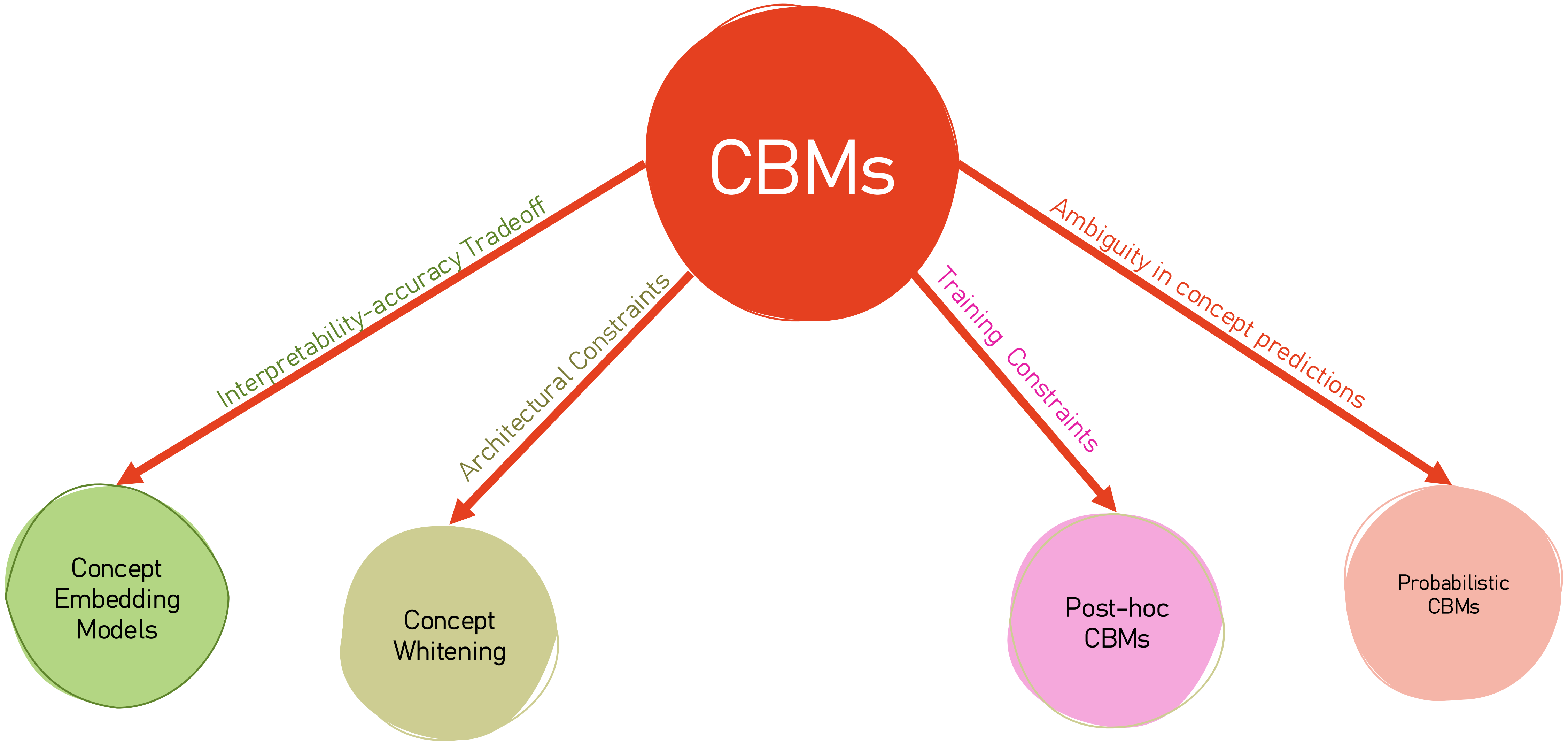
A concept's distribution's volume can be used to quantify its uncertainty



As embeddings are modelled as Gaussians, this is the determinant of the covariance!

[1] Kim et al. "Probabilistic Concept Bottleneck Models." ICML (2023).

END OF THE SPEED DATES!



CBMs

Interpretability-accuracy Tradeoff

Architectural Constraints

Training Constraints

Ambiguity in concept predictions

Concept Embedding Models

Concept Whitening

Post-hoc CBMs

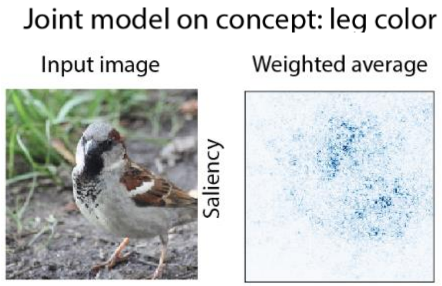
Probabilistic CBMs

DO CBMS PROPERLY LEARN TO EXPLAIN?

DO CBMS PROPERLY LEARN TO EXPLAIN?

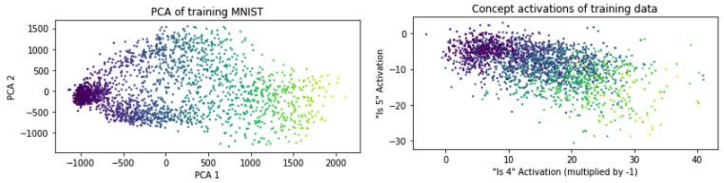
Several recent works suggest CBMs may have issues with unwanted leakage

Attending spurious features



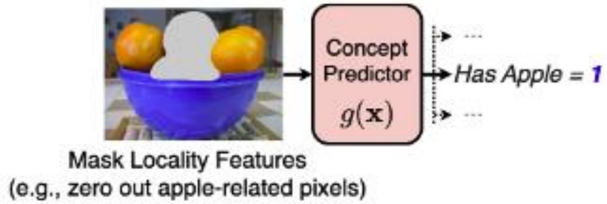
Saliency maps seem to suggest concepts are not properly attended (Margeloiu et al.) [1]

Leaking unwanted information



CBMs may have incentives to encode the entire data representation in the concepts' soft predictions (Mahinpei et al.) [2]

Failing to capture concept locality



CBMs may fail to capture a concept's locality (e.g., physical location) even if it is only found on a fixed feature subset (Raman et al.) [3]

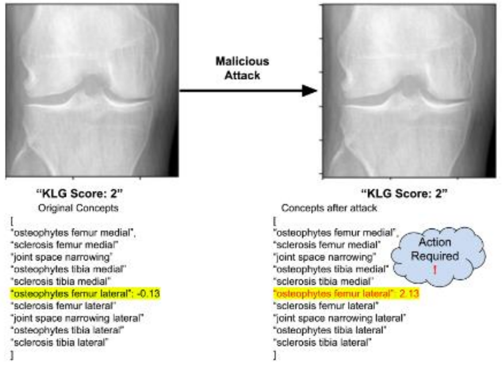
---

[1] Margeloiu et al. "Do concept bottleneck models learn as intended?." ICLR Workshop on Responsible AI (2021).
[2] Mahinpei et al. "Promises and pitfalls of black-box concept learning models." ICML Workshop on Theoretic Foundation, Criticism, and Application of XAI (2021).
[3] Raman et al. "Do Concept Bottleneck Models Respect Localities?." NeurIPS Workshop on XAI in Action (2024).

DO CBMS PROPERLY LEARN TO EXPLAIN?

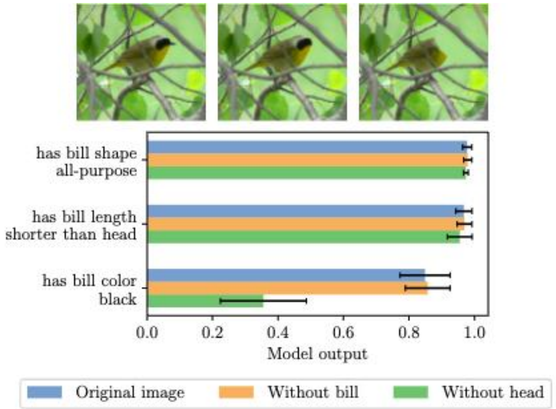
Many more works have dived deeper into these issues!

Adversarial attacks and defences



CBMs concepts predictions can be changed **without affecting the final prediction** (Sinha et al.) [1]

Studying concept correlations



Simple changes to the loss, like **loss weighting**, can help **avoid CBMs exploiting unwanted correlations** (Heidenmann et al.) [2]

Formalisms and metrics for leakage

Metrics for unwanted leakage [3]

Benchmark suite for reasoning robustness [4]

Formalisation of leakage [5]

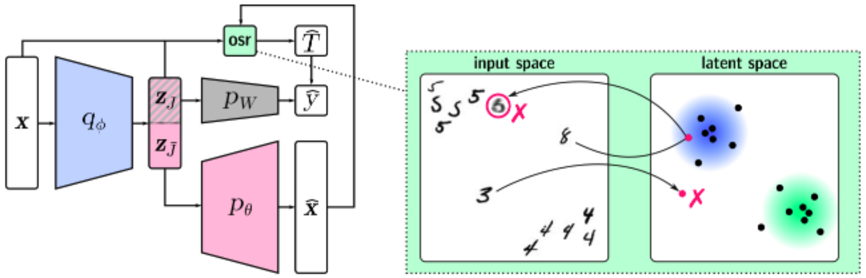
Several works proposed ways to **formalise** or **measure concept** leakage [3, 4, 5]

[1] Sinha et al. "Understanding and enhancing robustness of concept-based models." AAAI (2023).
[2] Heidemann et al. "Concept correlation and its effects on concept-based models." WACV (2023).
[3] Espinosa Zarlenga, Barbiero, Shams et al. "Towards robust metrics for concept representation evaluation." AAAI (2023).
[4] Bortolotti, Marconato, et al. "A Neuro-Symbolic Benchmark Suite for Concept Quality and Reasoning Shortcuts." NeurIPS (2024).
[5] Marconato et al. "Interpretability is in the mind of the beholder: A causal framework for human-interpretable representation learning." Entropy (2023).

STEPS TOWARDS ADDRESSING LEAKAGE

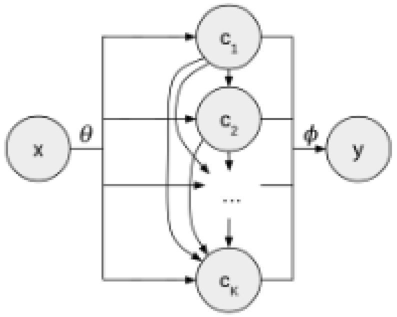
This have brought forth attempts to address or mitigate the effects of leakage:

GlanceNets



[Main Idea] Frame leakage in terms of disentanglement learning and use an open-set recognition to detect it at inference

Autoregressive CBMs



[Main Idea] Reduce leakage between concepts by modeling cross-concept relationships using an autoregressive architecture

[1] Marconato et al. "Glancenets: Interpretable, leak-proof concept-based models." NeurIPS (2022).
[2] Havasi et al. "Addressing leakage in concept bottleneck models." NeurIPS (2022).

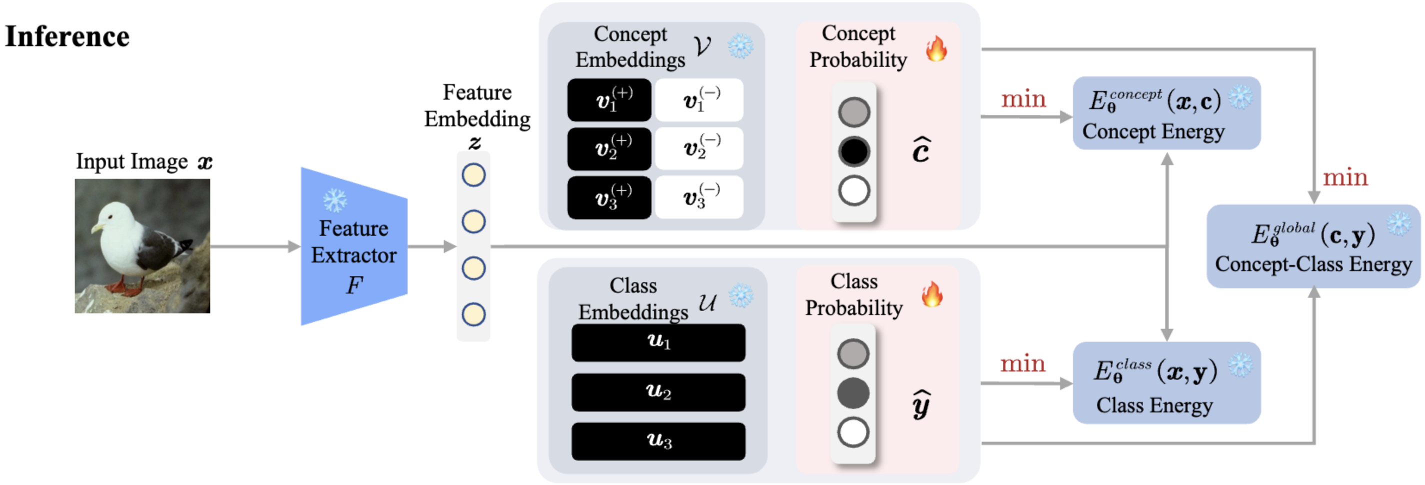
RECENT DIRECTIONS

CBMs have become very popular in XAI with several active areas of research:

RECENT DIRECTIONS

CBMs have become very popular in XAI with several active areas of research:

1. Capturing more complex relationships between concepts and tasks labels



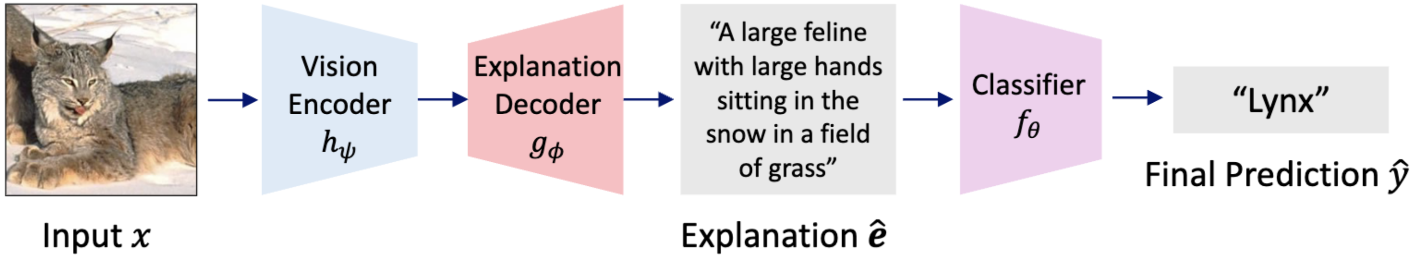
Energy-based Concept Bottleneck Models (Xu et al., 2024)

[1] Xu et al. "Energy-based concept bottleneck models: unifying prediction, concept intervention, and conditional interpretations." ICLR (2024).

RECENT DIRECTIONS

CBMs have become **very popular in XAI** with several active **areas of research**:

1. Capturing more **complex relationships** between concepts and tasks labels
2. Producing entirely **language-based bottlenecks** (accepted to this AAAI!)



Explanation Bottleneck Models (Yamaguchi et al.)

[1] Yamaguchi et al. "Explanation Bottleneck Models." AAAI (2025).

RECENT DIRECTIONS

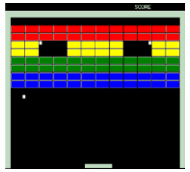
CBMs have become **very popular in XAI** with several active **areas of research**:

1. Capturing more **complex relationships** between concepts and tasks labels
2. Producing entirely **language-based bottlenecks** (accepted to this AAAI!)
3. Exploring concepts in **modalities and tasks** other than **supervised visual tasks**

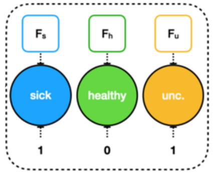
Graph Data
(e.g., Xuanyuan et al.)

Tabular Data
(e.g., Espinosa Zarlenga et al.)

RL Tasks
(e.g., Ye et al.)



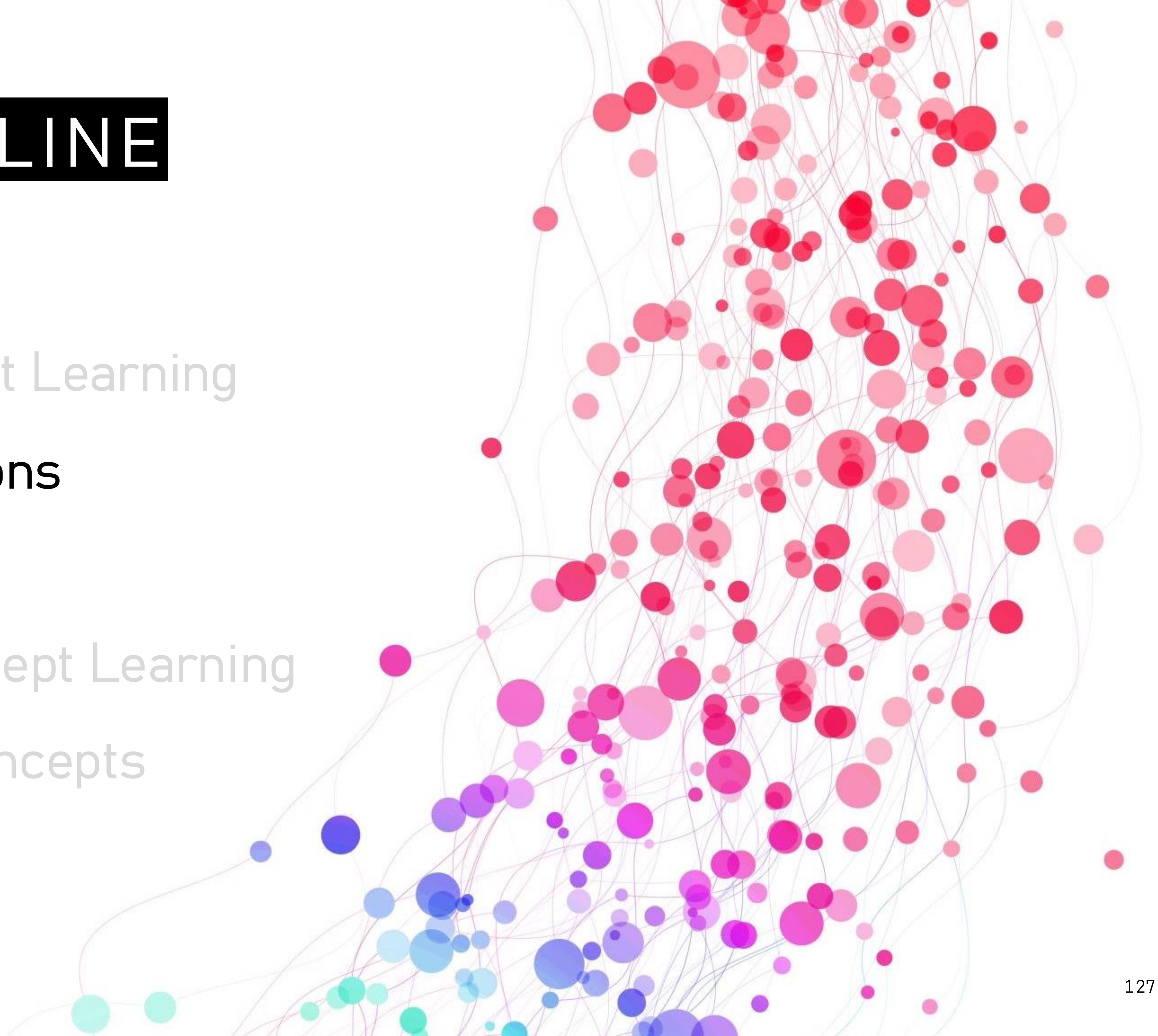
Time Series Data
(e.g., Kazhdan et al.)



[1] Xuanyuan al. "Global concept-based interpretability for graph neural networks via neuron analysis." AAAI (2023).
[2] Espinosa Zarlenga et al. "Tabcbm: Concept-based interpretable neural networks for tabular data." TMLR (2024).
[3] Ye et al. "Concept-based interpretable reinforcement learning with limited to no human labels." ICML (2024).
[4] Kazhdan et al. "MEME: generating RNN model explanations via model extraction." arXiv (2020).

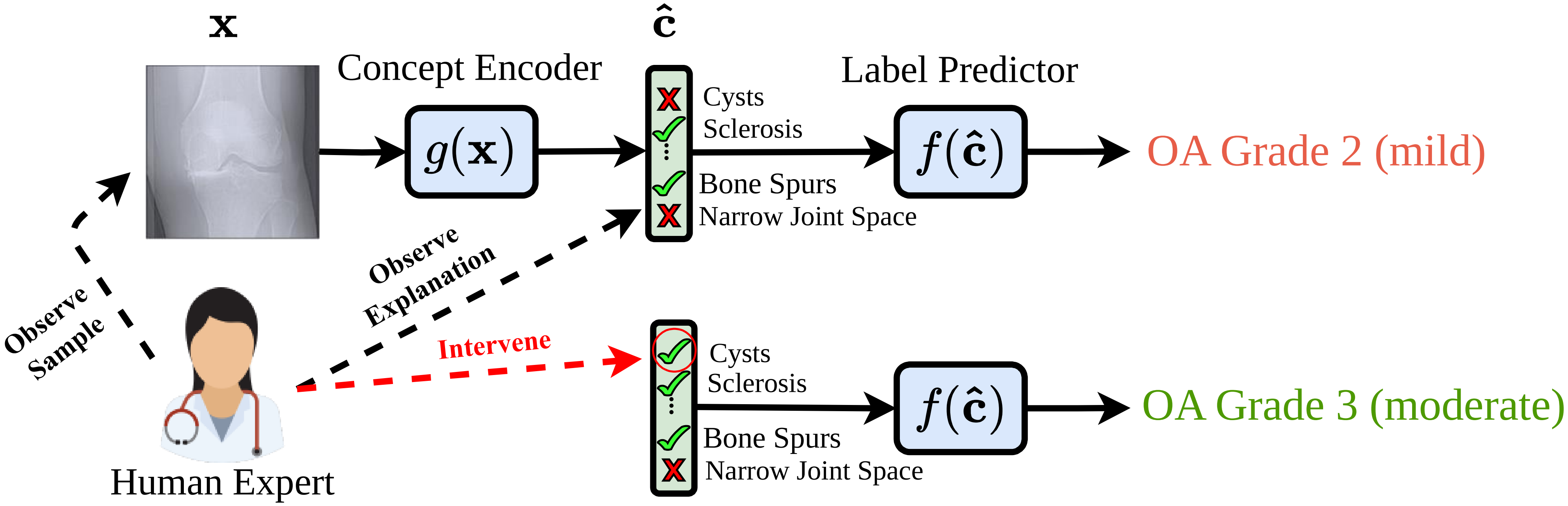
TUTORIAL OUTLINE

1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break
5. Unsupervised Concept Learning
6. Reasoning With Concepts
7. Future Directions
8. Q&A



RECALL CONCEPT INTERVENTIONS

Concept interventions enable experts to **“inject” knowledge during inference**



SOME CONCEPTS ARE BETTER THAN OTHERS

When intervening on a CBM, it is important to realise that some concepts are:

1. Less **informative** than others (e.g., redundant w.r.t. other concepts)
2. Less **certain** than others (e.g., due to occlusions or inherent difficulties)

To identify a Raven from a Crow, "*tail shape*"
is more informative than "*wing color*"

Concept "*Belly Color*" is partially occluded

[1] mage taken from Leah Lin , "How to Identify a Crow and a Raven: Key Features Explained" Birdfy (2024).

SOME CONCEPTS ARE BETTER THAN OTHERS

When intervening on a CBM, it is important to realise that some concepts are:

1. Less **informative** than others (e.g., redundant w.r.t. other concepts)
2. Less **certain** than others (e.g., due to occlusions or inherent difficulties)

Hence, an intervention's effectiveness **depends on the intervened concept!**

SELECTING MEANINGFUL CONCEPTS

Intervention policies select which concept to intervene on next by assigning each concept c_i a score s_i and selecting concepts in decreasing score order:

Given $\boldsymbol{x}$ and concept predictions $\hat{\boldsymbol{c}}$, what concept should I intervene on next ***to minimize my model's task uncertainty****?*

[1] Shin et al. "A Closer Look at the Intervention Procedure of Concept Bottleneck Models." ICML 2023.

SELECTING MEANINGFUL CONCEPTS

Intervention policies select which concept to intervene on next by assigning each concept c_i a score s_i and selecting concepts in decreasing score order:

Uncertainty of concept prediction (UCP)

Select the concept c_i with the highest predicted entropy $\mathrm{s_i} = \mathcal{H}(\hat{c}_i)$

[1] Shin et al. "A Closer Look at the Intervention Procedure of Concept Bottleneck Models." ICML 2023.

SELECTING MEANINGFUL CONCEPTS

Intervention policies select which concept to intervene on next by assigning each concept c_i a score s_i and selecting concepts in decreasing score order:

Uncertainty of concept prediction (UCP)

Select the concept c_i with the highest predicted entropy $s_i = \mathcal{H}(\hat{c}_i)$

Contribution of concept on target prediction (CCTP)

Select the concept c_i with the highest contribution on target prediction $s_i = \sum_{j=1}^{L} \left| \hat{c}_i \frac{\partial f_j(x)}{\partial \hat{c}_i} \right|$.

[1] Shin et al. "A Closer Look at the Intervention Procedure of Concept Bottleneck Models." ICML 2023.

SELECTING MEANINGFUL CONCEPTS

Intervention policies select which concept to intervene on next by assigning each concept c_i a score s_i and selecting concepts in decreasing score order:

Uncertainty of concept prediction (UCP)

Select the concept c_i with the highest predicted entropy $s_i = \mathcal{H}(\hat{c}_i)$

Contribution of concept on target prediction (CCTP)

Select the concept c_i with the highest contribution on target prediction $s_i = \sum_{j=1}^{L} \left| \hat{c}_i \frac{\partial f_j(x)}{\partial \hat{c}_i} \right|$.

Expected change in target prediction (ECTP)

Select the concept c_i with the highest expected change in the target predictive distribution

$$s_i = (1 - \hat{c}_i) D_{KL}\left(\hat{y}_{\hat{c}_i=0} || \hat{y}\right) + \hat{c}_i D_{KL}\left(\hat{y}_{\hat{c}_i=1} || \hat{y}\right)$$

[1] Shin et al. "A Closer Look at the Intervention Procedure of Concept Bottleneck Models." ICML 2023.

SELECTING MEANINGFUL CONCEPTS

Intervention policies select which concept to intervene on next by assigning each concept c_i a score s_i and selecting concepts in decreasing score order:

Uncertainty of concept prediction (UCP)

Select the concept c_i with the highest predicted entropy $\mathrm{s_i} = \mathcal{H}(\hat{c}_i)$

Contribution of concept on target prediction (CCTP)

Select the concept c_i with the highest contribution on target prediction $s_i = \sum_{j=1}^{L} \left| \hat{c}_i \frac{\partial f_j(x)}{\partial \hat{c}_i} \right|$.

Expected change in target prediction (ECTP)

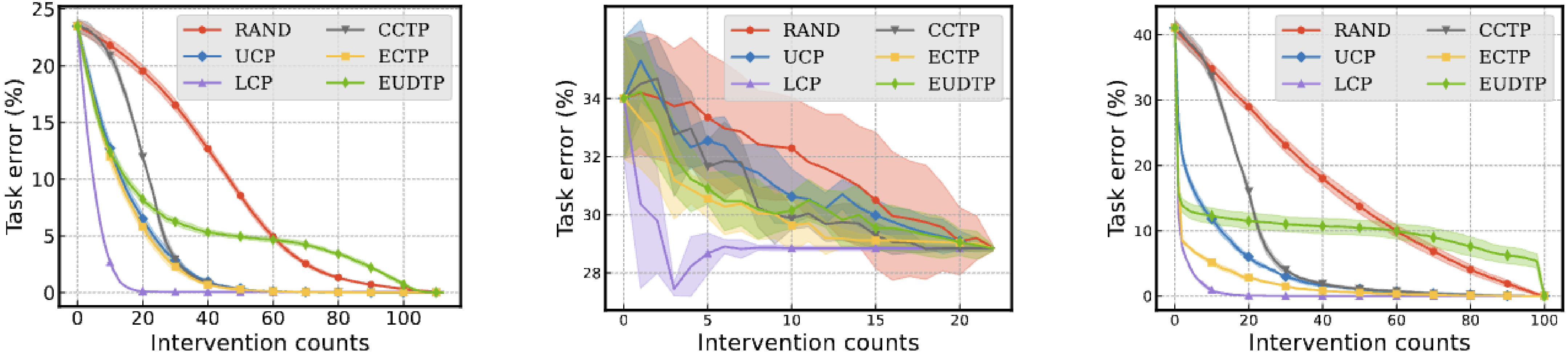
Select the concept c_i with the highest expected change in the target predictive distribution
$s_i = (1 - \hat{c}_i) D_{KL}\left(\hat{y}_{\hat{c}_i=0} || \hat{y}\right) + \hat{c}_i D_{KL}\left(\hat{y}_{\hat{c}_i=1} || \hat{y}\right)$

One can think of these policies as proxies for a concept's information content and certainty

[1] Shin et al. "A Closer Look at the Intervention Procedure of Concept Bottleneck Models." ICML 2023.

INTERVENTION POLICIES RESULTS

This leads to significantly different intervention curves:



(a) CUB (b) SkinCon (c) Synthetic

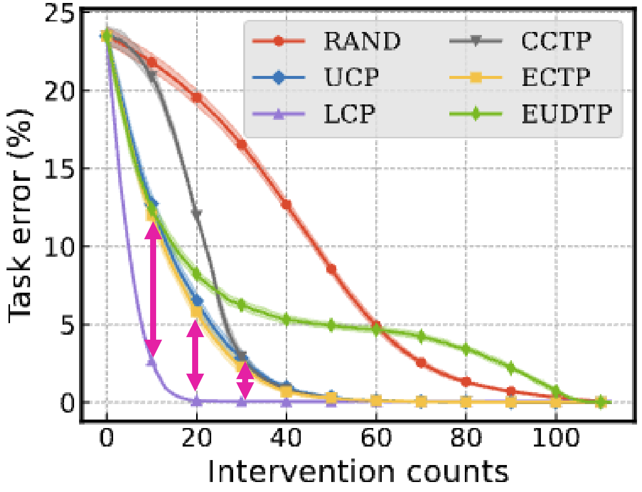
Best performing non-oracle policy is Expected change in target prediction (ECTP) but even the simple Uncertainty of concept prediction (UCP) is significantly better than the random policy (RAND)

(Intuition: one should select the concept leading to the highest expected change in the task's distribution)

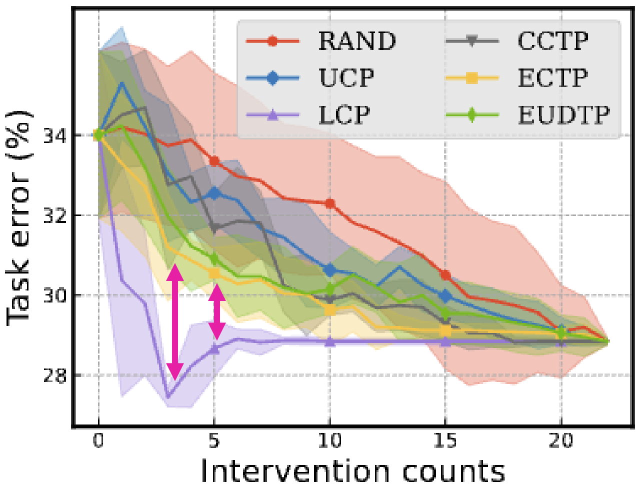
[1] Shin et al. "A Closer Look at the Intervention Procedure of Concept Bottleneck Models." ICML 2023.

INTERVENTION POLICIES RESULTS

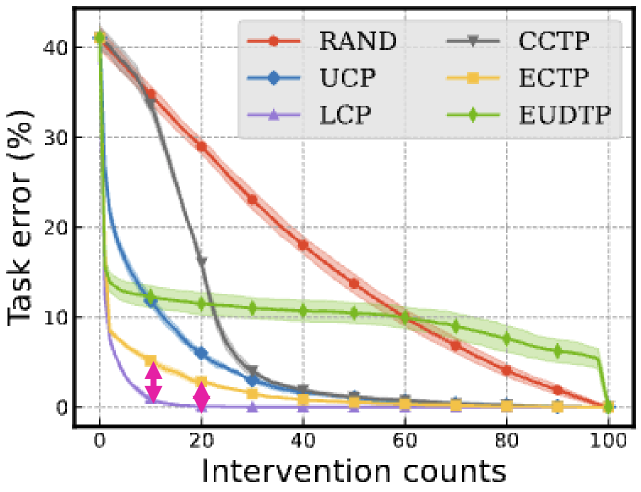
This leads to significantly different intervention curves:



(a) CUB



(b) SkinCon



(c) Synthetic

We still observe a significant gap between the best policy and an optimal greedy policy (LCP)

[1] Shin et al. "A Closer Look at the Intervention Procedure of Concept Bottleneck Models." ICML 2023.

COMBINING POLICIES

It may be possible to shorten this gap by learning a weighting between the concept uncertainty and the expected change in current prediction policies

Cooperative Prediction Intervention Policy (CooP)

$$s_i = \alpha\mathcal{H}(\hat{c}_i) + \beta\left|E_{v\sim p(c_i\,|\,x)}\left[\hat{y}_{\hat{c}_i=v} - \hat{y}\right]\right| + \gamma q_i$$

[1] Chauhan et al. "Interactive concept bottleneck models." AAAI (2023).

COMBINING POLICIES

It may be possible to shorten this gap by learning a weighting between the concept uncertainty and the expected change in current prediction policies

Cooperative Prediction Intervention Policy (CooP)

$$s_i = \alpha \underbrace{\mathcal{H}(\hat{c}_i)}_{\text{Uncertainty of concept prediction}} + \beta \underbrace{\left| E_{v \sim p(c_i \mid x)}\left[\hat{y}_{\hat{c}_i = v} - \hat{y}\right]\right|}_{\text{Expected change to the current predicted label if we were to intervene on c\_i based on c\_i's current prediction}} + \gamma \underbrace{q_i}_{\text{Cost of the intervention}}$$

[1] Chauhan et al. "Interactive concept bottleneck models." AAAI (2023).

COMBINING POLICIES

It may be possible to shorten this gap by selecting concepts based on both

Can we do any better than this? Can we avoid the need for calculating computationally expensive scores all concepts?

We can find the hyperparameters α, β, and γ using a concept-annotated validation set

[1] Chauhan et al. "Interactive concept bottleneck models." AAAI (2023).

INSIGHT #1: TRAINING-TIME INCENTIVES

There is a disconnect between how CBMs are trained and how they are used at test-time when they are intervened on

Train-time

Predict task labels from a set of **entirely predicted concepts**

≠

Test-time

Predict task labels from a set of concepts where **some are ground-truth** labels and others are predicted

During training, concept-based models are not even aware they may be intervened on!

During testing, concept interventions may lead to out-of-distribution bottlenecks for a CBM!

[1] Espinosa Zarlenga et al. "Learning to receive help: Intervention-aware concept embedding models." NeurIPS (2023)

INSIGHT #2: USEFUL TRAINING FEEDBACK

If we know all task and concept labels, we can compute the optimal greedy concept intervention:

$$c_*(\mathbf{x}, \mu, \mathbf{c}, y) := \arg\max_{1 \leq i \leq k} f\left(\tilde{g}(\mathbf{x}, \mu \vee \mathbb{1}_i, \mathbf{c})\right)_y$$

(**Translation**: Attempt every intervention and select that one maximises the ground truth label's confidence)

This is feedback we have at training time and can use to learn an intervention policy!

[1] Espinosa Zarlenga et al. "Learning to receive help: Intervention-aware concept embedding models." NeurIPS (2023)

INSIGHT #3: INTERVENTIONS CAN BE DIFFERENTIABLE

When modelling concepts as being a **mixture of two learnable embeddings** $\{\boldsymbol{c}_i^{+}, \boldsymbol{c}_i^{-}\}$ as in CEMs, **interventions are differentiable**:

Original Embedding Construction

$$\hat{\boldsymbol{c}}_{\boldsymbol{i}} \coloneqq \hat{p}_i \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{+} + (1 - \hat{p}_i)\, \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{-}$$

Intervened Embedding Construction

$$\hat{\boldsymbol{c}}_{\boldsymbol{i}} \coloneqq (\mu_i c_i + (1 - \mu_i)\hat{p}_i)\, \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{+} + \big(1 - (\mu_i c_i + (1 - \mu_i)\hat{p}_i)\big)\, \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{-}$$

Whether we intervene on the $\boldsymbol{i}$-th concept (can be relaxed to be in [0,1])

[1] Espinosa Zarlenga et al. "Learning to receive help: Intervention-aware concept embedding models." NeurIPS (2023)

INSIGHT #3: INTERVENTIONS CAN BE DIFFERENTIABLE

When modelling concepts as being a mixture of two learnable embeddings $\{\boldsymbol{c}_i^{+}, \boldsymbol{c}_i^{-}\}$ as in CEMs, interventions are differentiable:

Original Embedding Construction

$$\hat{\boldsymbol{c}}_{\boldsymbol{i}} \coloneqq \hat{p}_i \hat{\boldsymbol{c}}_i^{+} + (1 - \hat{p}_i)\, \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{-}$$

Intervened Embedding Construction

$$\hat{\boldsymbol{c}}_{\boldsymbol{i}} \coloneqq (\mu_i c_i + (1 - \mu_i)\hat{p}_i)\, \hat{\boldsymbol{c}}_i^{+} + \left(1 - (\mu_i c_i + (1 - \mu_i)\hat{p}_i)\right) \hat{\boldsymbol{c}}_{\boldsymbol{i}}^{-}$$

Whether we intervene on the $\boldsymbol{i}$-th concept (can be relaxed to be in [0,1])

This means an intervention policy deciding μ_i can be learnt via gradient descent!

[1] Espinosa Zarlenga et al. "Learning to receive help: Intervention-aware concept embedding models." NeurIPS (2023)

INTERVENTION-AWARE MODELS

Intervention-Aware Concept Embedding Models (IntCEMs) incorporate these insights into an end-to-end architecture that:

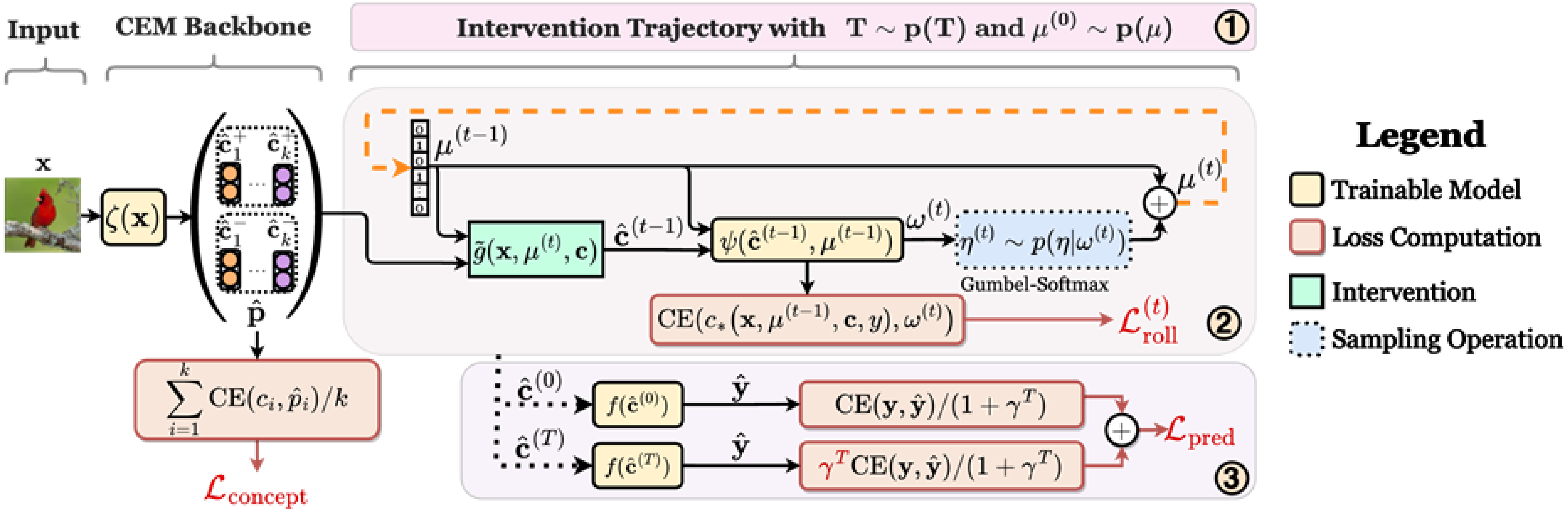
1. Introduces an intervention-aware training loss that encourages receptiveness to concept interventions at test-time

2. Learns an efficient intervention policy in an end-to-end fashion.

[1] Espinosa Zarlenga et al. "Learning to receive help: Intervention-aware concept embedding models." NeurIPS (2023)

HOW TO TRAIN YOUR INTCEM?

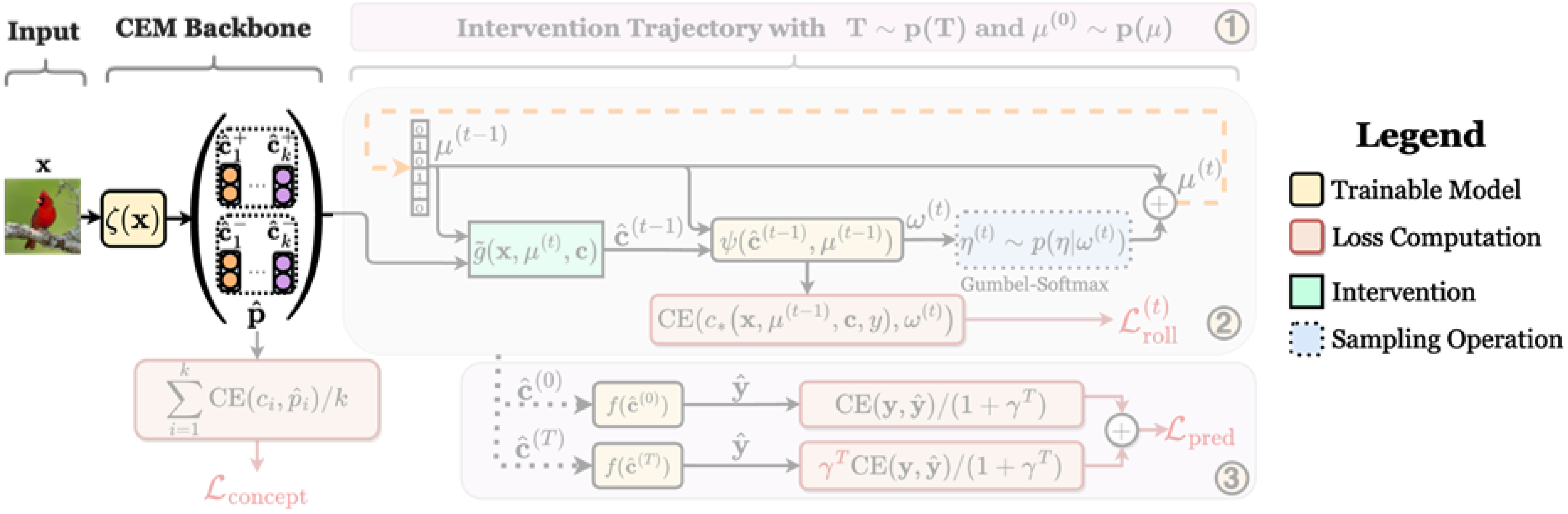
This can be done using an end-to-end neural architecture:



[1] Espinosa Zarlenga et al. "Learning to receive help: Intervention-aware concept embedding models." NeurIPS (2023)

HOW TO TRAIN YOUR INTCEM?

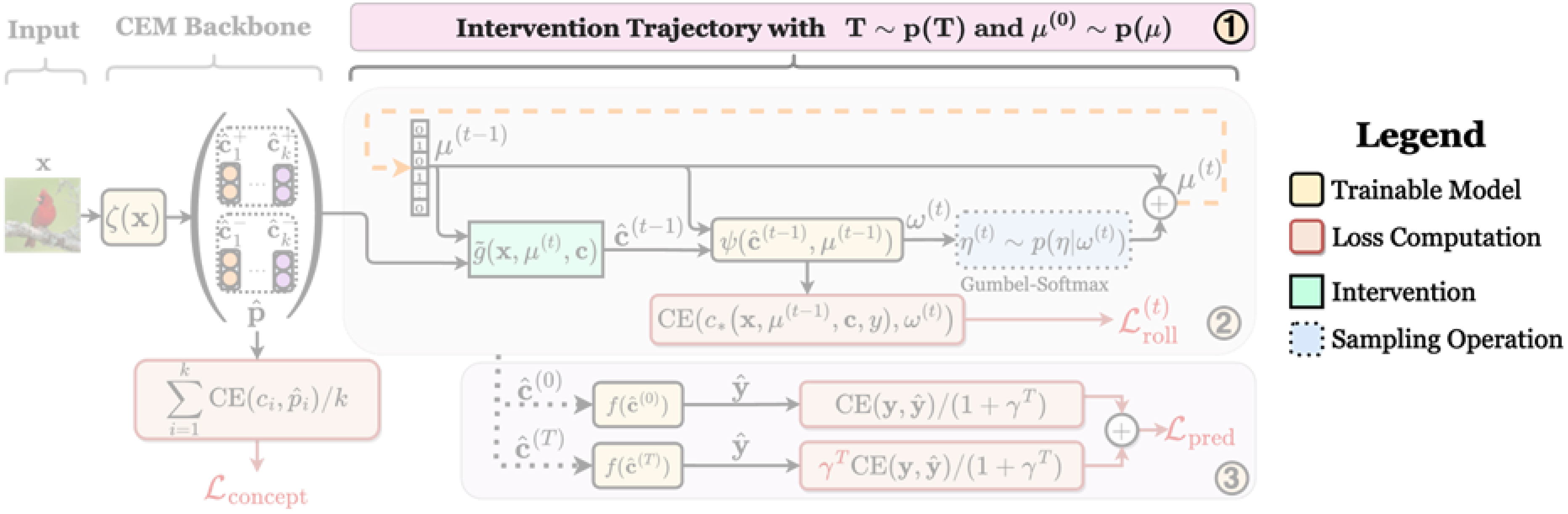
This can be done using an end-to-end neural architecture:



(1) Construct a positive and negative embedding for each training concept

HOW TO TRAIN YOUR INTCEM?

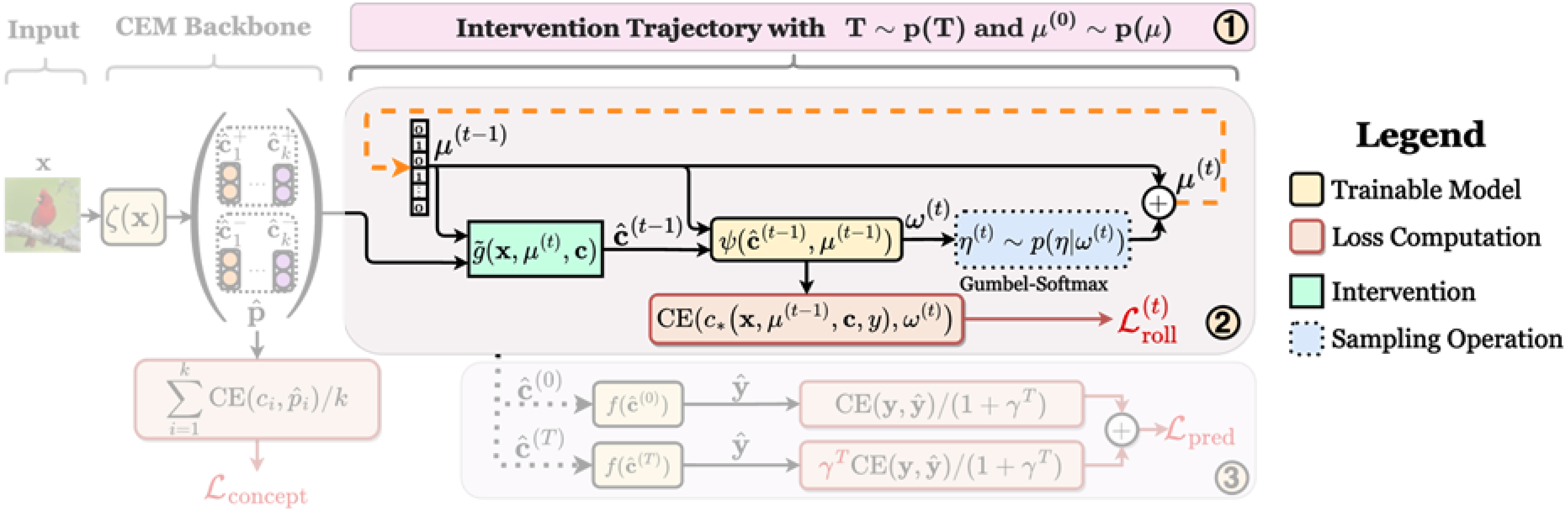
This can be done using an end-to-end neural architecture:



(2) Randomly select a subset of concepts which we will initially intervene on and a number of interventions T we will perform in this training step

HOW TO TRAIN YOUR INTCEM?

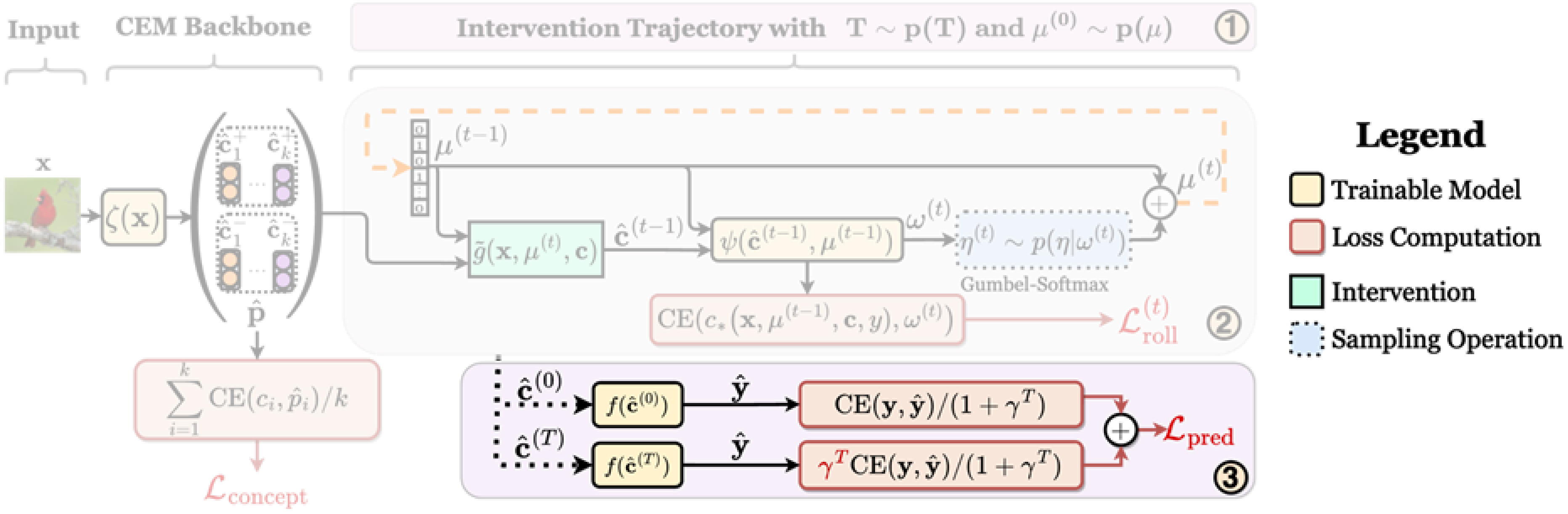
This can be done using an end-to-end neural architecture:



(3) Recursively sample a trajectory of T interventions from this set using a learnable intervention policy. We train this policy to align to the “oracle” optimal policy.

HOW TO TRAIN YOUR INTCEM?

This can be done using an end-to-end neural architecture:



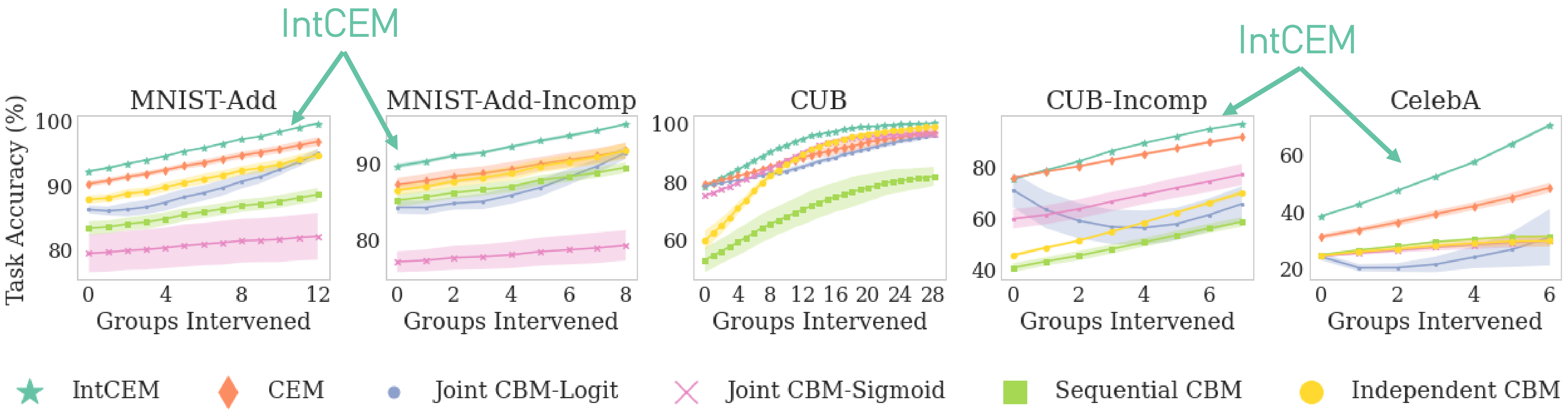
(4) Penalise the model more heavily for mispredicting the task label at the end of the intervention trajectory vs mispredicting the task label at the start of the trajectory

$$\mathbb{E}_{(x,c,y)\sim\mathcal{D}}\left[\frac{\mathcal{L}_{task}\left(y,f\left(\hat{c}^{(0)}\right)\right)+\gamma^T\mathcal{L}_{task}\left(y,f\left(\hat{c}^{(T)}\right)\right)}{1+\gamma^T}\right]$$

WHAT DOES ALL OF THIS GIVE YOU?

WHAT DOES ALL OF THIS GIVE YOU?

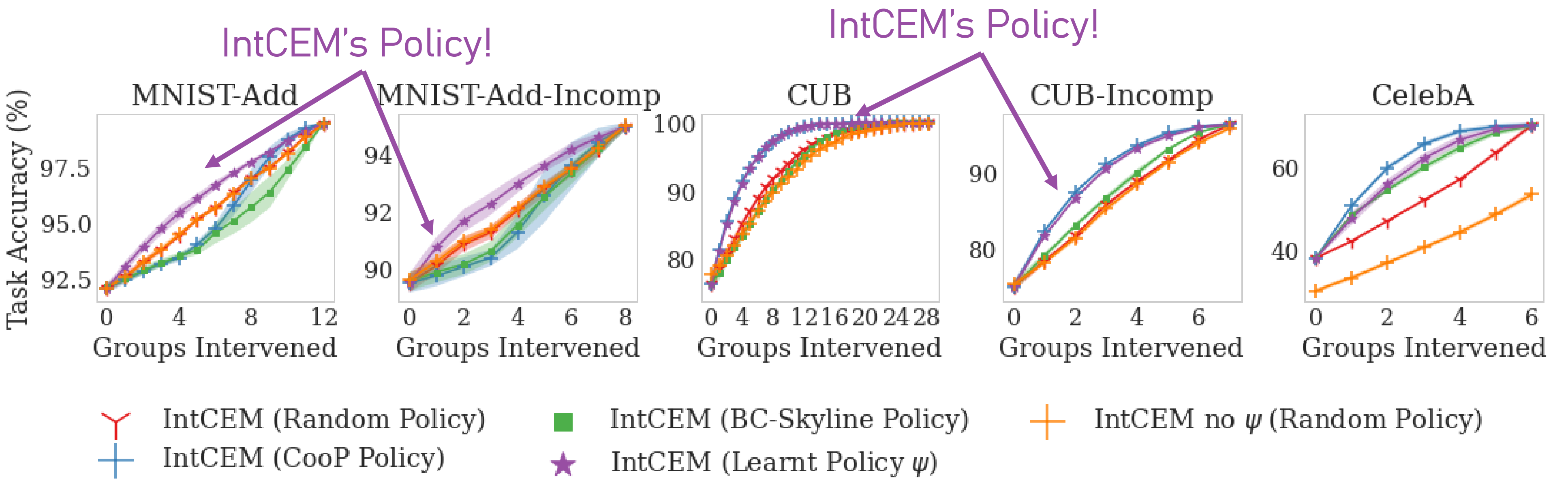
(1) A model that is much better at receiving test-time feedback **even if concepts are intervened in a random order**


IntCEM
IntCEM

Up to 9% in absolute improvement when 25% of concepts are randomly selected to be intervened on!

WHAT DOES ALL OF THIS GIVE YOU?

(2) An **efficient intervention policy** that selects useful concepts to intervene on next

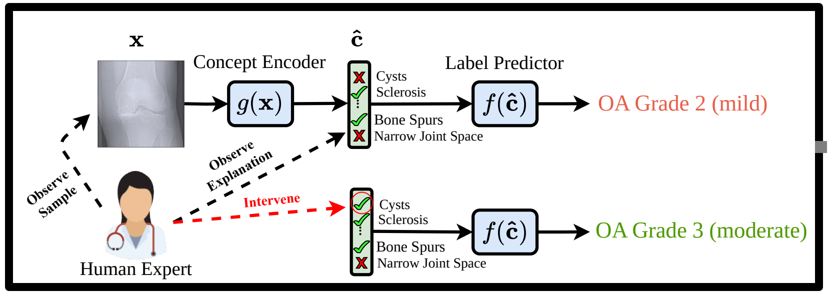


CRITICAL LIMITATIONS OF INTERVENTIONS

When intervening, we assume that concept interventions are:

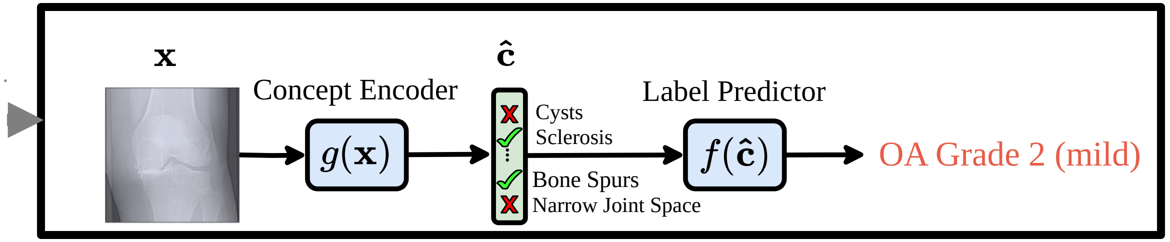
1. **Transient**: after an intervention is made, it is **forgotten**

If an intervention is made on a sample $\boldsymbol{x}$



Some time later…

The same mistake will be made if $\boldsymbol{x}$ is seen again



CRITICAL LIMITATIONS OF INTERVENTIONS

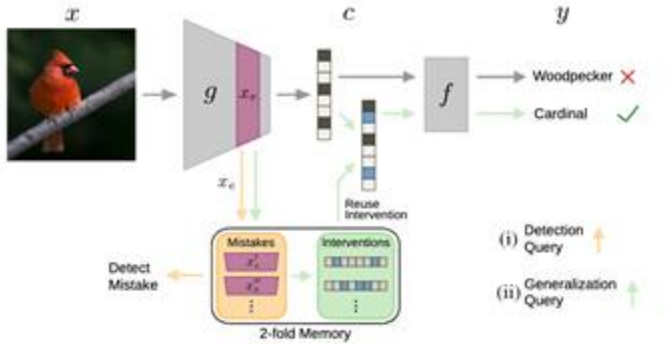
When intervening, we assume that concept interventions are:

1. **Transient**: after an intervention is made, it is forgotten
2. **Independent**: intervening on concept c_i will not affect other concepts' values

RELAXING KEY ASSUMPTIONS

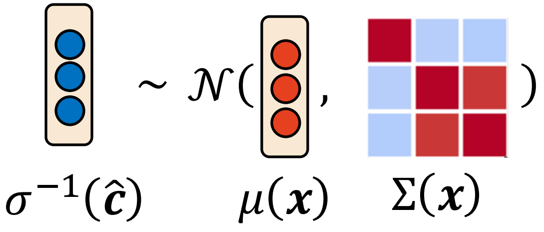
These constraints can be relaxed via clever modelling:

Concept Bottleneck Memory Models



Addresses: Transient nature of a concept intervention
Approach: Introduce a **learnable memory module** that keeps previously seen interventions and re-applies them in the future,

Stochastic Concept Bottleneck Models



Addresses: the assumption that concepts are independent
Approach: Model the predicted concept logits as a **normal distribution with a (learnable) non-diagonal covariance.**

[1] Steinmann et al. "Learning to Intervene on Concept Bottlenecks." ICML (2024).
[2] Vandenhirtz, Laguna et al. "Stochastic Concept Bottleneck Models." NeurIPS (2024).

CAN INTERVENTIONS EXTEND BEYOND CBMS?

So far, the concept intervention strategies we have considered require one to operate on a CBM-like model

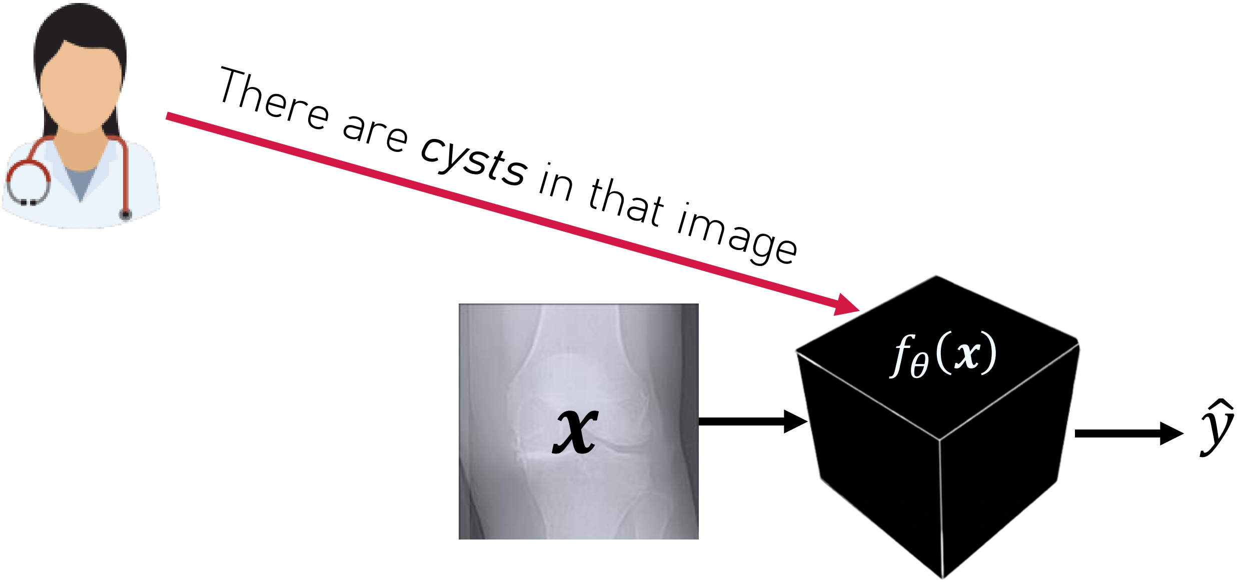
Could we potentially extend these ideas to models beyond CBMs?

[1] Laguna, Marcinkevičs, et al. "Beyond Concept Bottleneck Models: How to Make Black Boxes Intervenable?." NeurIPS (2024).

INJECTING KNOWLEDGE TO BLACK BOXES

Given a black-box model $f_\theta(\boldsymbol{x}) = g_\psi\left(h_\phi(\boldsymbol{x})\right)$ and a test sample $\boldsymbol{x}$, we may want to

inject knowledge about the presence or absence of a concept in $\boldsymbol{x}$ at test time



[1] Laguna, Marcinkevičs, et al. "Beyond Concept Bottleneck Models: How to Make Black Boxes Intervenable?," NeurIPS (2024).

INJECTING KNOWLEDGE TO BLACK BOXES

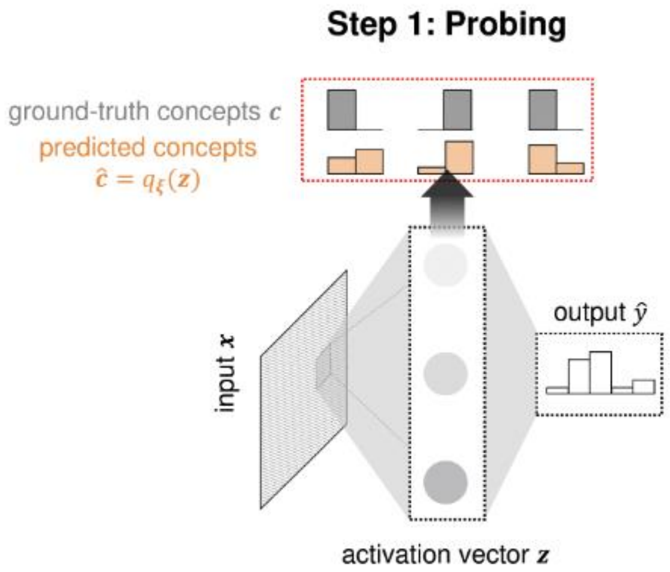
Given a black-box model $f_\theta(\boldsymbol{x}) = g_\psi\left(h_\phi(\boldsymbol{x})\right)$ and a test sample x, we may want to inject knowledge about the presence or absence of a concept in x at test time

If we have a concept-annotated validation set $\left\{\left(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}, y^{i}\right)\right\}_i$, we can do this!

[1] Laguna, Marcinkevičs, et al. "Beyond Concept Bottleneck Models: How to Make Black Boxes Intervenable?." NeurIPS (2024).

BLACK-BOX INTERVENABILITY: PROBING

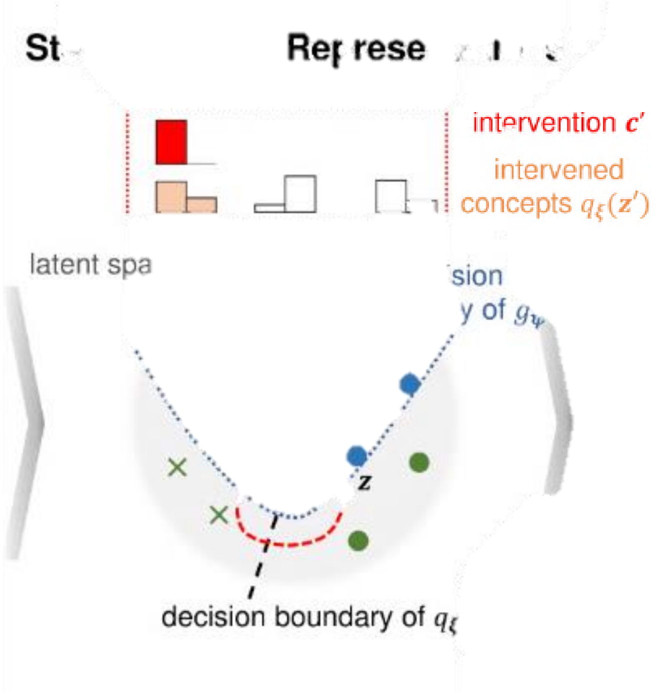
We first learn a multivariate probe $\hat{\mathbf{c}} = \xi(\boldsymbol{z})$ that predicts all concepts given the latent space $\mathbf{z} = h_\phi(x)$ using the annotated validation set



[1] Laguna, Marcinkevičs, et al. "Beyond Concept Bottleneck Models: How to Make Black Boxes Intervenable?," NeurIPS (2024).

BLACK-BOX INTERVENABILITY: EDITING

Given user-provided concept labels c' for sample $\boldsymbol{x}$, we edit the representation $\mathrm{z} = h_{\phi}(x)$ so that it maps to c' as predicted by the probe $\xi(\boldsymbol{z})$



We find a new latent representation z' by solving

$$\arg\min_{\boldsymbol{z}'} \lambda \mathcal{L}^{\boldsymbol{c}} \underbrace{\left(q_{\boldsymbol{\xi}}\left(\boldsymbol{z}'\right), \boldsymbol{c}'\right)}_{\text{Concept alignment}} + \underbrace{d\left(\boldsymbol{z}, \boldsymbol{z}'\right)}_{\text{Distance Penalty}}$$

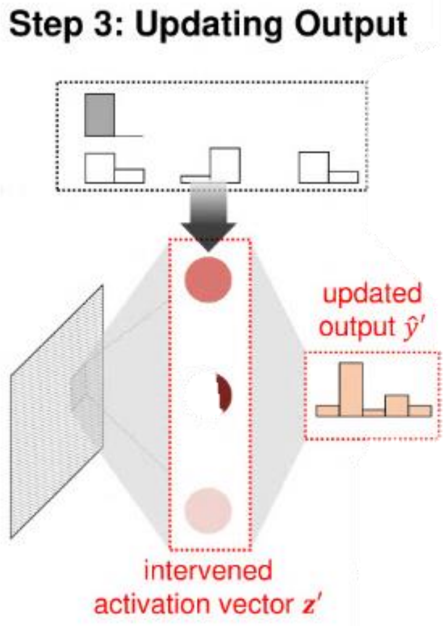
Make the latent representation map to the set of user-provided concepts c'

Keep the new latent representation as close as possible to the original

[1] Laguna, Marcinkevičs, et al. "Beyond Concept Bottleneck Models: How to Make Black Boxes Intervenable?," NeurIPS (2024).

BLACK-BOX INTERVENABILITY: OUTPUT

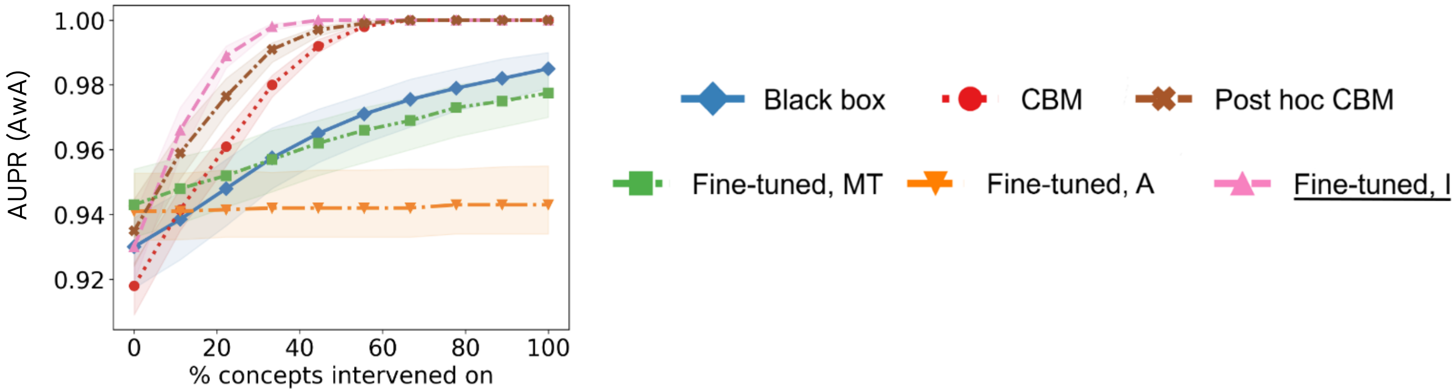
Finally, fed the edited representation $\mathbf{z}'$ to the second part of the DNN to obtain an updated prediction $\widehat{y}' = g_\psi(z')$



[1] Laguna, Marcinkevičs, et al. "Beyond Concept Bottleneck Models: How to Make Black Boxes Intervenable?." NeurIPS (2024).

WHAT THIS GIVES YOU

This process allows you to improve the task accuracy of a black-box model when you have extra test-time knowledge in the form of concepts labels

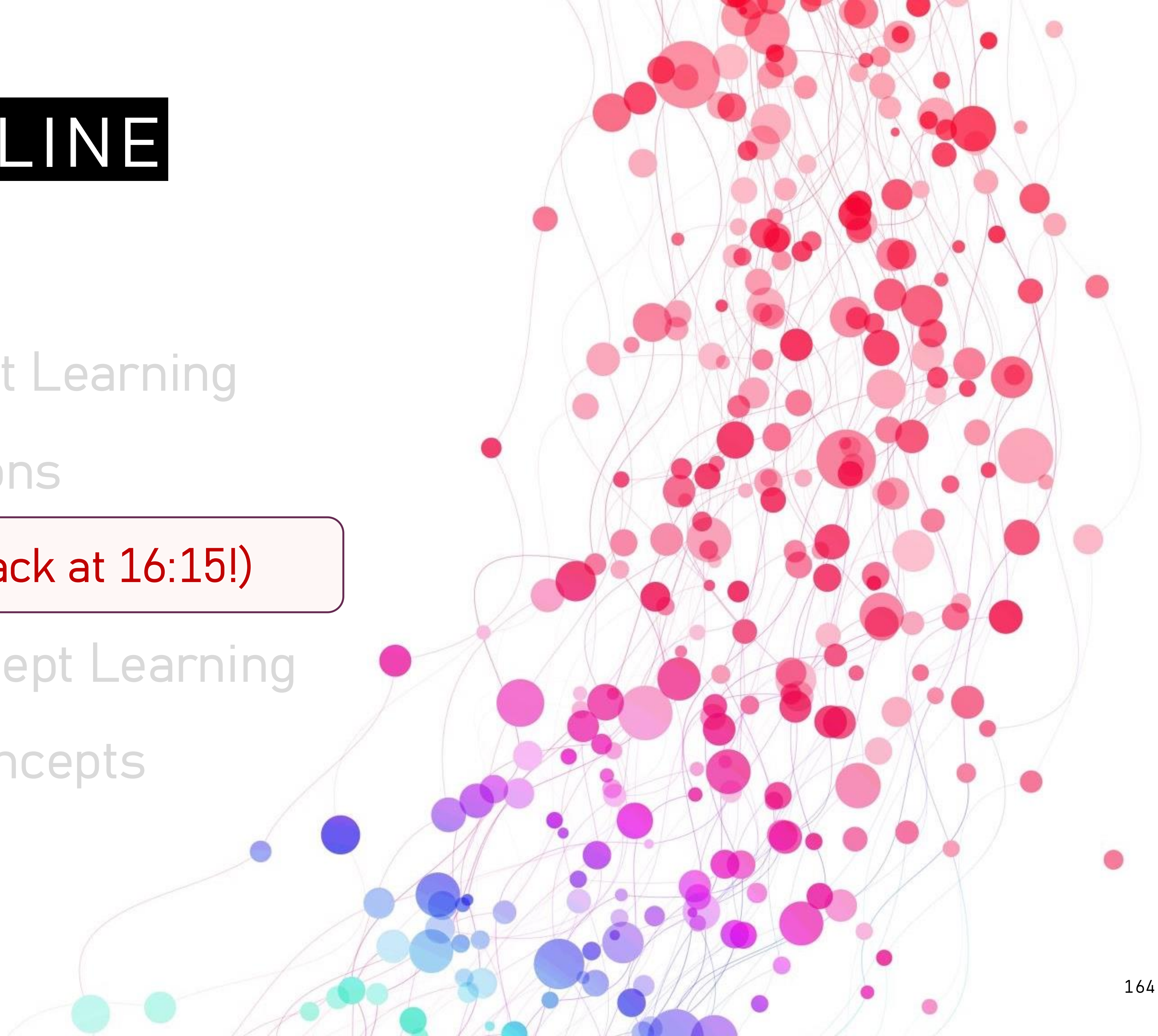


More importantly, you can fine-tune a model to be more receptive to this type of interventions by directly optimizing for an edit's positive effect

[1] Laguna, Marcinkevičs, et al. "Beyond Concept Bottleneck Models: How to Make Black Boxes Intervenable?." NeurIPS (2024).

TUTORIAL OUTLINE

1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break (Back at 16:15!)
5. Unsupervised Concept Learning
6. Reasoning With Concepts
7. Future Directions
8. Q&A



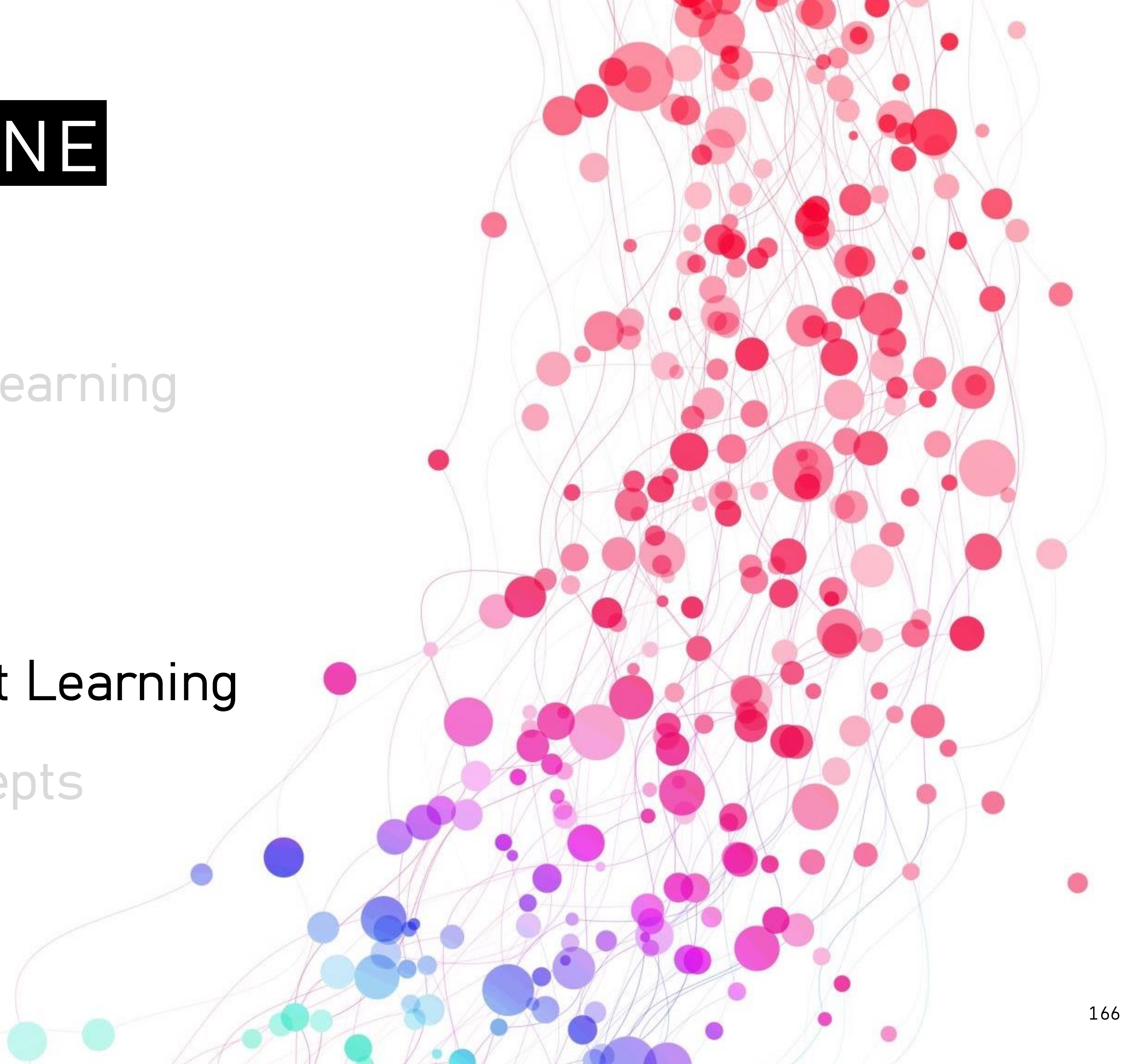
Q&A + BREAK

conceptlearning.github.io/

Back at 16:15 for part II!

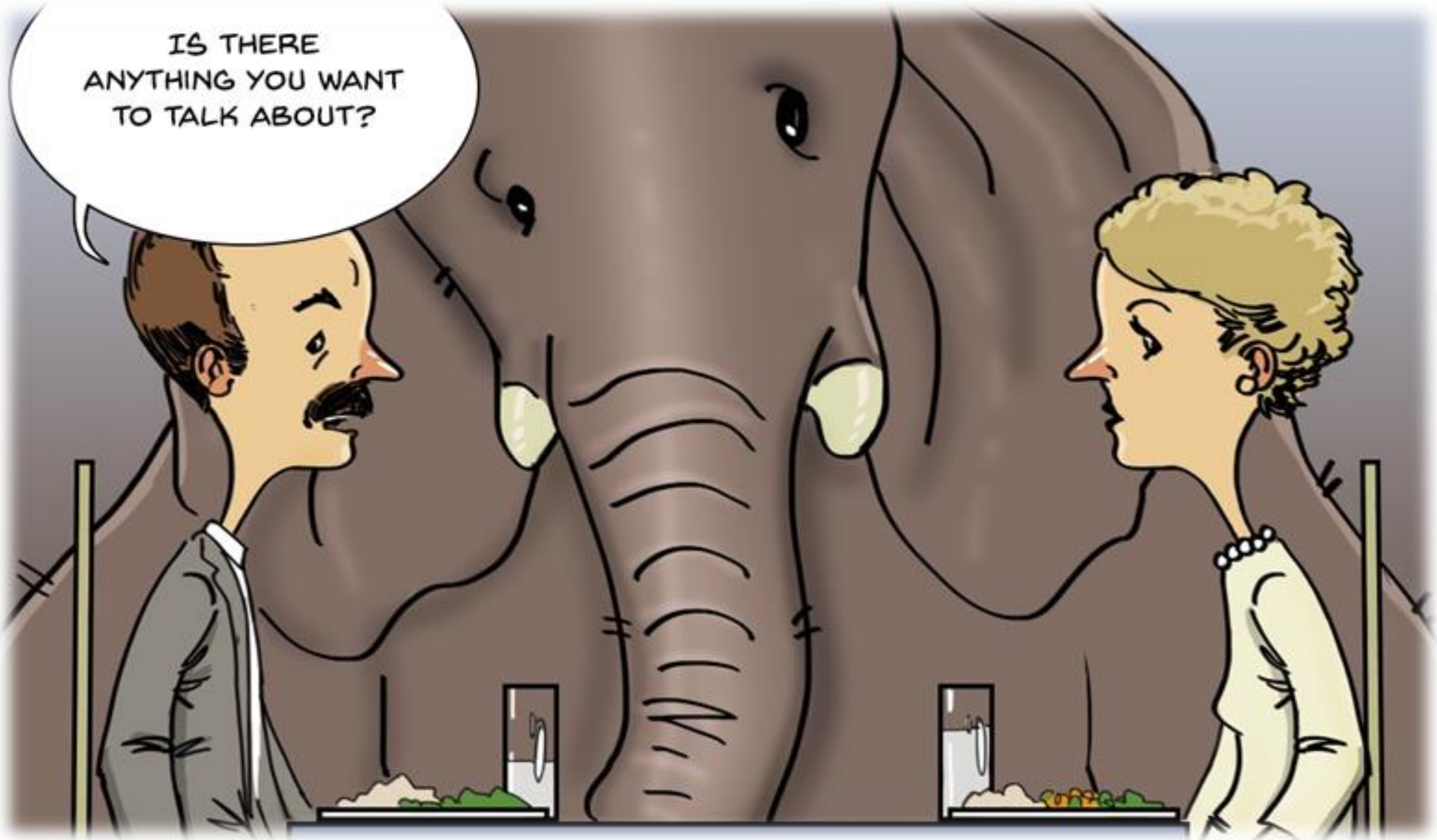
TUTORIAL OUTLINE

1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break
5. **Unsupervised Concept Learning**
6. Reasoning With Concepts
7. Future Directions
8. Q&A



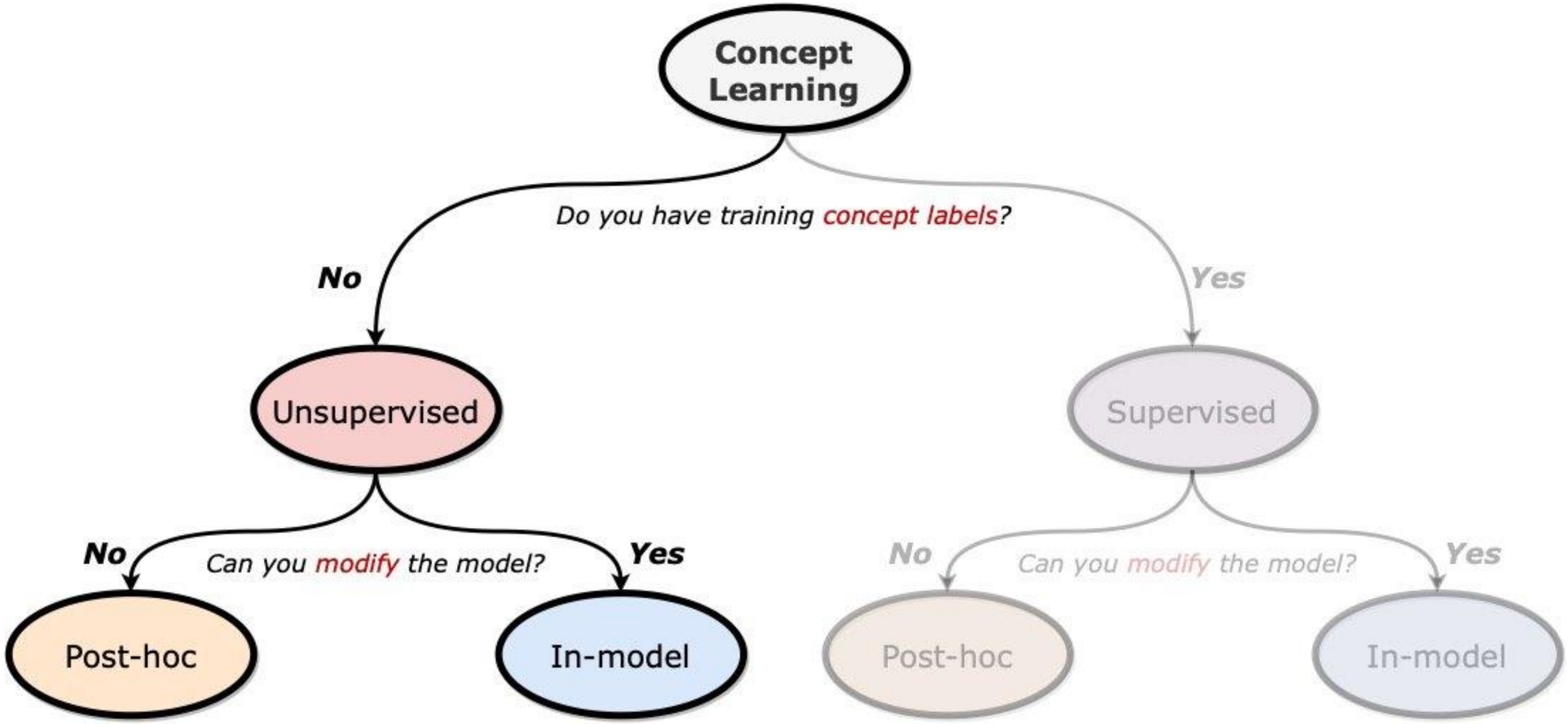
THE COST OF BEING GREAT

What if you **don't have access to concept supervisions**?



THE COST OF BEING GREAT

What if you **don't have access to concept supervisions**?



THE COST OF BEING GREAT

T-CAV requires large sets of examples of each concept of interest:

For example, when finding the influence of the concept “stripes” for a DNN, T-CAV requires a set of samples that all have the concept “stripes”

But, obtaining concept labels can be expensive and intractable

THE COST OF BEING GREAT

T-CAV requires large sets of examples of each concept of interest:

Can we extract patches automatically?

For example, when finding the influence of the concept "stripes" for a DNN, T-CAV requires a set of samples that all have the concept "stripes"

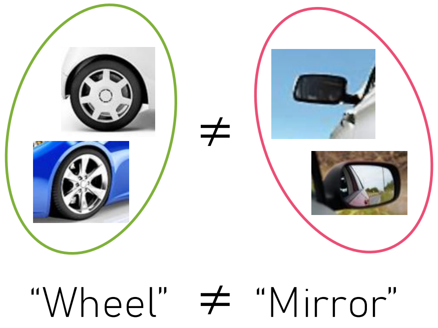
But, obtaining concept labels can be expensive and intractable

AUTOMATIC CONCEPT EXTRACTION (ACE)

Desiderata: We would like to discover concepts / patches that are:

Meaningful

Coherent

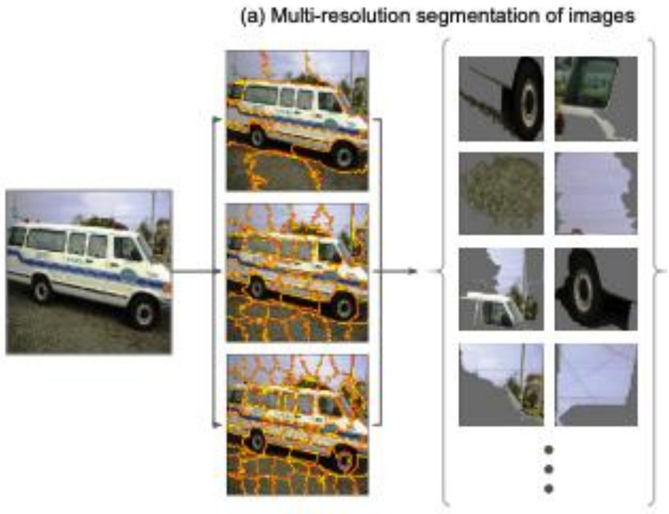


Important

[1] Ghorbani et al. "Towards automatic concept-based explanations." *Advances in Neural Information Processing Systems* 32 (2019).

AUTOMATIC CONCEPT EXTRACTION (ACE)

Proposed Solution



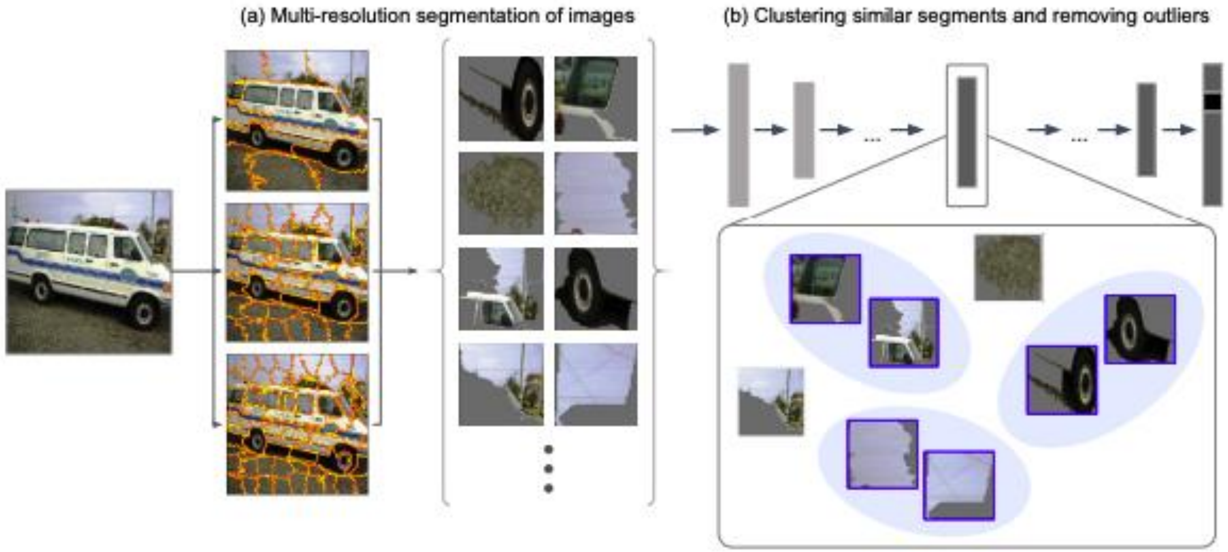
Step 1: Multi-resolution segmentation (**why?** concepts have different granularities)

Desiderata enforced: meaningfulness

[1] Ghorbani et al. "Towards automatic concept-based explanations." *Advances in Neural Information Processing Systems* 32 (2019).

AUTOMATIC CONCEPT EXTRACTION (ACE)

Proposed Solution



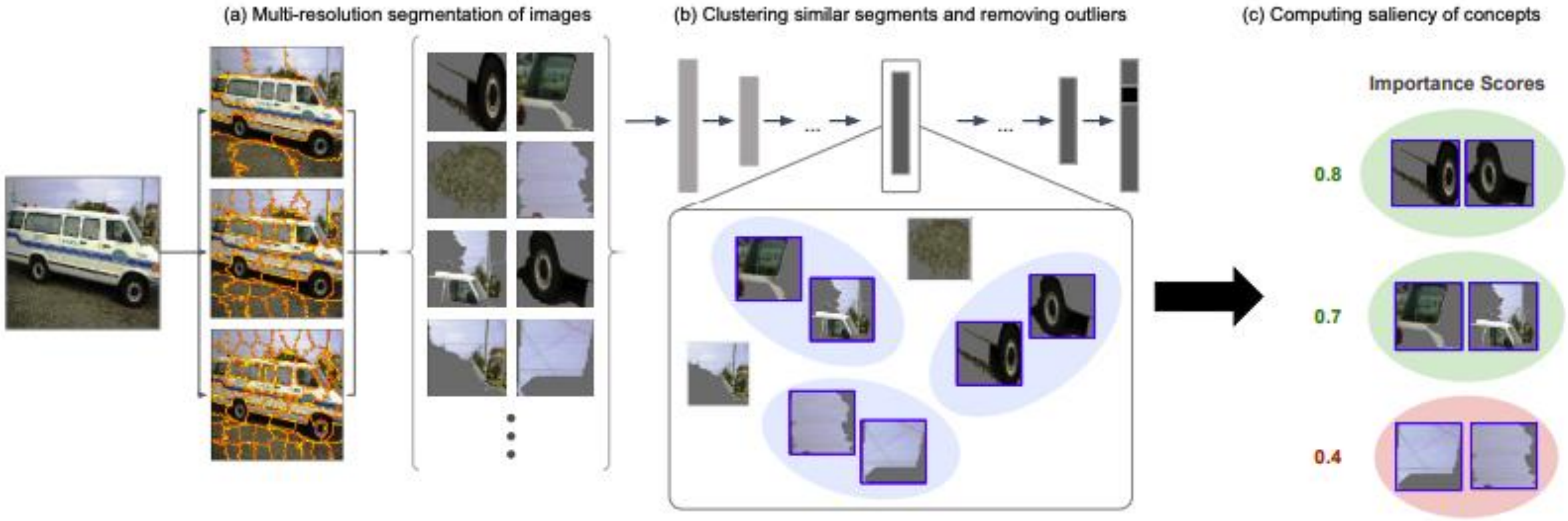
Step 2: cluster extracted segments using a hidden layer (**which one?)** of a CNN as a feature extractor (**why?** ensure invariances). Then get rid of outliers (**why?** noisy!).

Desiderata enforced: coherence

[1] Ghorbani et al. "Towards automatic concept-based explanations." *Advances in Neural Information Processing Systems* 32 (2019).

AUTOMATIC CONCEPT EXTRACTION (ACE)

Proposed Solution



Step 3: use T-CAV with the newly discovered concepts to explain the prediction of the sample of interest!

Desiderata enforced: importance

[1] Ghorbani et al. "Towards automatic concept-based explanations." *Advances in Neural Information Processing Systems* 32 (2019).

AUTOMATIC CONCEPT EXTRACTION (ACE)

What are the most salient discovered concepts for some of the ImageNet classes?

[1] Ghorbani et al. "Towards automatic concept-based explanations." *Advances in Neural Information Processing Systems* 32 (2019).

AUTOMATIC CONCEPT EXTRACTION (ACE)

What are the most salient discovered concepts for some of the ImageNet classes?

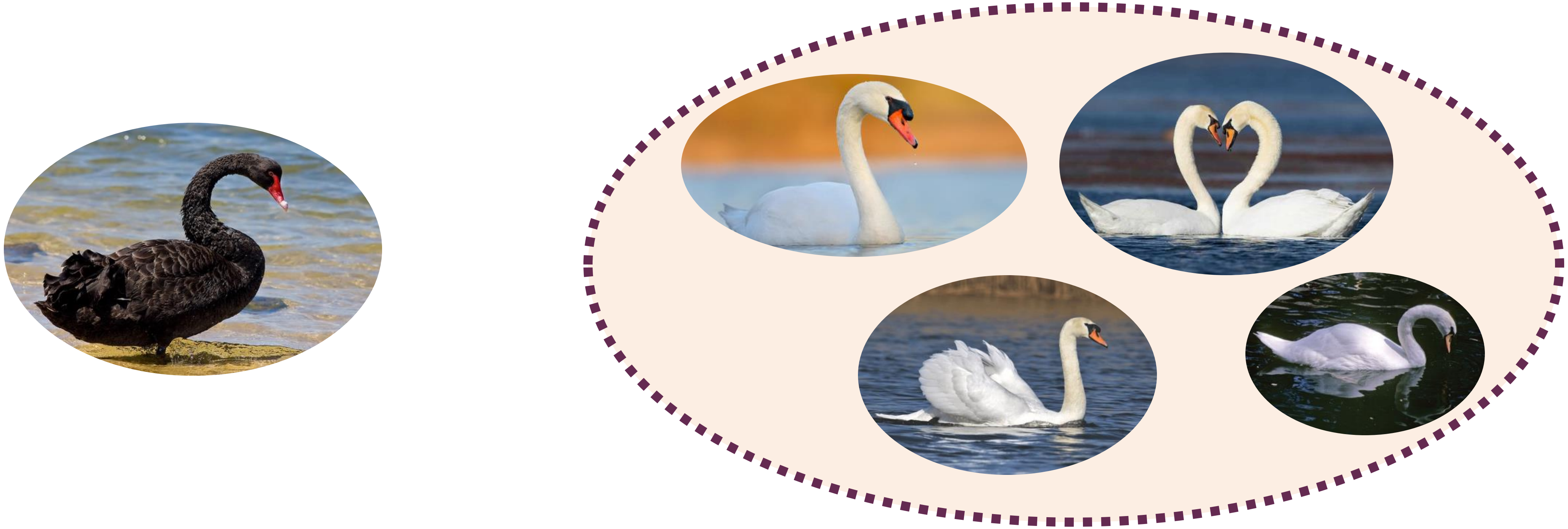
ACE has also been **generalised to learn concepts in Graph Neural Networks** in GCExplainer (Magister et al. 2021) [2]

[1] Ghorbani et al. "Towards automatic concept-based explanations." *Advances in Neural Information Processing Systems* 32 (2019).
[2] Magister et al. "GCExplainer: Human-in-the-Loop Concept-based Explanations for Graph Neural Networks." *arXiv preprint arXiv:2107.11889* (2021).

AUTOMATIC CONCEPT EXTRACTION (ACE)

ACE's hyperparameters and processing steps have several limitations:

1. We can never be certain that we properly cover all useful concepts



Important concepts for underrepresented populations could be removed as outliers!

AUTOMATIC CONCEPT EXTRACTION (ACE)

ACE's hyperparameters and processing steps have several limitations:

1. We can never be certain that we properly cover all useful concepts
2. We won't detect concepts that interact non-linearly with the output labels

Looking at the gradients provides understanding of local (linear) sensitivity

$$S_{C,k,l}(x) = \nabla h_{l,k}\big(f_l(x)\big)\,.\,v_C^l$$

AUTOMATIC CONCEPT EXTRACTION (ACE)

ACE's hyperparameters and processing steps have several limitations:

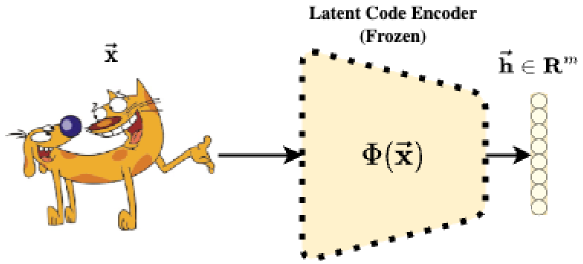
1. We can never be certain that we properly cover all useful concepts
2. We won't detect concepts that interact non-linearly with the output labels

Can we optimize accounting for concept usefulness and non-linear interactions?

COMPLETENESS-AWARE CONCEPT EXTRACTION

Proposed Solution

Step 1: project the input sample to DNN's intermediate hidden layer $\Phi(\boldsymbol{x})$

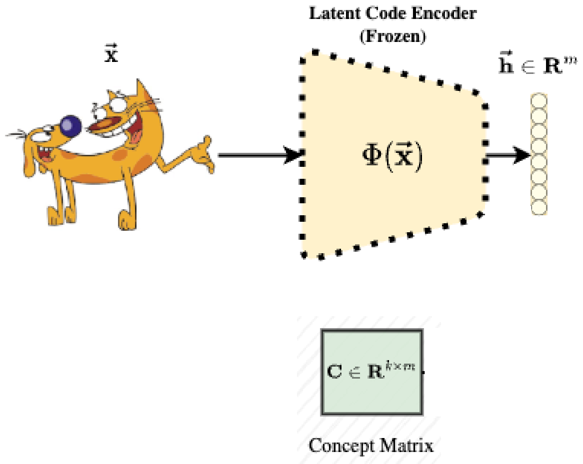


[1] Yeh et al. "On completeness-aware concept-based explanations in deep neural networks." *NeurIPS* (2020).

COMPLETENESS-AWARE CONCEPT EXTRACTION

Proposed Solution

Step 2: randomly initialize a latent, learnable concept bank of k concepts $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_k]^T$

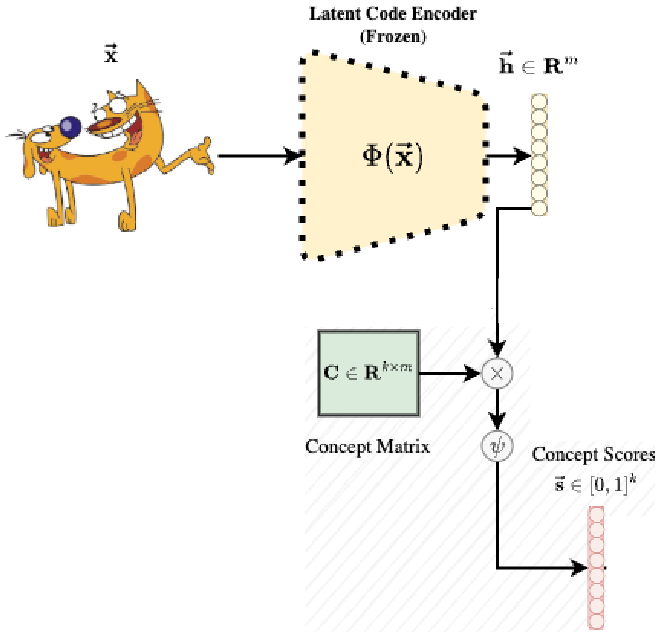


[1] Yeh et al. "On completeness-aware concept-based explanations in deep neural networks." *NeurIPS* (2020).

COMPLETENESS-AWARE CONCEPT EXTRACTION

Proposed Solution

Step 3: compute a set of concept scores by projecting the input embedding into the concept space

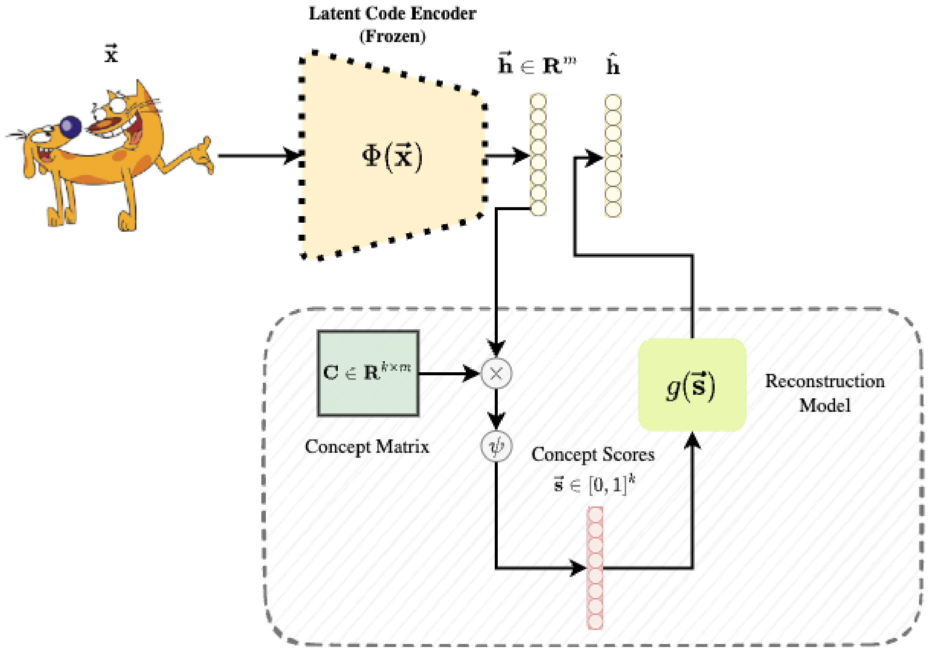


[1] Yeh et al. "On completeness-aware concept-based explanations in deep neural networks." *NeurIPS* (2020).

COMPLETENESS-AWARE CONCEPT EXTRACTION

Proposed Solution

Step 4: pass the concepts scores to a learnable model $g(\vec{s}) = \hat{h}$ that aims to reconstruct $\vec{h}$ from $\vec{s}$

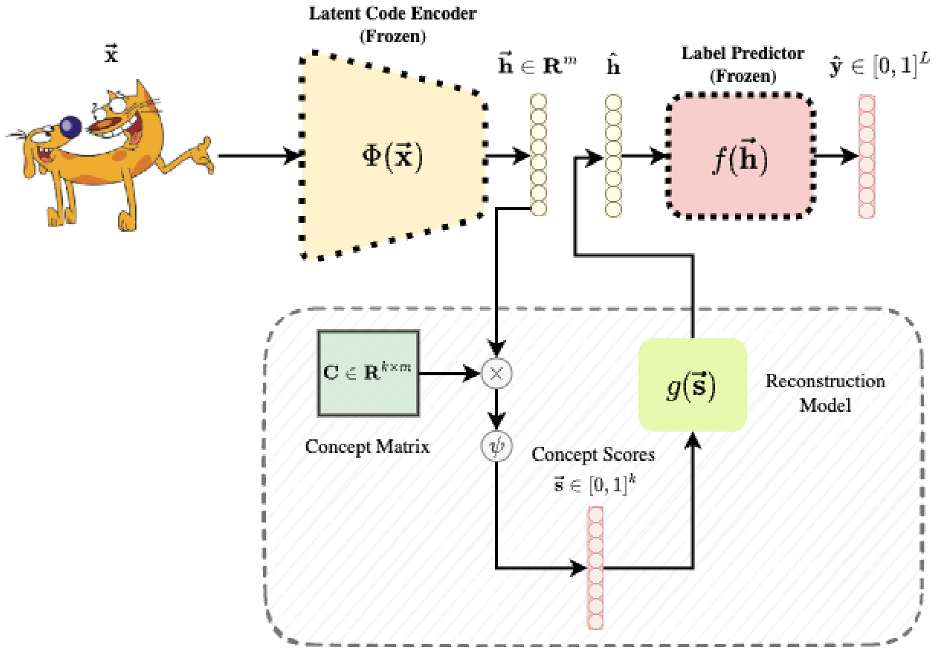


[1] Yeh et al. "On completeness-aware concept-based explanations in deep neural networks." *NeurIPS* (2020).

COMPLETENESS-AWARE CONCEPT EXTRACTION

Proposed Solution

Step 5: use $\widehat{\boldsymbol{h}}$ as the reconstructed hidden layer and predict an output class using f



[1] Yeh et al. "On completeness-aware concept-based explanations in deep neural networks." *NeurIPS* (2020).

COMPLETENESS-AWARE CONCEPT EXTRACTION

Proposed Solution

Step 6: maximise a "*concept completeness score*"

DNN's accuracy via concept projection

$$n_f(\boldsymbol{c}_1, \dots, \boldsymbol{c}_m) = \frac{\sup\limits_{g} \mathbb{P}_{x,y\sim V}\left[y = \underset{y'}{\operatorname{argmax}}\, f_{y'}\left(g\left(\boldsymbol{C}\, \phi(x)\right)\right)\right] - a_r}{\mathbb{P}_{x,y\sim V}\left[y = \underset{y'}{\operatorname{argmax}}\, f_{y'}(\boldsymbol{x})\right] - a_r}$$

Original DNN's accuracy

Score is ~ 1 if and only if the projection in the concept space preserves all the information needed to predict y!

[1] Yeh et al. "On completeness-aware concept-based explanations in deep neural networks." *NeurIPS* (2020).

COMPLETENESS-AWARE CONCEPT EXTRACTION

CCE further encourages discovered concepts to be:

1. **Coherent**: similar samples should remain close in concept-space
2. **Diverse**: concept vectors should be as distinct from each other as possible

$$R(\mathbf{c}) = \lambda_1 \overset{\text{Coherency}}{\frac{\sum_{k=1}^{m} \sum_{\mathbf{x}_a^b \in T_{\mathbf{c}_k}} \Phi(\mathbf{x}_a^b) \cdot \mathbf{c}_k}{mK}} - \lambda_2 \overset{\text{Diversity}}{\frac{\sum_{j \neq k} \mathbf{c}_j \cdot \mathbf{c}_k}{m(m-1)}}$$

[1] Yeh et al. "On completeness-aware concept-based explanations in deep neural networks." *NeurIPS* (2020).

COMPLETENESS-AWARE CONCEPT EXTRACTION

And it can be applied to different data modalities!

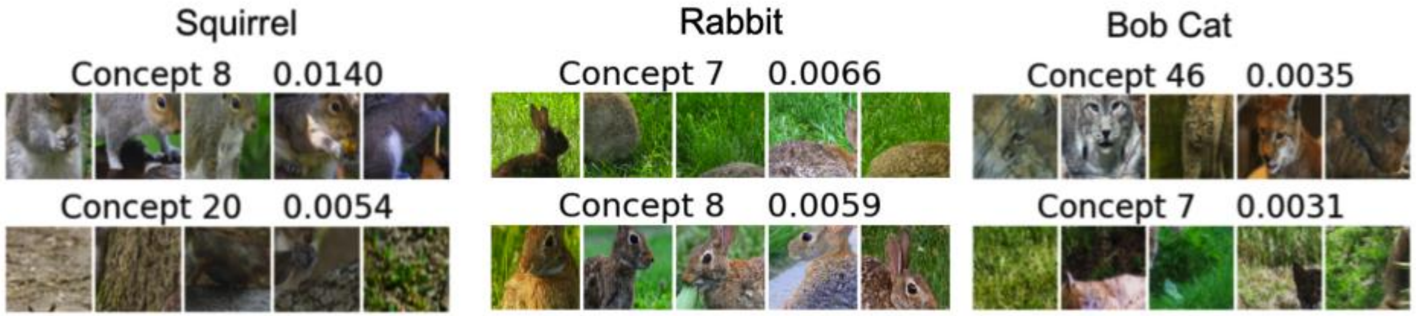
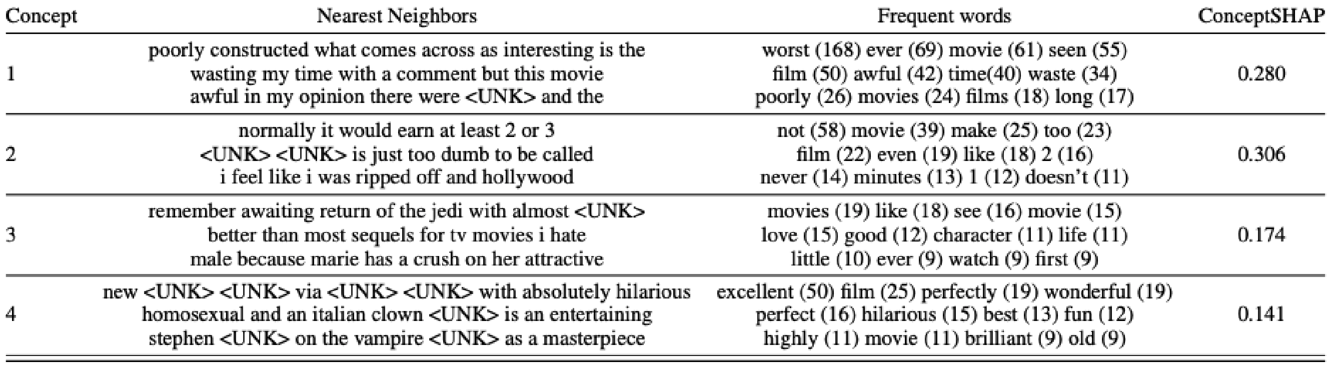


Table 2: The 4 discovered concepts and some nearest neighbors along with the most frequent words that appear in top-500 nearest neighbors.



Concept	Nearest Neighbors	Frequent words	ConceptSHAP
1	poorly constructed what comes across as interesting is the wasting my time with a comment but this movie awful in my opinion there were <UNK> and the	worst (168) ever (69) movie (61) seen (55) film (50) awful (42) time(40) waste (34) poorly (26) movies (24) films (18) long (17)	0.280
2	normally it would earn at least 2 or 3 <UNK> <UNK> is just too dumb to be called i feel like i was ripped off and hollywood	not (58) movie (39) make (25) too (23) film (22) even (19) like (18) 2 (16) never (14) minutes (13) 1 (12) doesn’t (11)	0.306
3	remember awaiting return of the jedi with almost <UNK> better than most sequels for tv movies i hate male because marie has a crush on her attractive	movies (19) like (18) see (16) movie (15) love (15) good (12) character (11) life (11) little (10) ever (9) watch (9) first (9)	0.174
4	new <UNK> <UNK> via <UNK> <UNK> with absolutely hilarious homosexual and an italian clown <UNK> is an entertaining stephen <UNK> on the vampire <UNK> as a masterpiece	excellent (50) film (25) perfectly (19) wonderful (19) perfect (16) hilarious (15) best (13) fun (12) highly (11) movie (11) brilliant (9) old (9)	0.141

[1] Yeh et al. "On completeness-aware concept-based explanations in deep neural networks." *NeurIPS* (2020).

REMEMBER CONCEPT BOTTLENECKS?

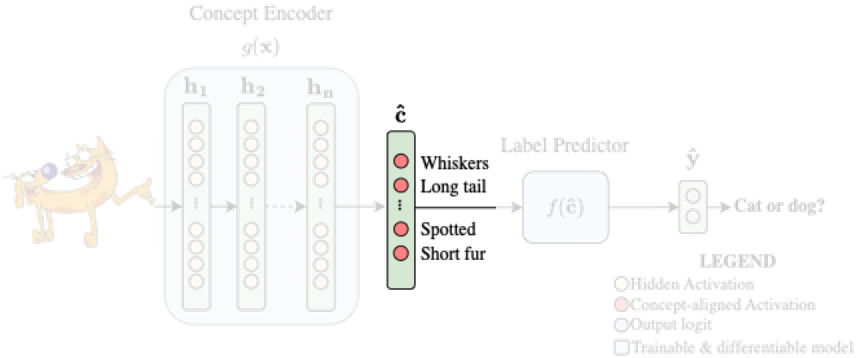
We took care of T-CAV & friends…

What about CBMs family?

LABEL-FREE CONCEPT BOTTLENECKS

Limitation Being Addressed

CBMs & co **require some known concepts**, or we have no bottleneck at all!



And post-hoc CBMs still **require one to know which concepts are potentially useful for a downstream task**!

ICLR23

CVPR23

[1] Oikarinen et al. "Label-Free Concept Bottleneck Models." ICLR (2023).
[2] Yang et al. "Language in a bottle: Language model guided concept bottlenecks for interpretable image classification." CVPR (2023).

LABEL-FREE CONCEPT BOTTLENECKS

Proposed Solution

Why not simply ask GPT for a set of useful concepts for a specific class?

“List the most important features for recognizing something as a {class}:”

ICLR23

CVPR23

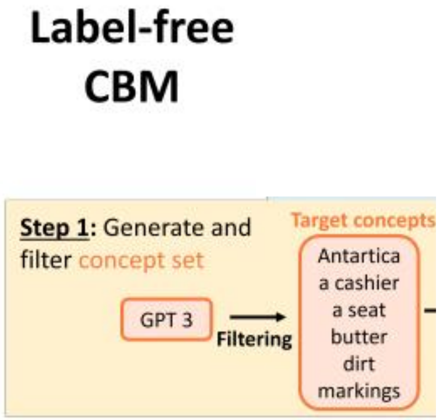
---

[1] Oikarinen et al. "Label-Free Concept Bottleneck Models." ICLR (2023).
[2] Yang et al. "Language in a bottle: Language model guided concept bottlenecks for interpretable image classification." CVPR (2023).

LABEL-FREE CONCEPT BOTTLENECKS

Proposed Solution

Step 1: Generate a concept set by "asking" an LLM



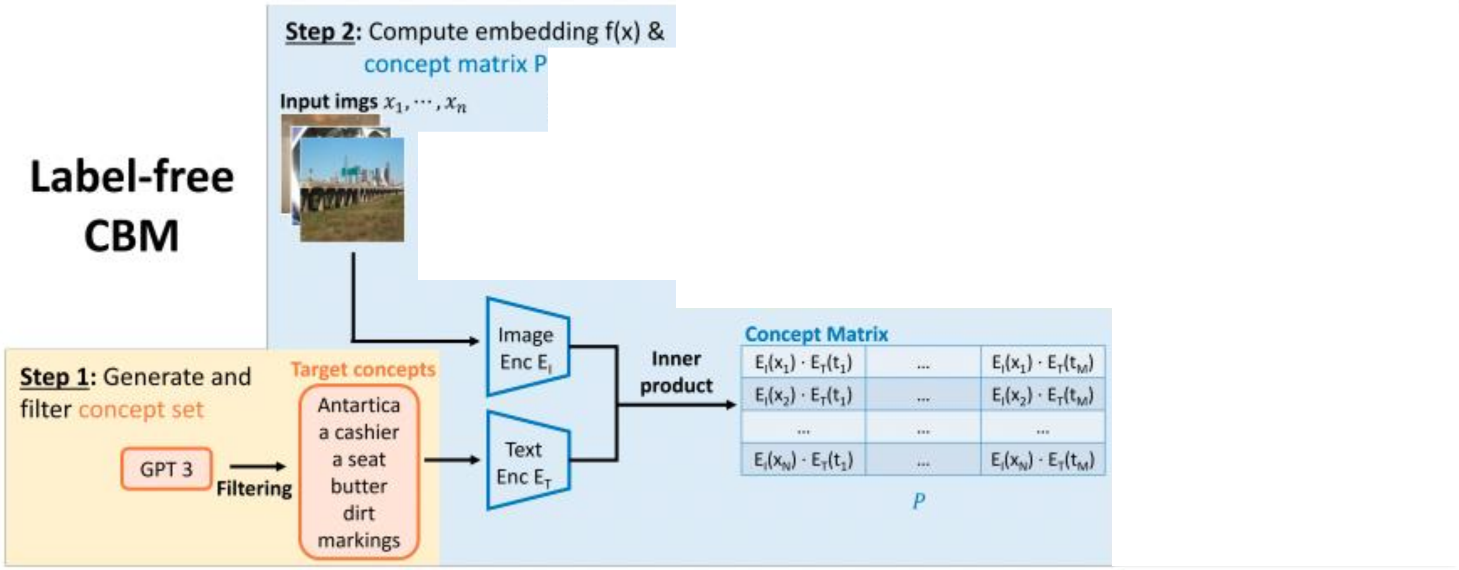
ICLR23 CVPR23

[1] Oikarinen et al. "Label-Free Concept Bottleneck Models." ICLR (2023).
[2] Yang et al. "Language in a bottle: Language model guided concept bottlenecks for interpretable image classification." CVPR (2023).

LABEL-FREE CONCEPT BOTTLENECKS

Proposed Solution

Step 2: Use multi-modal contrastive language model (e.g., CLIP) to compute similarity of image-text embeddings



$E_I(x_1) \cdot E_T(t_1)$	...	$E_I(x_1) \cdot E_T(t_M)$
$E_I(x_2) \cdot E_T(t_1)$	...	$E_I(x_2) \cdot E_T(t_M)$
...	...	...
$E_I(x_N) \cdot E_T(t_1)$	...	$E_I(x_N) \cdot E_T(t_M)$

ICLR23

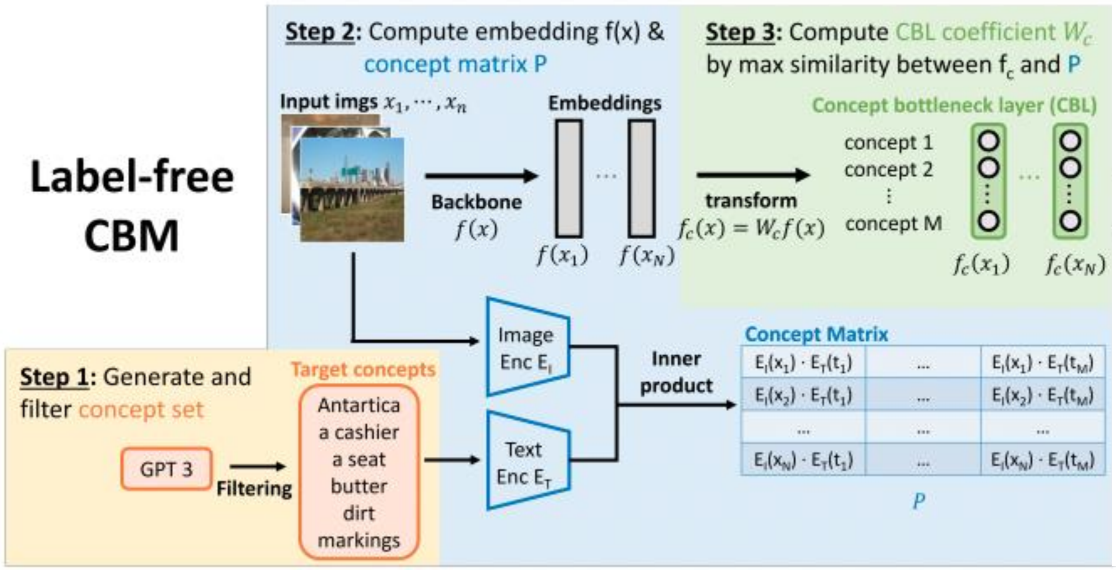
CVPR23

[1] Oikarinen et al. "Label-Free Concept Bottleneck Models." ICLR (2023).
[2] Yang et al. "Language in a bottle: Language model guided concept bottlenecks for interpretable image classification." CVPR (2023).

LABEL-FREE CONCEPT BOTTLENECKS

Proposed Solution

Step 3: Train DNN activations to align with similarity scores predicted by the contrastive LM



$E_I(x_1)\cdot E_T(t_1)$	...	$E_I(x_1)\cdot E_T(t_M)$
$E_I(x_2)\cdot E_T(t_1)$	...	$E_I(x_2)\cdot E_T(t_M)$
...	...	...
$E_I(x_N)\cdot E_T(t_1)$	...	$E_I(x_N)\cdot E_T(t_M)$

ICLR23 CVPR23

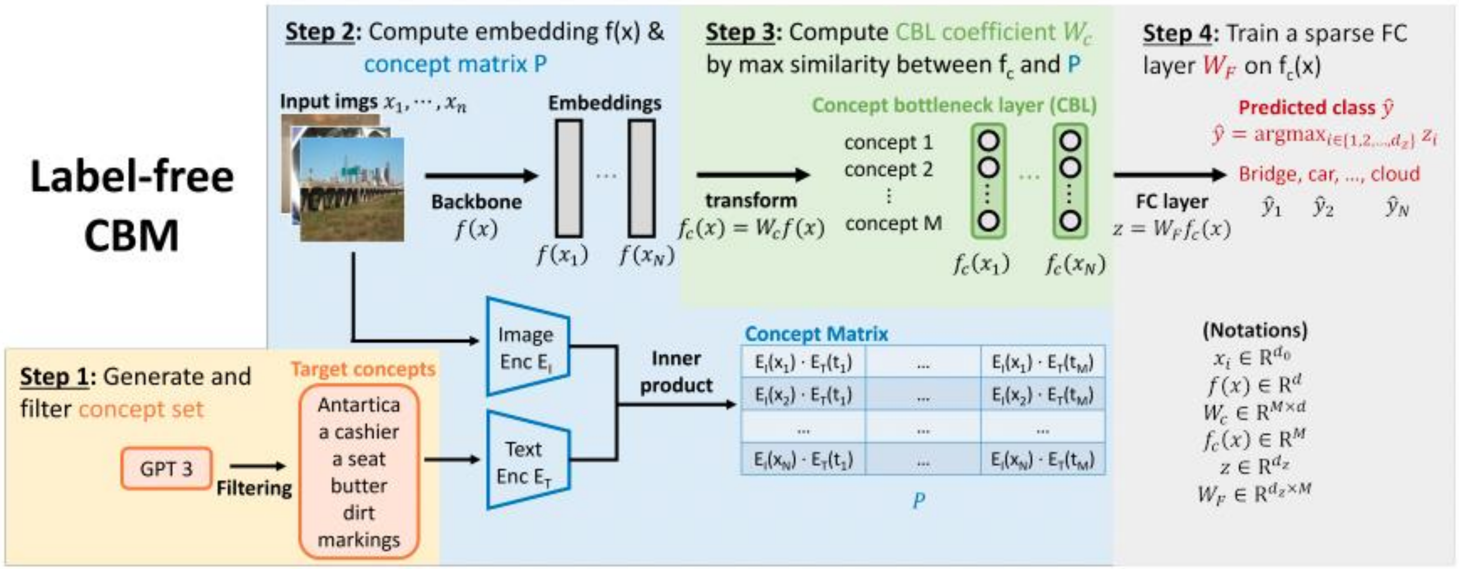
[1] Oikarinen et al. "Label-Free Concept Bottleneck Models." ICLR (2023).

[2] Yang et al. "Language in a bottle: Language model guided concept bottlenecks for interpretable image classification." CVPR (2023).

LABEL-FREE CONCEPT BOTTLENECKS

Proposed Solution

Step 4: Train a simple (linear) model to map predicted concept scores to tasks



$E_I(x_1) \cdot E_T(t_1)$	...	$E_I(x_1) \cdot E_T(t_M)$
$E_I(x_2) \cdot E_T(t_1)$	...	$E_I(x_2) \cdot E_T(t_M)$
...	...	...
$E_I(x_N) \cdot E_T(t_1)$	...	$E_I(x_N) \cdot E_T(t_M)$

ICLR23

CVPR23

[1] Oikarinen et al. "Label-Free Concept Bottleneck Models." ICLR (2023).
[2] Yang et al. "Language in a bottle: Language model guided concept bottlenecks for interpretable image classification." CVPR (2023).

STRIPPING CBMS TO THEIR BONES

What if we want a CBM, but… we **don't have**:

- Concept supervisions
- Pre-trained contrastive LMs

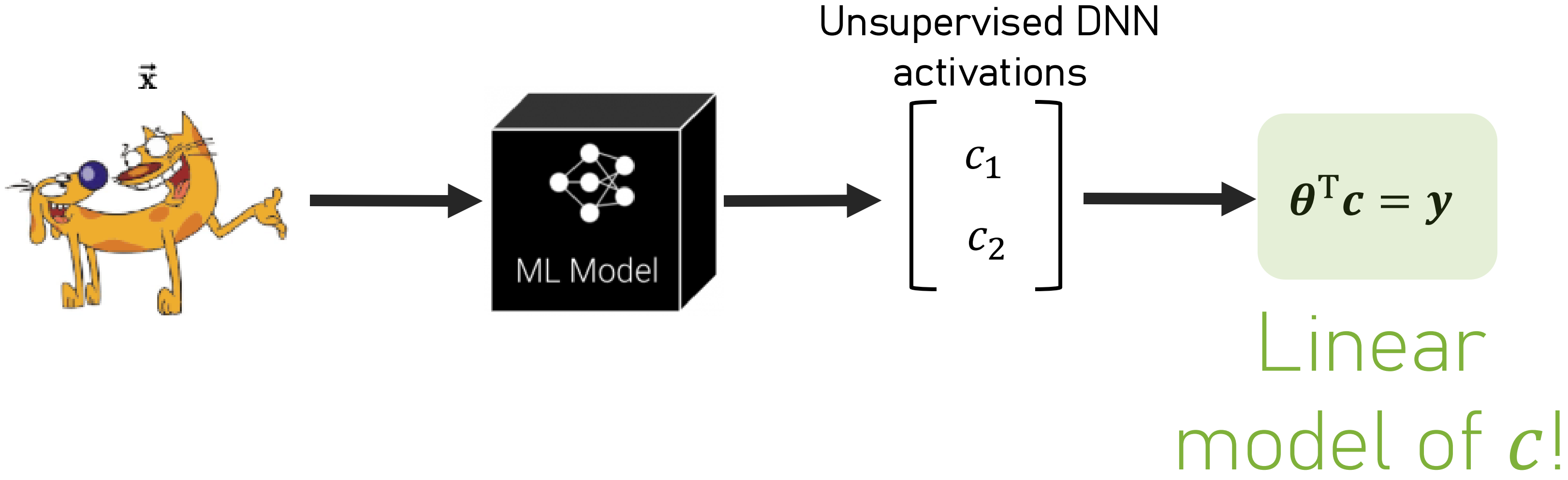
What's left??

STRIPPING CBMS TO THEIR BONES

What if we want a CBM, but… we don't have:

- Concept supervisions
- Pre-trained contrastive LMs

What's left??



STRIPPING CBMS TO THEIR BONES

What if we want a CBM, but… we don’t have:

- Concept supervisions
- Pre-trained contrastive LMs

What’s left??

Can we make a DNN behave like a proper linear model?

LINEARIZING A DNN

If we want to make a DNN **act as a linear model while maintaining its expressive power**, we need a few things:

LINEARIZING A DNN

If we want to make a DNN act as a linear model while maintaining its expressive power, we need a few things:

1. [*Expressiveness*] The relevance weights used to make the output prediction must be able to dynamically adapt depending on the input:

Linear Model Output: $f(\vec{x}) = \theta^T \vec{x}$

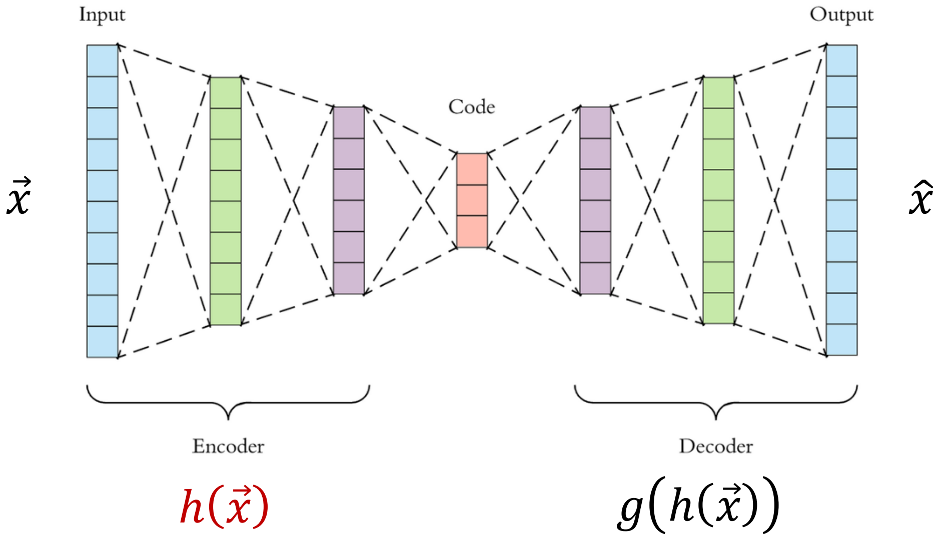
"Linear-ish DNN" Model output: $f(\vec{x}) = \theta(\vec{x})^T \vec{x}$

where $\theta : \mathcal{X} \rightarrow \mathcal{W}$ is parameterised as a learnable DNN!

LINEARIZING A DNN

If we want to make a DNN act as a linear model while maintaining its expressive power, we need a few things:

2. [*Interpretability*] If the features are not interpretable (e.g., individual pixels), then we should learn a high-level “concept” representation $h(\vec{x})$:



LINEARIZING A DNN

If we want to make a DNN **act as a linear model while maintaining its expressive power**, we need a few things:

2. [*Interpretability*] If the features are not interpretable (e.g., individual pixels), then **we should learn a high-level "concept" representation** $h(\vec{x})$:

Linear Model Output: $f(\vec{x}) = \theta^T \vec{x}$

"Linear-ish DNN" Model output: $f(\vec{x}) = \theta(\vec{x})^T h(\vec{x})$

where $\theta: \mathcal{X} \to \mathcal{W}$ and $h: \mathcal{X} \to \mathcal{Z}$ are parameterised as a learnable DNNs!

LINEARIZING A DNN

If we want to make a DNN **act as a linear model while maintaining its expressive power**, we need a few things:

3. [*Local Linearity*] The model should behave, at least in the neighborhood of a sample, as a linear classifier.

What does this imply?

LINEARIZING A DNN

If we want to make a DNN act as a linear model while maintaining its expressive power, we need a few things:

3. [*Local Linearity*] The model should behave, at least in the neighborhood of a sample, as a linear classifier.

$$\nabla_{h(\vec{x})} f(\vec{x}) \approx \theta(\vec{x})$$

Relevance coefficients adapt with the inputs
but they do so in a stable/slow manner

LINEARIZING A DNN

If we want to make a DNN act as a linear model while maintaining its expressive power, we need a few things:

3. [*Local Linearity*] The model should behave, at least in the neighborhood of a sample, as a linear classifier.

$$\nabla_{h(\vec{x})} f(\vec{x}) \approx \theta(\vec{x})$$

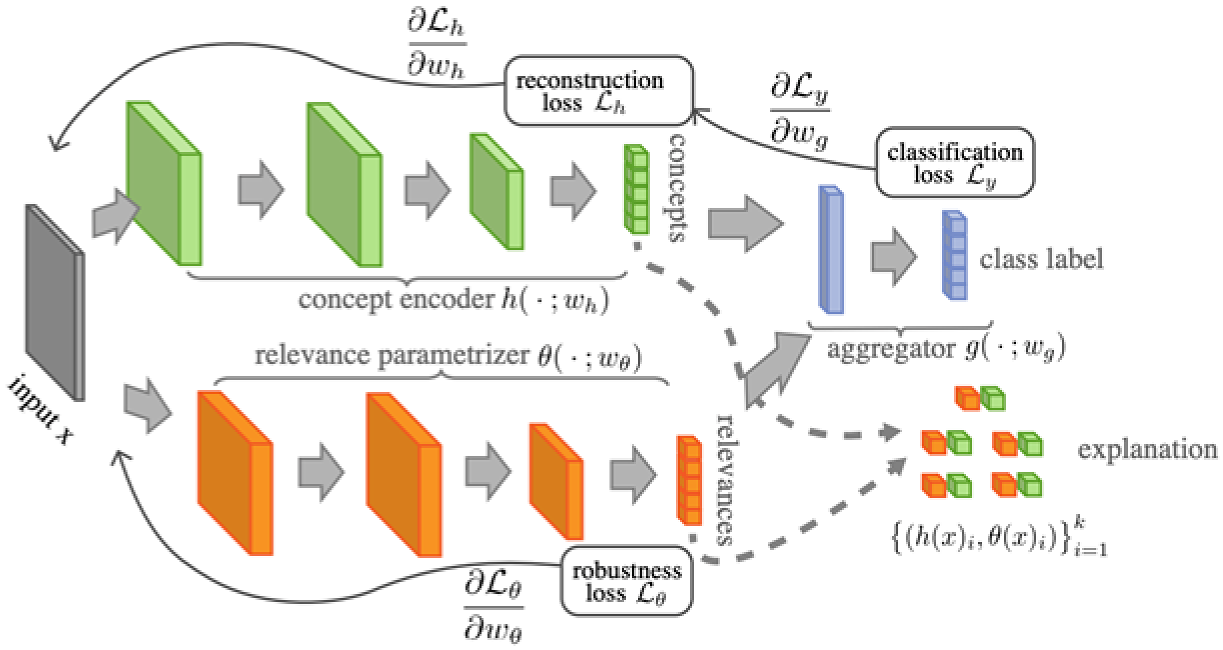
Relevance coefficients adapt with the inputs but they do so in a stable/slow manner

We can encourage this local linearity by including the following training regulariser:

$$\mathcal{L}_{reg}(\vec{x}) \coloneqq \left\| \nabla_{\vec{x}} f(\vec{x}) \; - J_{\vec{x}}^{h(\vec{x})}(\vec{x}) \right\|$$

SELF-EXPLAINING NEURAL NETS

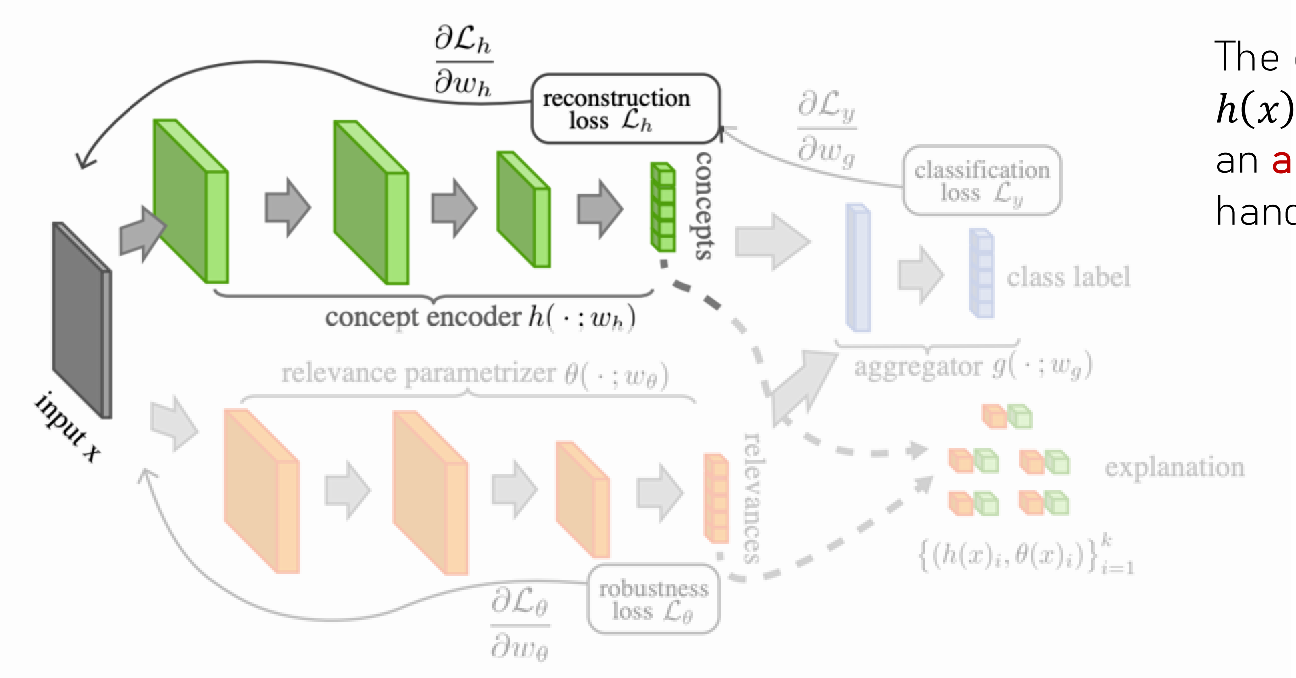
This is the idea behind Self Explaining Neural Networks (SENNs)!



[1] Alvarez Melis et al. "Towards robust interpretability with self-explaining neural networks." *NeurIPS* (2018).

SELF-EXPLAINING NEURAL NETS

Step 1: extract concepts from our input distribution:

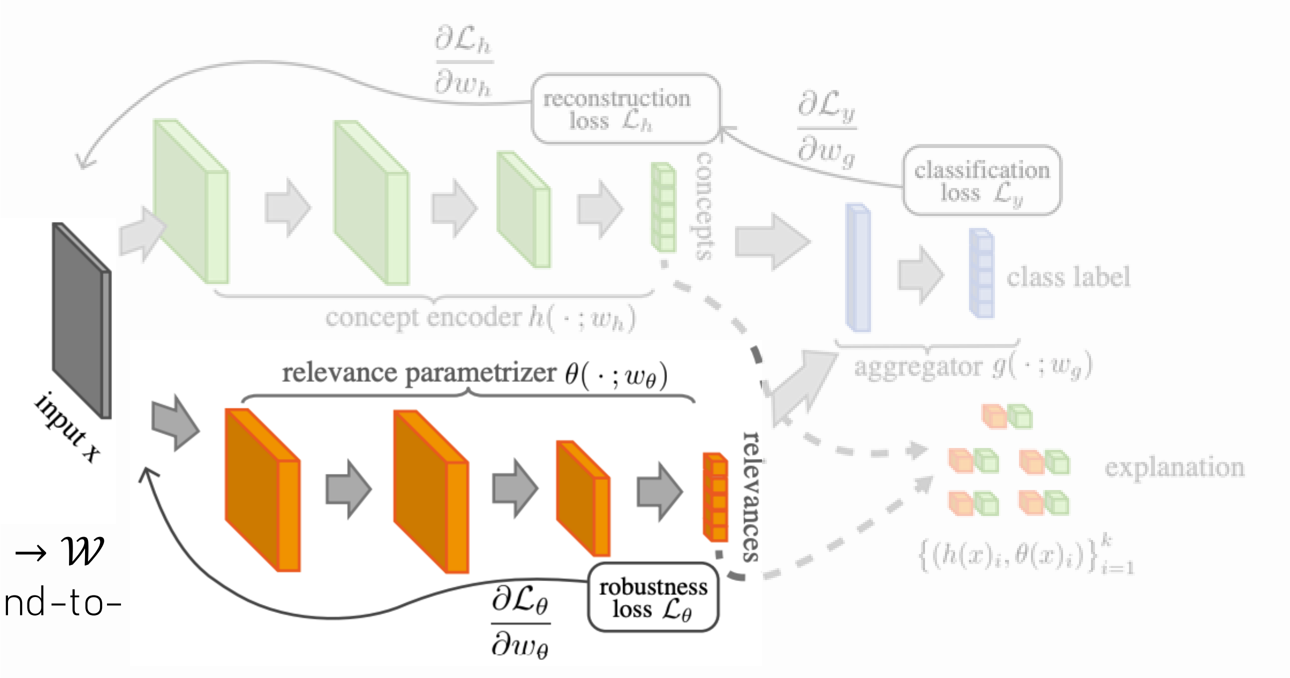


The concept extractor $h(x): \mathcal{X} \to \mathcal{Z}$ can be learnt via an autoencoder model or via handcrafted feature extractors

[1] Alvarez Melis et al. "Towards robust interpretability with self-explaining neural networks." *NeurIPS* (2018).

SELF-EXPLAINING NEURAL NETS

Step 2: use DNN to dynamically predict the set of linear weights for each sample:



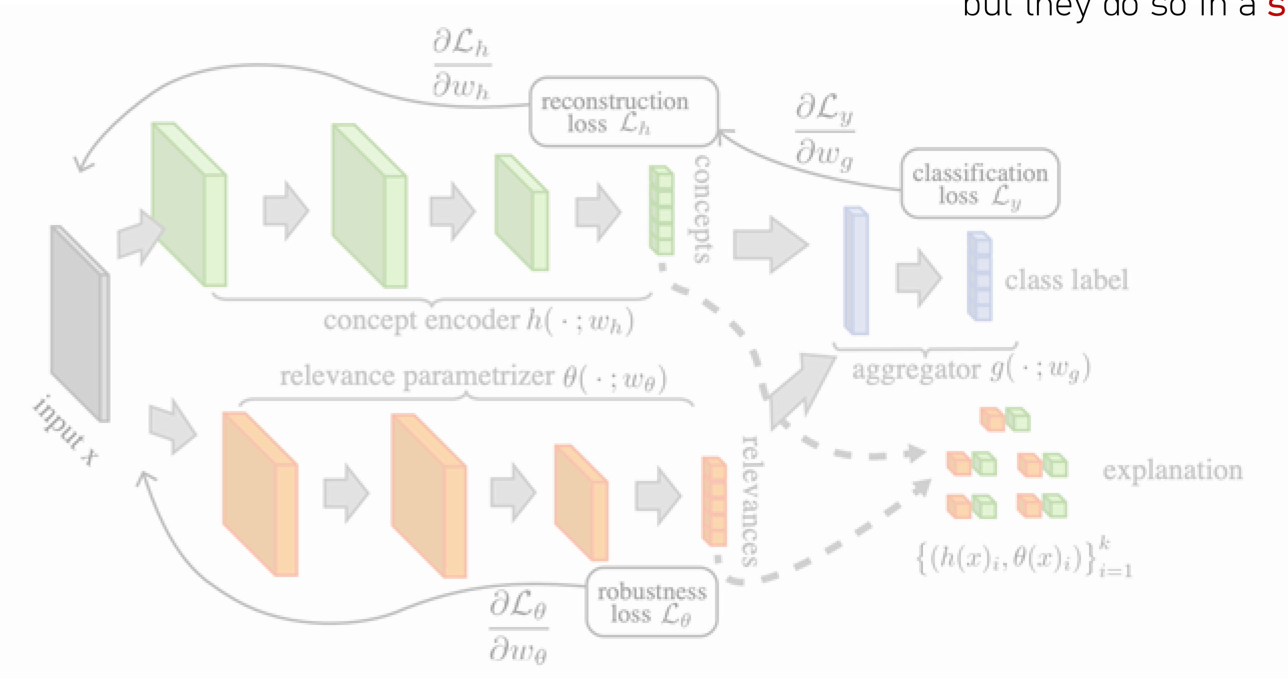
This is done via a weight relevance model $\theta(x): \mathcal{X} \to \mathcal{W}$ that can be learnt in an end-to-end fashion

[1] Alvarez Melis et al. "Towards robust interpretability with self-explaining neural networks." *NeurIPS* (2018).

SELF-EXPLAINING NEURAL NETS

Step 3: Add regulariser that will encourage local linearity: $\mathcal{L}_\theta(f(x)) := \|\nabla_x f(x) - \theta(x)^\top J_x^h(x)\|$

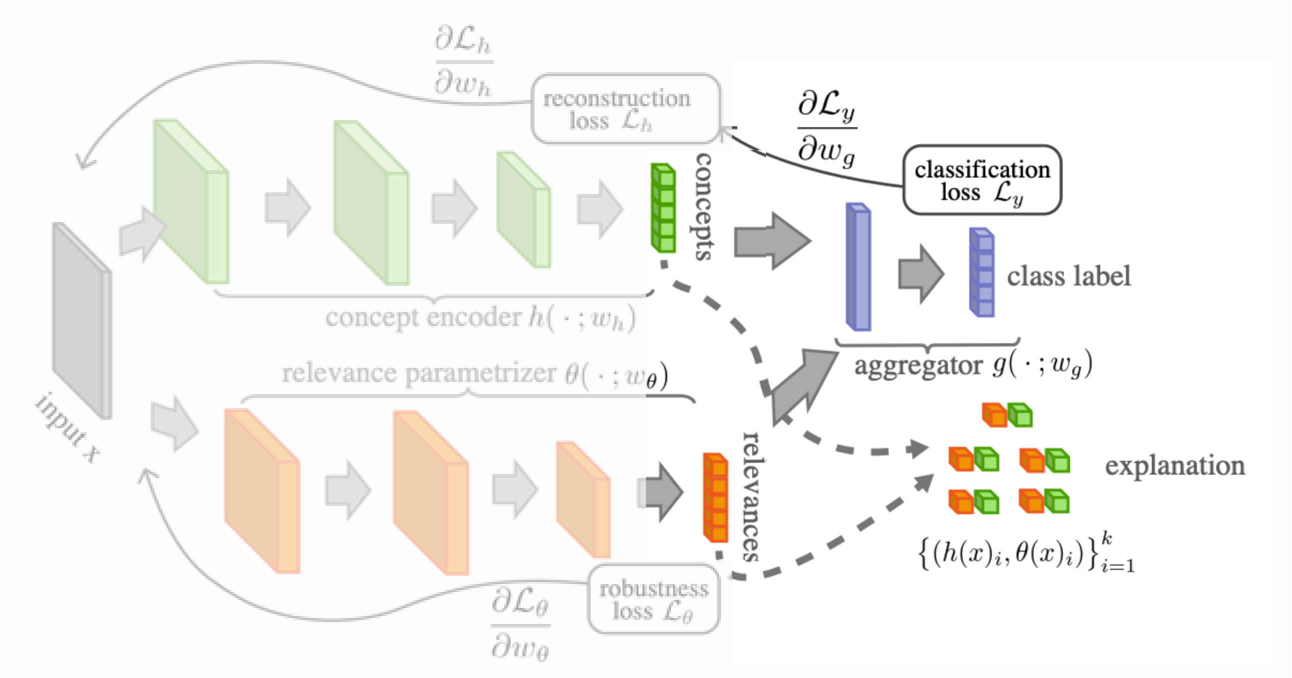
Relevance coefficients adapt with the inputs
but they do so in a stable/slow manner



[1] Alvarez Melis et al. "Towards robust interpretability with self-explaining neural networks." *NeurIPS* (2018).

SELF-EXPLAINING NEURAL NETS

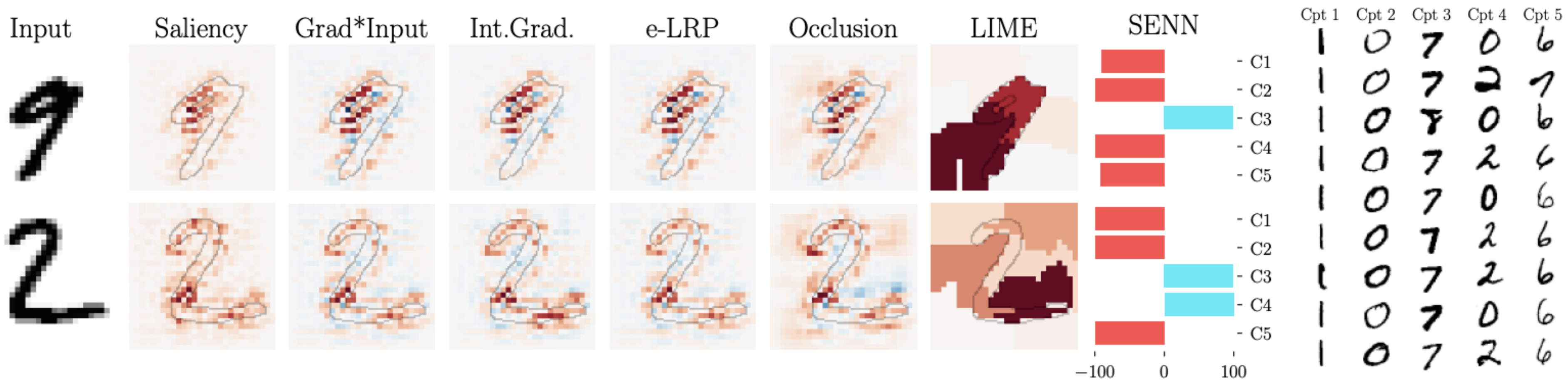
Step 4: Generate prediction with the linear form $\theta(x)^T h(x)$. The explanation is the tuple (concept, relevance weight)



[1] Alvarez Melis et al. "Towards robust interpretability with self-explaining neural networks." *NeurIPS* (2018).

SELF-EXPLAINING NEURAL NETS

When features lack useful semantics, learnt concepts can be understood via prototypical examples:



[1] Alvarez Melis et al. "Towards robust interpretability with self-explaining neural networks." *NeurIPS* (2018).

TUTORIAL OUTLINE

1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break
5. Unsupervised Concept Learning
6. **Reasoning With Concepts**
7. Future Directions
8. Q&A

CONCEPT-BASED REASONING

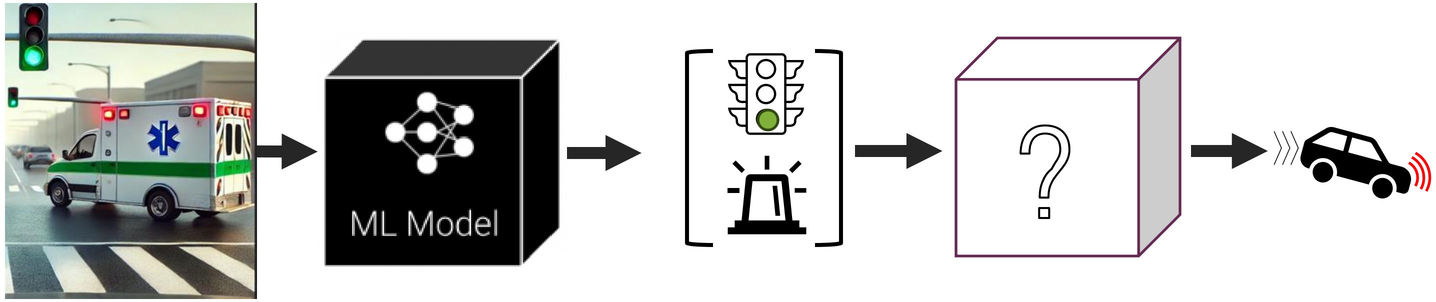
We'll focus on two main branches of concept-based reasoning:

Neural symbolic concept reasoning

Causal concept reasoning

TIME TO GET YOUR C*EPTS TOGETHER

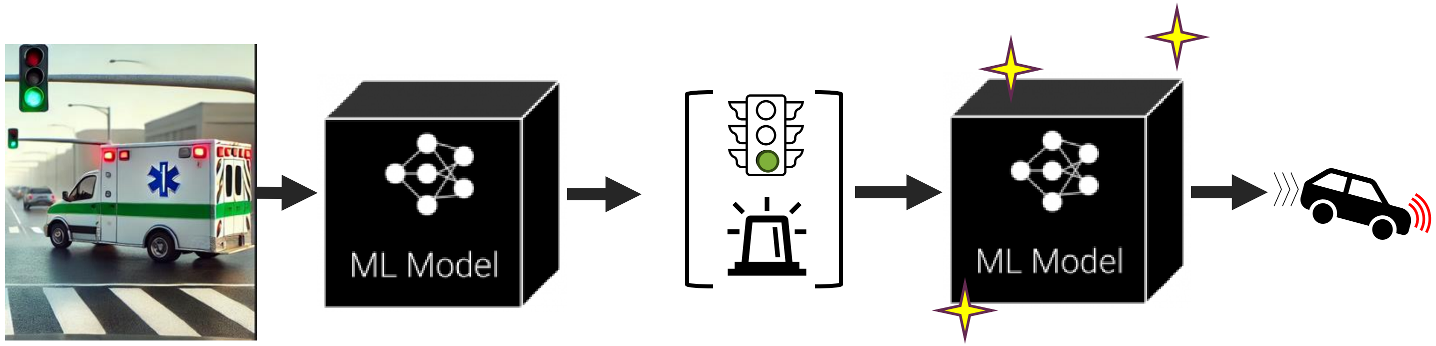
Let's say we have a nice set of concepts, what should we use as classification head?



TIME TO GET YOUR C*EPTS TOGETHER

Let's say we have a nice set of concepts, what should we use as classification head?

… what about an opaque DNN?



Back to square 1!

TIME TO GET YOUR C*EPTS TOGETHER

Let's say we have a nice set of concepts, what should we use as classification head?

… what about an opaque DNN?

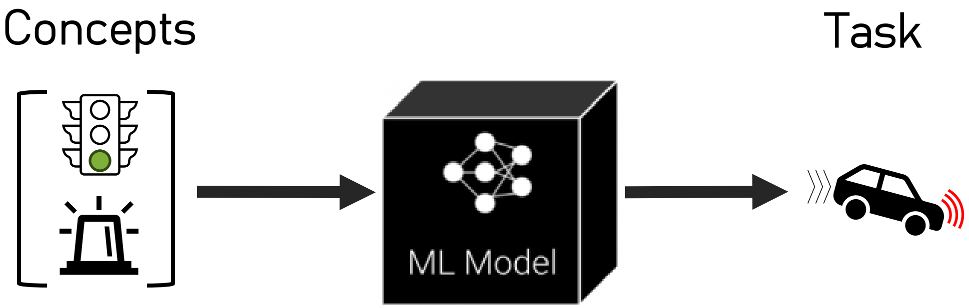
Can we do better?

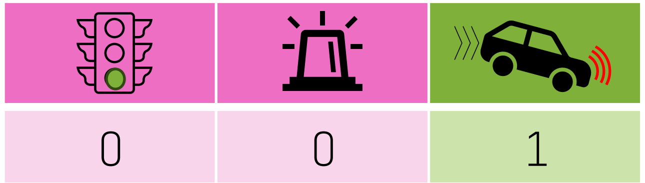
Back to square 1!

FROM INTERVENTIONS TO LOGIC REASONING

Let's say we have a nice set of concepts, what should we use as classification head?

… what about an opaque DNN?



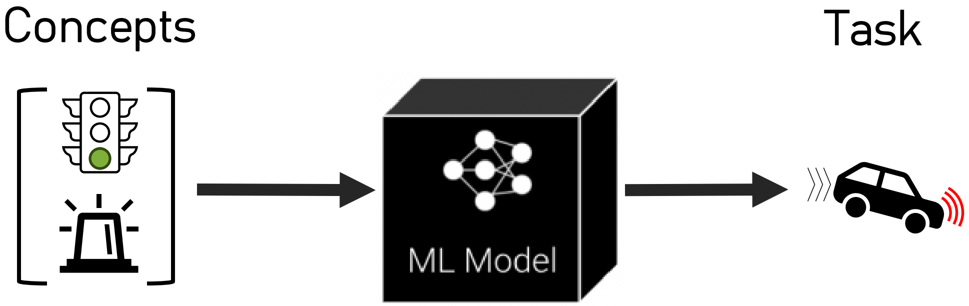


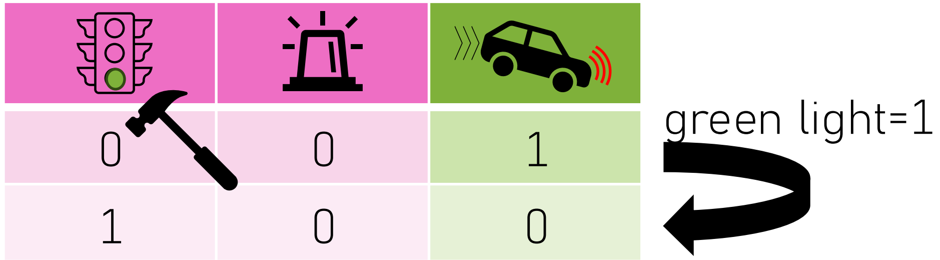
0	0	1

FROM INTERVENTIONS TO LOGIC REASONING

Let's say we have a nice set of concepts, what should we use as classification head?

… what about an opaque DNN?



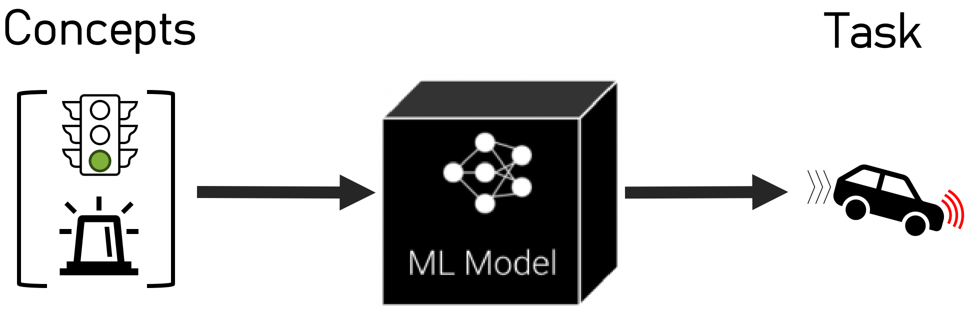


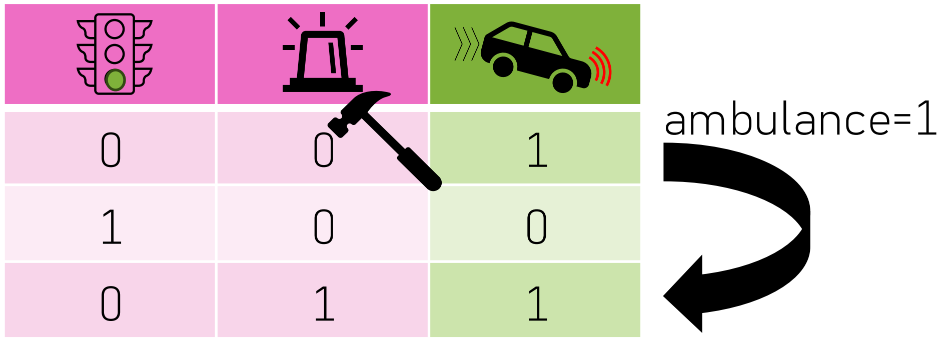
0	0	1
1	0	0

FROM INTERVENTIONS TO LOGIC REASONING

Let's say we have a nice set of concepts, what should we use as classification head?

… what about an opaque DNN?



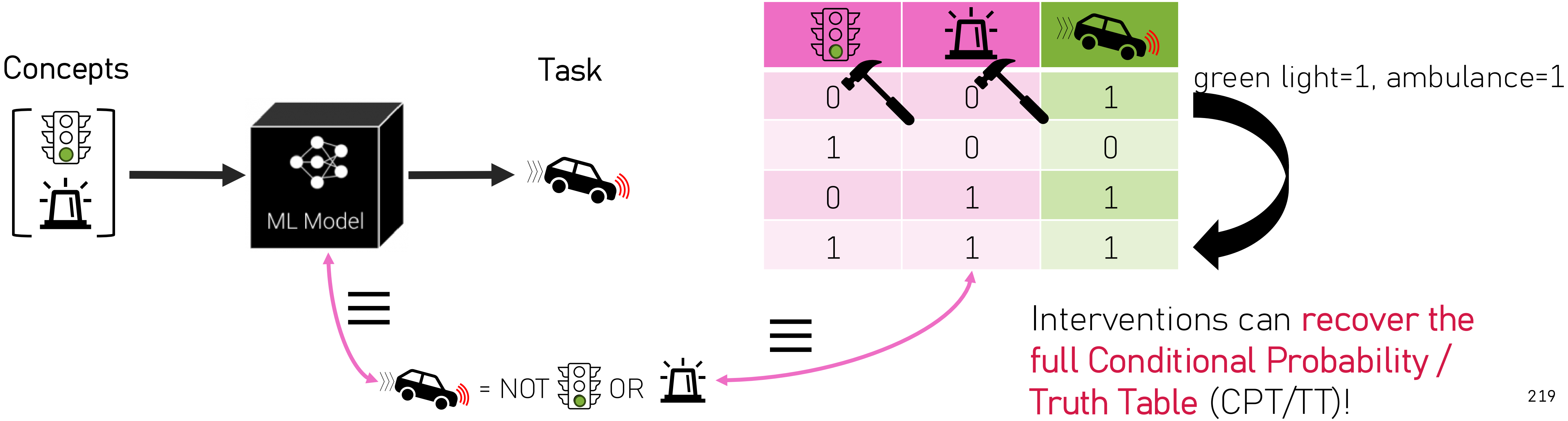


🚦	🚨	🚗
0	0	1
1	0	0
0	1	1

FROM INTERVENTIONS TO LOGIC REASONING

Let's say we have a nice set of concepts, what should we use as **classification head**?

… what about an **opaque DNN**?



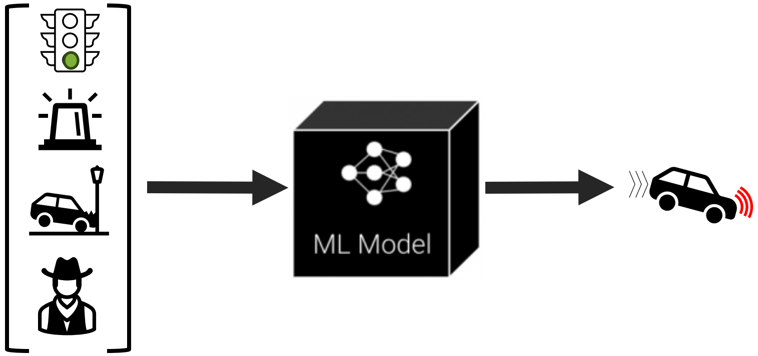
🚦	🚨	🚗
0	0	1
1	0	0
0	1	1
1	1	1

Interventions can **recover the full Conditional Probability / Truth Table** (CPT/TT)!

LOGIC-EXPLAINED NETWORKS (LENS)

Limitation Being Addressed

#interventions required to extract full CPT/TT is exponential in #concepts!



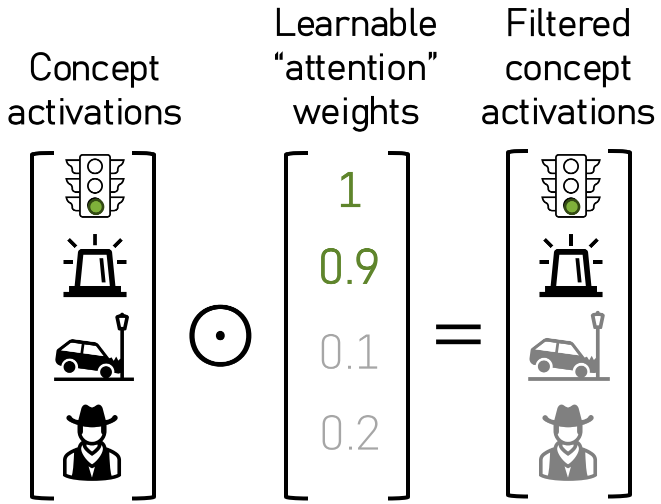
Can we extract a CPT/TT more efficiently?

[1] Barbiero, et al. "Entropy-based logic explanations of neural networks." *AAAI (2022).*

LOGIC-EXPLAINED NETWORKS (LENS)

Proposed Solution

Step 1: Filter concept activations using learnable attention weights α



Concept activations

Learnable “attention” weights

Filtered concept activations

$\odot$

1

0.9

0.1

0.2

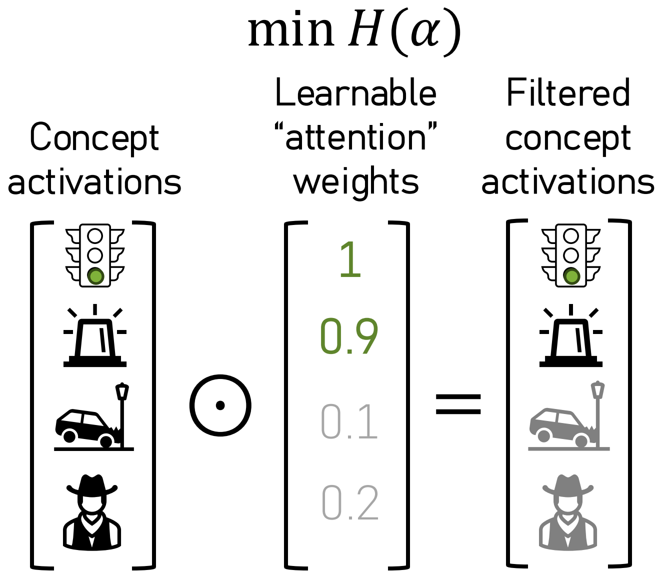
$=$

[1] Barbiero, et al. "Entropy-based logic explanations of neural networks." *AAAI* (2022).

LOGIC-EXPLAINED NETWORKS (LENS)

Proposed Solution

Step 2: Minimize the entropy of the attention weights α. Why? Concept set should be small!



$\min H(\alpha)$

Concept activations

Learnable “attention” weights

Filtered concept activations

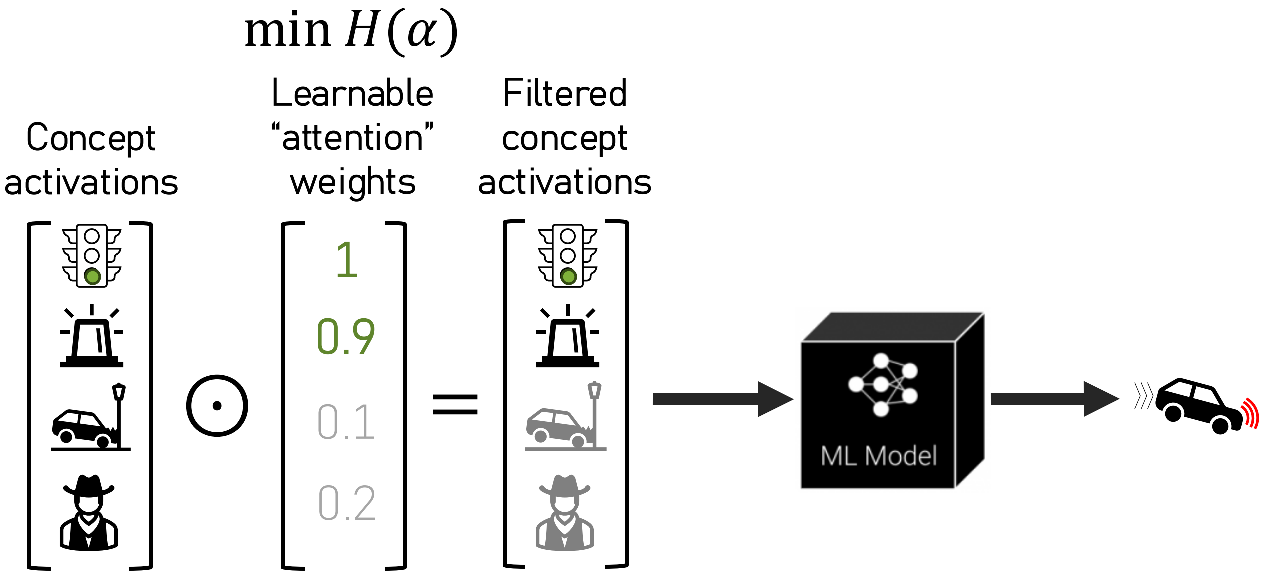
$$\odot \begin{bmatrix} 1 \\ 0.9 \\ 0.1 \\ 0.2 \end{bmatrix} =$$

[1] Barbiero, et al. "Entropy-based logic explanations of neural networks." *AAAI* (2022).

LOGIC-EXPLAINED NETWORKS (LENS)

Proposed Solution

Step 3: Solve task with the selected concepts. Why? Concept set should be relevant!



$$\min H(\alpha)$$

Concept activations

Learnable “attention” weights

Filtered concept activations

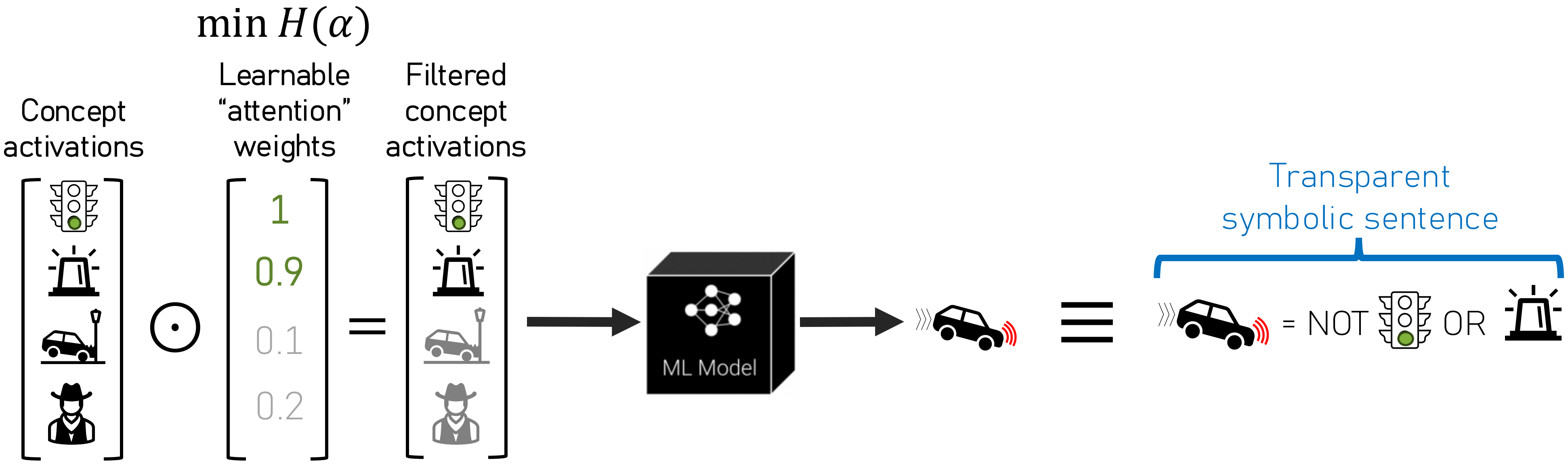
$$\odot \begin{bmatrix} 1 \\ 0.9 \\ 0.1 \\ 0.2 \end{bmatrix} =$$

[1] Barbiero, et al. "Entropy-based logic explanations of neural networks." *AAAI (2022).*

LOGIC-EXPLAINED NETWORKS (LENS)

Proposed Solution

Step 4: Derive explanation in DNF from the (empirical) truth table



$$\min H(\alpha)$$

Concept activations

Learnable “attention” weights

Filtered concept activations

1

0.9

0.1

0.2

ML Model

Transparent symbolic sentence

= NOT OR

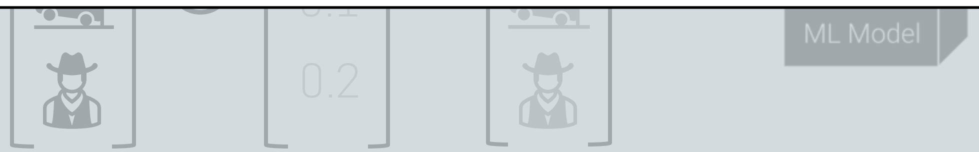
[1] Barbiero, et al. "Entropy-based logic explanations of neural networks." *AAAI* (2022).

LOGIC-EXPLAINED NETWORKS (LENS)

Proposed Solution

Step 4: Derive **explanation in DNF** from the (empirical) truth table

What if we know the logic program,
but we don't have concept supervisions?

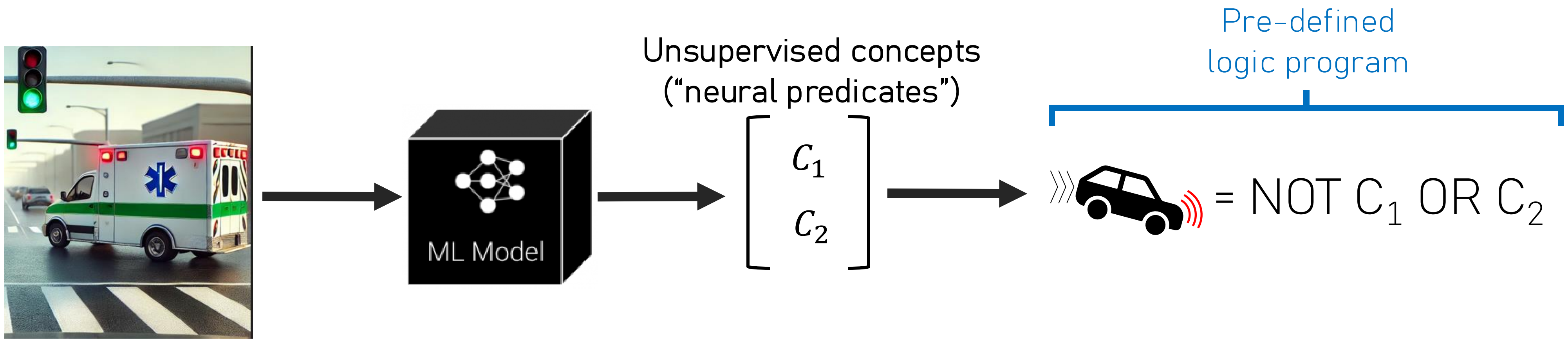


[1] Barbiero, Pietro, et al. "Entropy-based logic explanations of neural networks." *AAAI Conference on Artificial Intelligence*. PMLR, 2022.

NEURAL PROBABILISTIC LOGIC PROGRAMMING

Proposed Solution

Replace task predictor with a pre-defined logic program!



Unsupervised concepts
("neural predicates")

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

Pre-defined
logic program

= NOT C_1 OR C_2

[1] Manhaeve et al. "Deepproblog: Neural probabilistic logic programming." *NeurIPS* 2018.

NEURAL PROBABILISTIC LOGIC PROGRAMMING

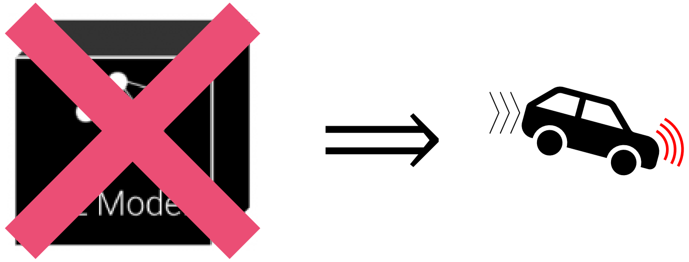
Proposed Solution

Replace task predictor with a pre-defined logic program!

Are symbolic classification heads sufficient for a model to be interpretable?

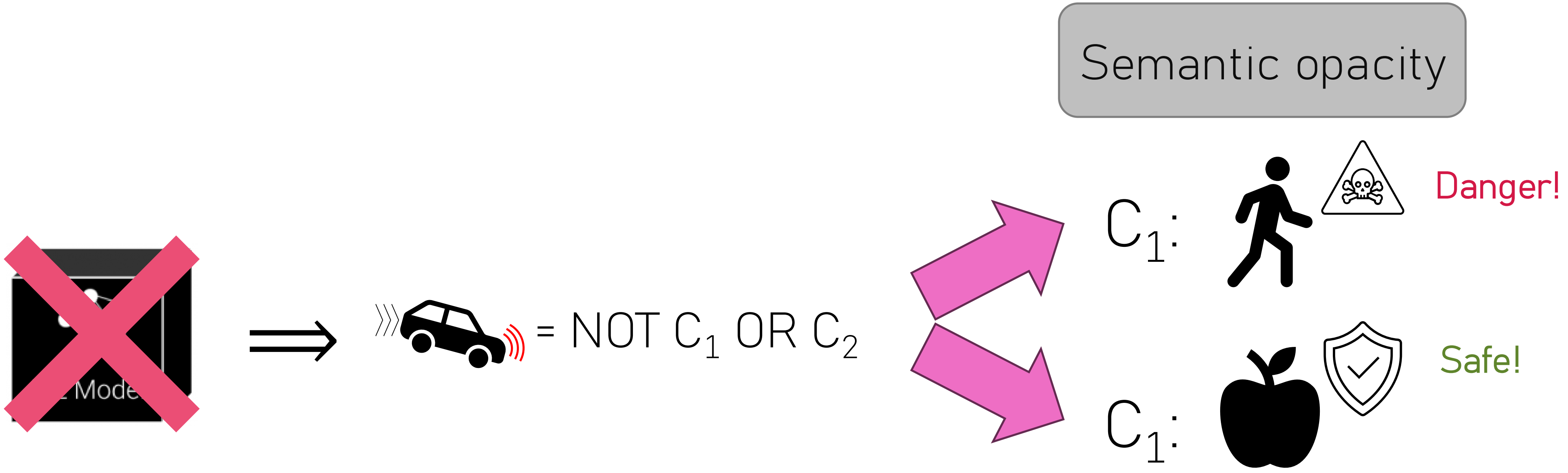
[1] Manhaeve et al. "Deepproblog: Neural probabilistic logic programming." *NeurIPS* 2018.

SEMANTIC & FUNCTIONAL OPACITY



$= \text{NOT } C_1 \text{ OR } C_2$

SEMANTIC & FUNCTIONAL OPACITY



Semantic opacity

$\Rightarrow$ = NOT C_1 OR C_2

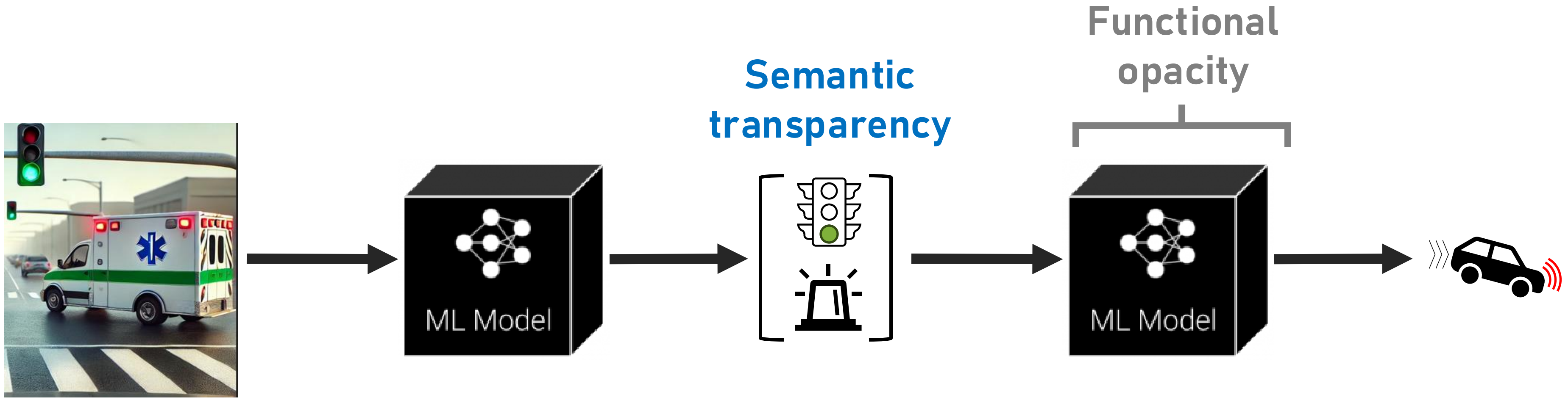
C_1: Danger!

C_1: Safe!

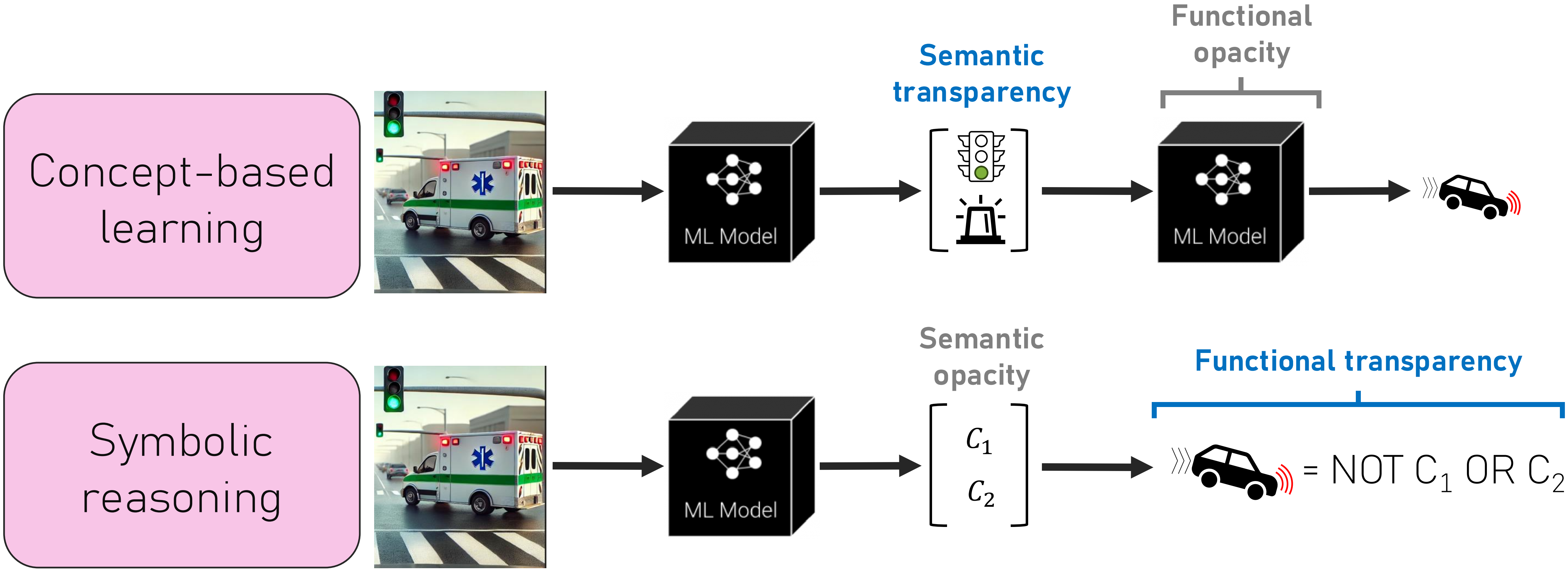
A symbolic classification head alone **does not guarantee semantic transparency**
(… as well as Logistic Regression, Additive Models, Decision Trees, etc…)!

SEMANTIC & FUNCTIONAL OPACITY

Concept-based learning



SEMANTIC & FUNCTIONAL OPACITY



Concept-based learning

ML Model

Semantic transparency

ML Model

Functional opacity

Symbolic reasoning

ML Model

Semantic opacity

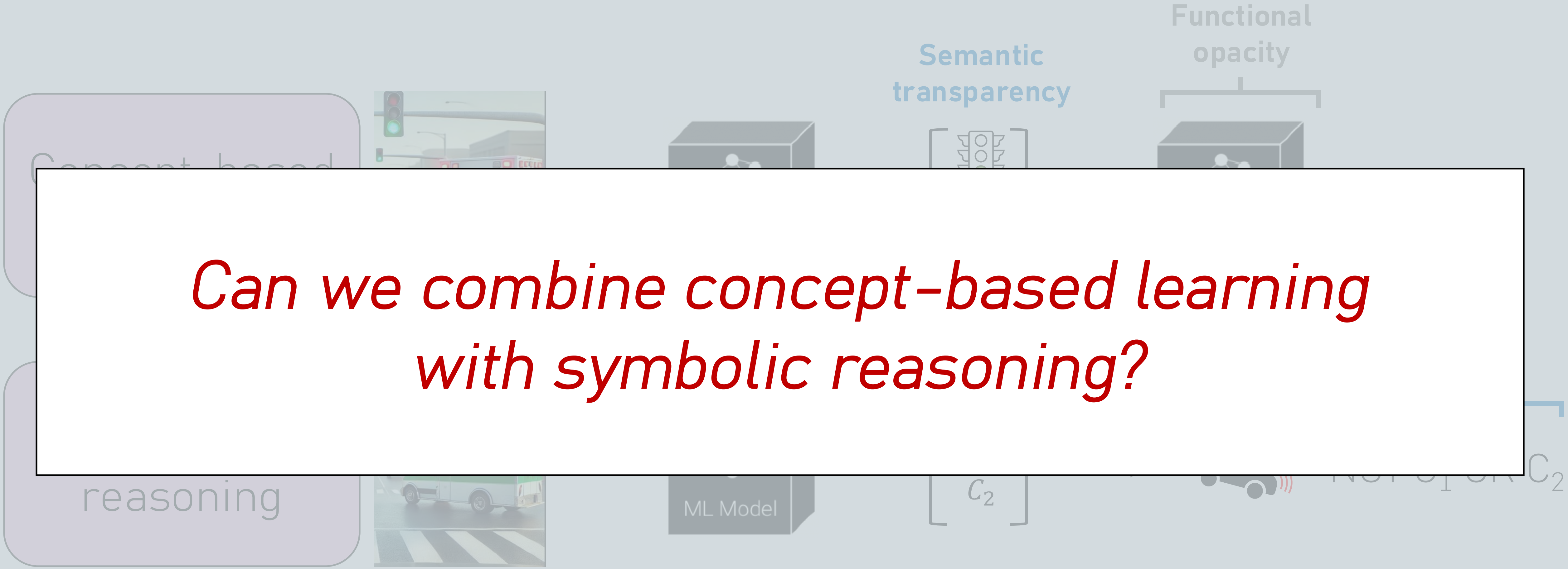
C_1

C_2

Functional transparency

$= \text{NOT } C_1 \text{ OR } C_2$

SEMANTIC & FUNCTIONAL OPACITY

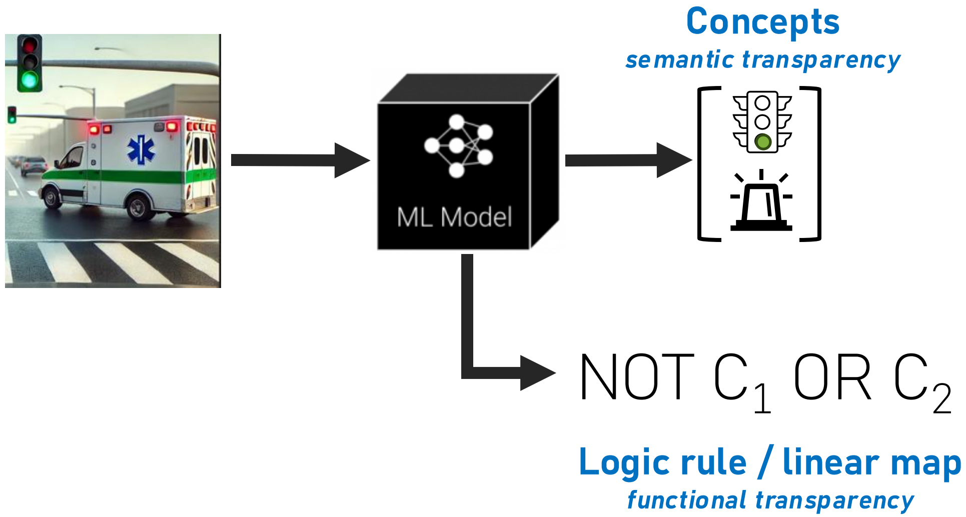


Can we combine concept-based learning with symbolic reasoning?

NEURAL-SYMBOLIC CONCEPT REASONING

Proposed Solution

Step 1: DNN generates both concept activations & rule parameters *(neural generation)*



Concepts

semantic transparency

NOT C_1 OR C_2

Logic rule / linear map

functional transparency

[1] Barbiero et al. "Interpretable neural-symbolic concept reasoning." International Conference on Machine Learning. PMLR, 2023.
[2] Debot et al. "Interpretable concept-based memory reasoning." NeurIPS 2024.

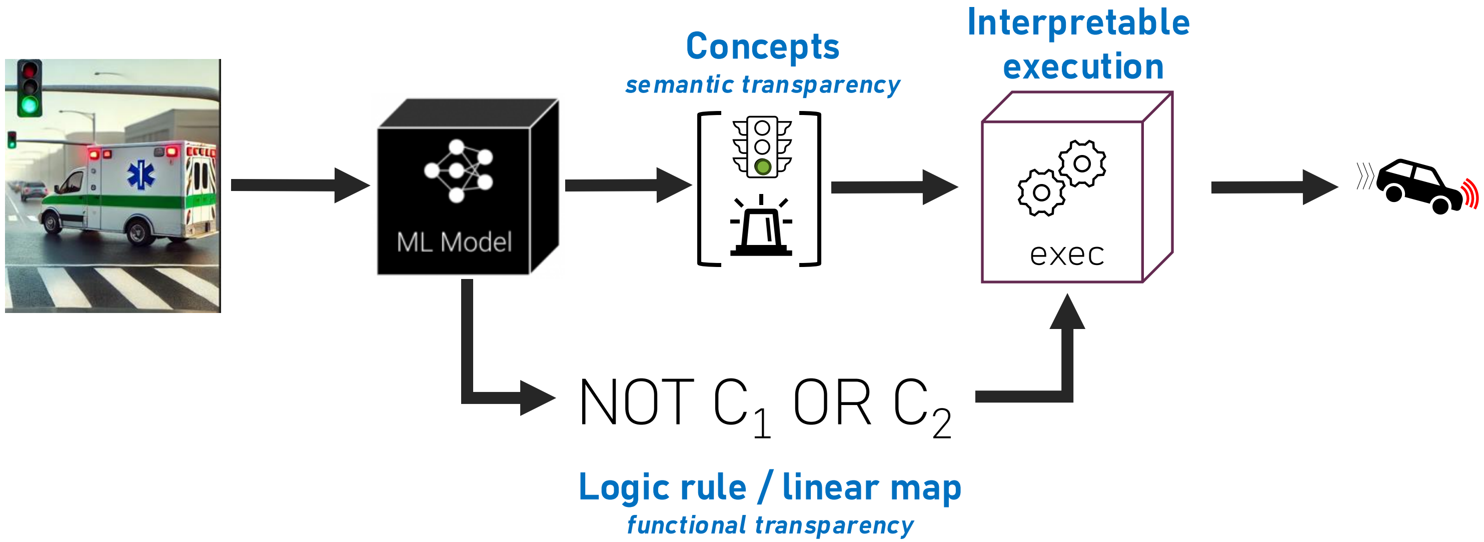
ICML23

NeurIPS24

NEURAL-SYMBOLIC CONCEPT REASONING

Proposed Solution

Step 1: DNN generates both concept activations & rule parameters *(neural generation)*
Step 2: Symbolic engine executes the rule using concept activations *(interpretable execution)*



Concepts
semantic transparency

Interpretable execution

NOT C_1 OR C_2

Logic rule / linear map
functional transparency

ICML23

NeurIPS24

[1] Barbiero et al. "Interpretable neural-symbolic concept reasoning." International Conference on Machine Learning. PMLR, 2023.
[2] Debot et al. "Interpretable concept-based memory reasoning." NeurIPS 2024.

CONCEPT-BASED MEMORY REASONING

Proposed Solution

Step 1: DNN predicts concept activations

$p(c \mid x)$

ML Model

Concepts

semantic transparency

= 0.2

= 0.9

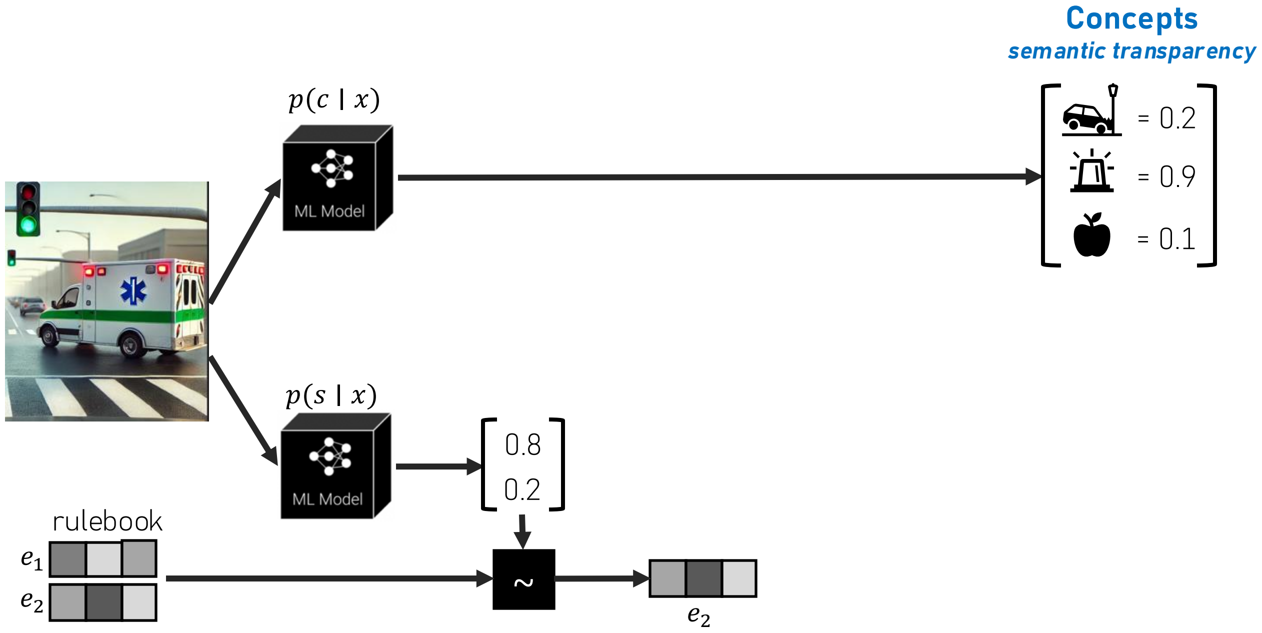
= 0.1

[1] Debot et al. "Interpretable concept-based memory reasoning." NeurIPS 2024.

CONCEPT-BASED MEMORY REASONING

Proposed Solution

Step 2: DNN predicts embedding to be selected from the latent rulebook

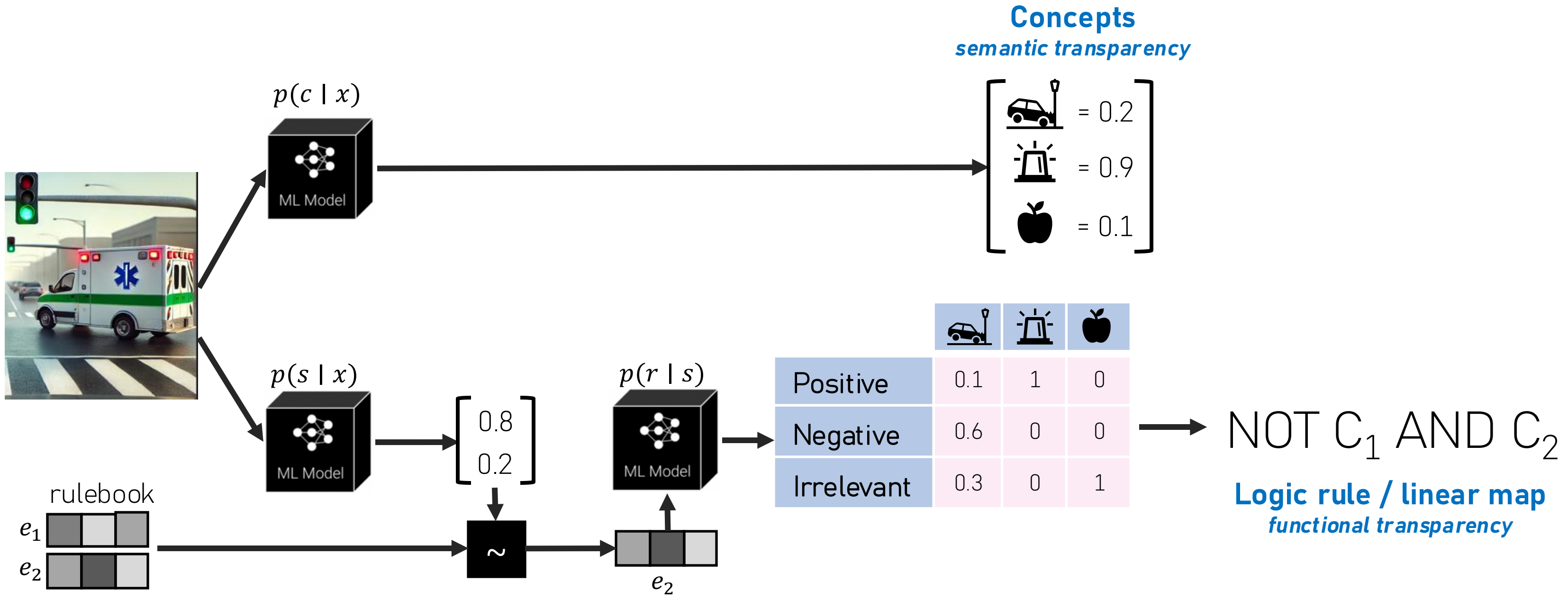


[1] Debot et al. "Interpretable concept-based memory reasoning." NeurIPS 2024.

CONCEPT-BASED MEMORY REASONING

Proposed Solution

Step 3: DNN decodes selected embedding into 3 states: positive, negative, irrelevant



	🚗	🚨	🍎
Positive	0.1	1	0
Negative	0.6	0	0
Irrelevant	0.3	0	1

NOT C_1 AND C_2

Logic rule / linear map

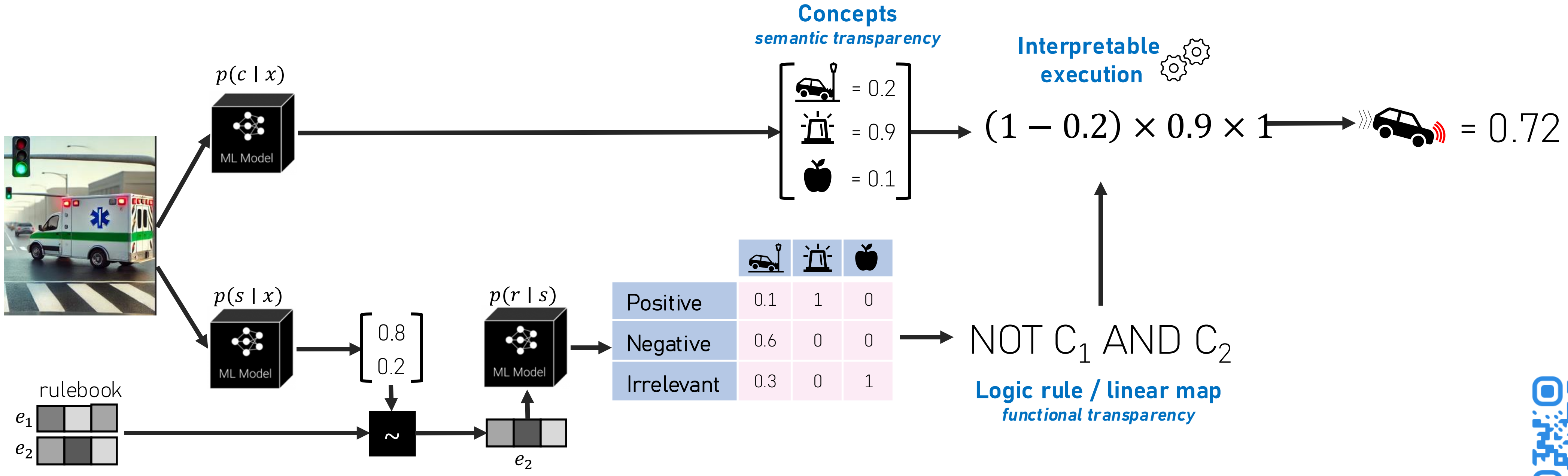
functional transparency

[1] Debot et al. "Interpretable concept-based memory reasoning." NeurIPS 2024.

CONCEPT-BASED MEMORY REASONING

Proposed Solution

Step 4: Execute the rule combining concept states and activations to predict the output label



	🚗	🚨	🍎
Positive	0.1	1	0
Negative	0.6	0	0
Irrelevant	0.3	0	1

[1] Debot, David, et al. "Interpretable concept-based memory reasoning." NeurIPS 2024.

CONCEPT-BASED MEMORY REASONING

CMR has 3 key features:

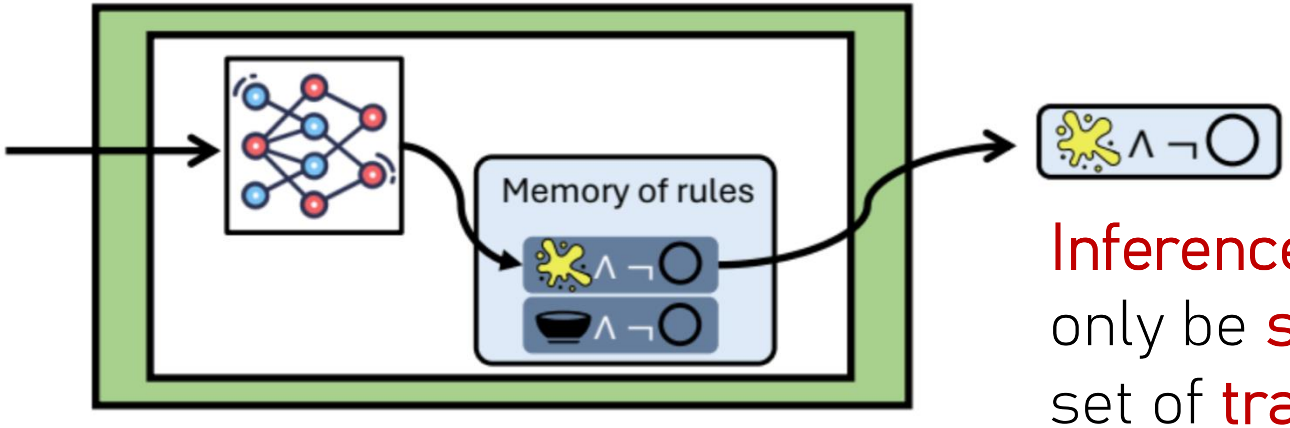
- Universal approximator akin to opaque DNNs (Theorem 4.1)

[1] Debot et al. "Interpretable concept-based memory reasoning." NeurIPS 2024.

CONCEPT-BASED MEMORY REASONING

CMR has 3 key features:

- Universal approximator akin to opaque DNNs (Theorem 4.1)
- Provides both local and global interpretability by design



Inference mechanisms can only be selected from a finite set of transparent rules!

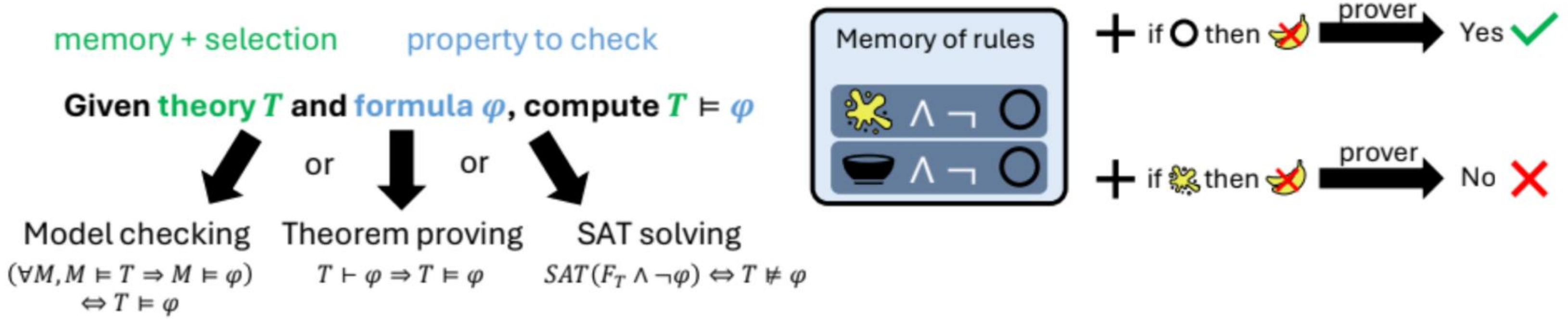
[1] Debot et al. "Interpretable concept-based memory reasoning." NeurIPS 2024.

CONCEPT-BASED MEMORY REASONING

CMR has 3 key features:

- Universal approximator akin to opaque DNNs (Theorem 4.1)
- Provides both local and global interpretability by design
- The concept memory allows formal verification of properties

"Does a property hold no matter which rule is selected?"



memory + selection property to check

Given theory T and formula φ, compute $T \vDash \varphi$

or or

Model checking $(\forall M, M \vDash T \Rightarrow M \vDash \varphi) \Leftrightarrow T \vDash \varphi$

Theorem proving $T \vdash \varphi \Rightarrow T \vDash \varphi$

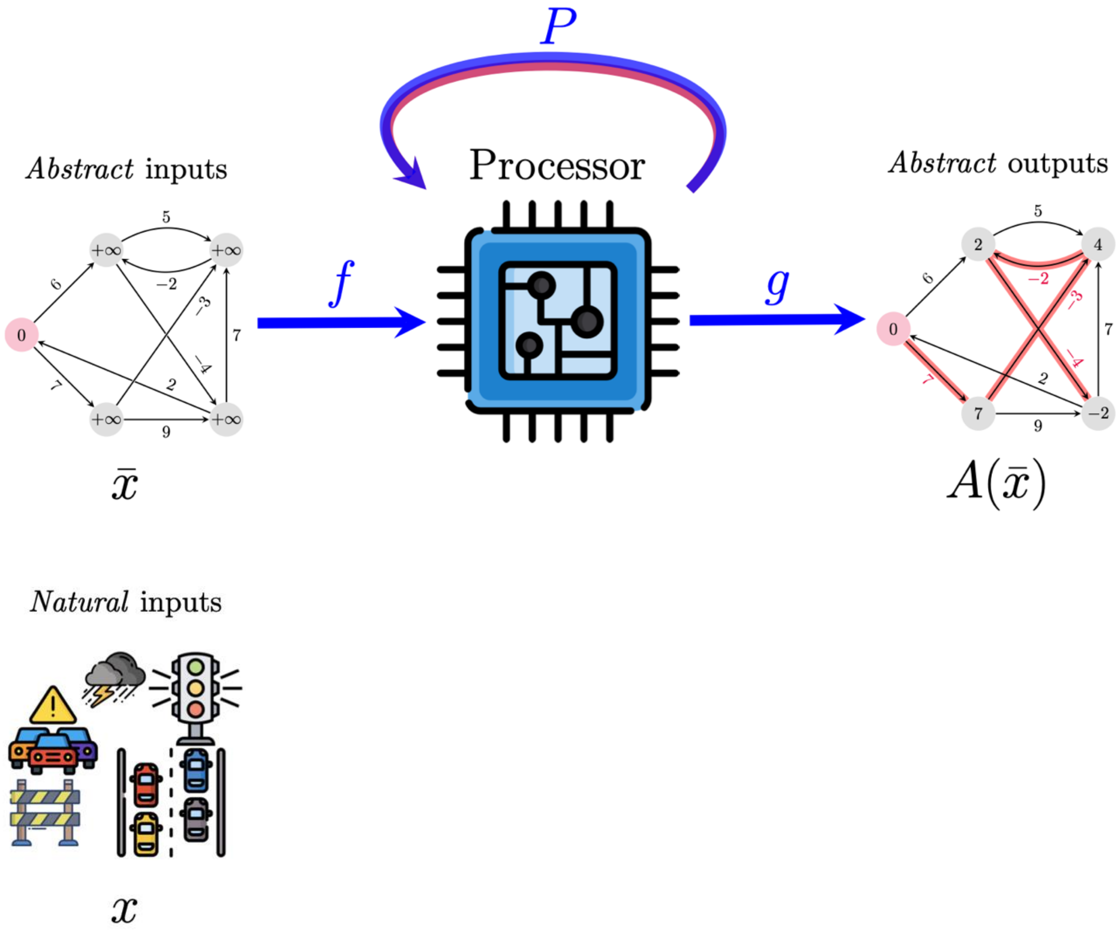
SAT solving $SAT(F_T \wedge \neg\varphi) \Leftrightarrow T \nvDash \varphi$

[1] Debot et al. "Interpretable concept-based memory reasoning." NeurIPS 2024.

ARE WE JUST TALKING HOT AIR?

NEURAL ALGORITHMIC REASONING

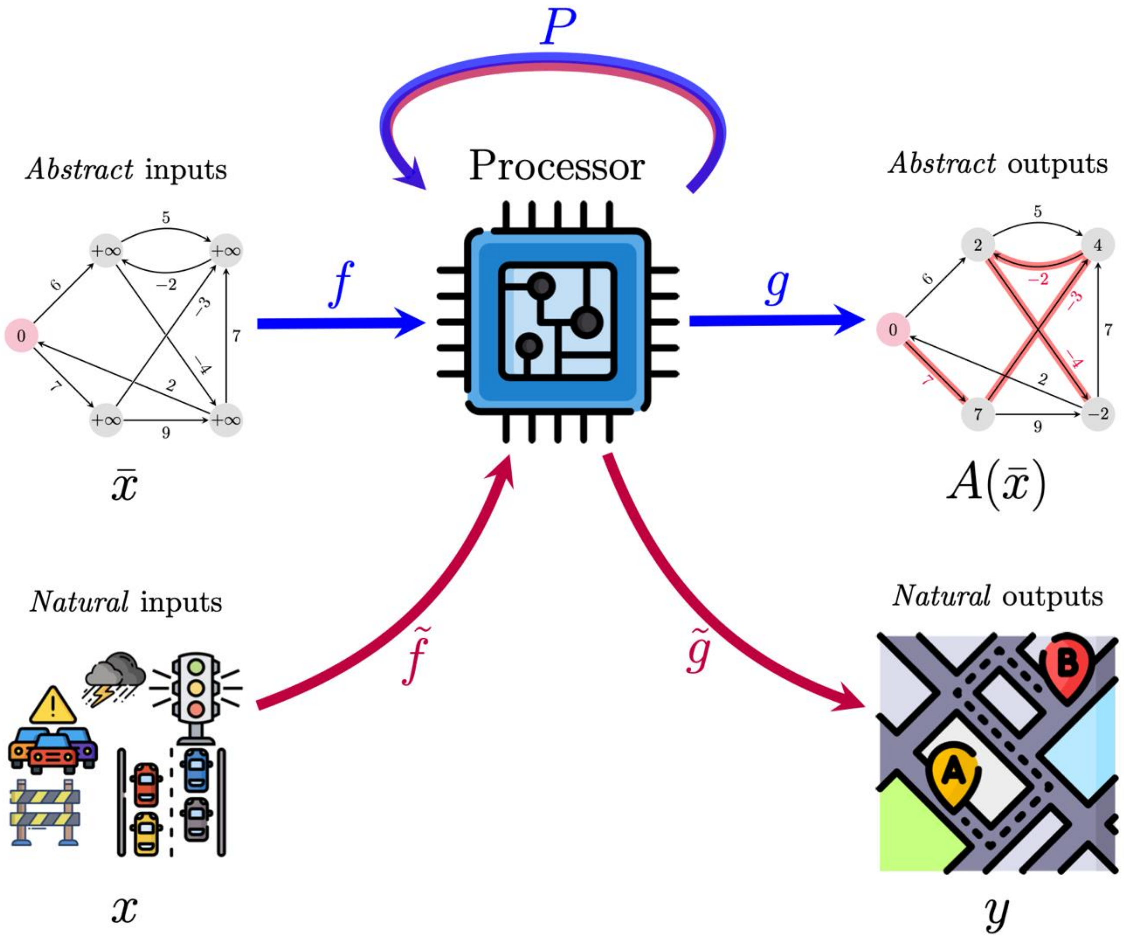
- Algorithmic reasoning
  - + OOD generalization
  - – discrete representations
- Neural nets
  - – OOD generalization
  - + continuous representations



[1] Veličković et al. "Neural algorithmic reasoning." Patterns (2021).

NEURAL ALGORITHMIC REASONING

- Execute algorithms with DNNs
  - + OOD generalization (from algorithm exec)
  - + adapt to real-world inputs (e.g., images)

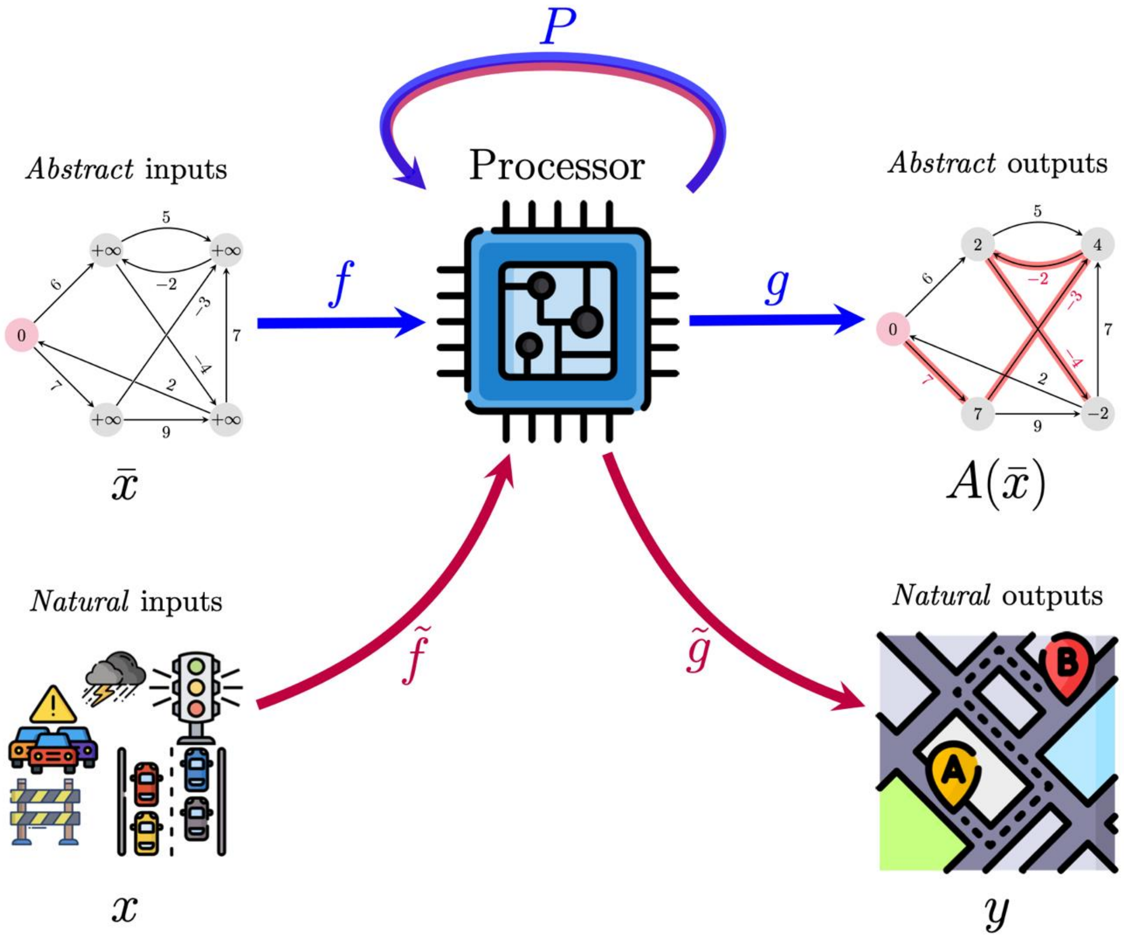


[1] Veličković et al. "Neural algorithmic reasoning." Patterns (2021).

NEURAL ALGORITHMIC REASONING

- Execute algorithms with DNNs
  - + OOD generalization (from algorithm exec)
  - + adapt to real-world inputs (e.g., images)
  - + (potentially) find new heuristics!

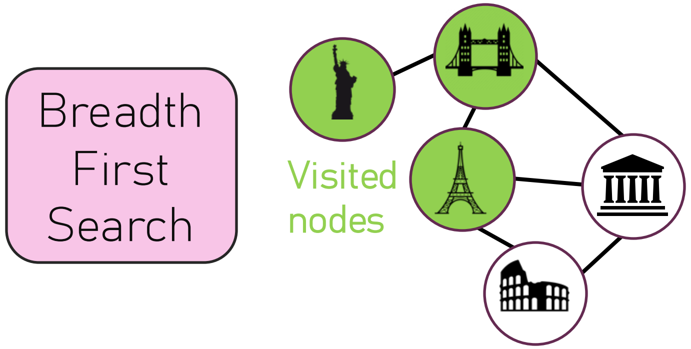
How?



[1] Veličković et al. "Neural algorithmic reasoning." Patterns (2021).

NEURAL ALGORITHMIC REASONING

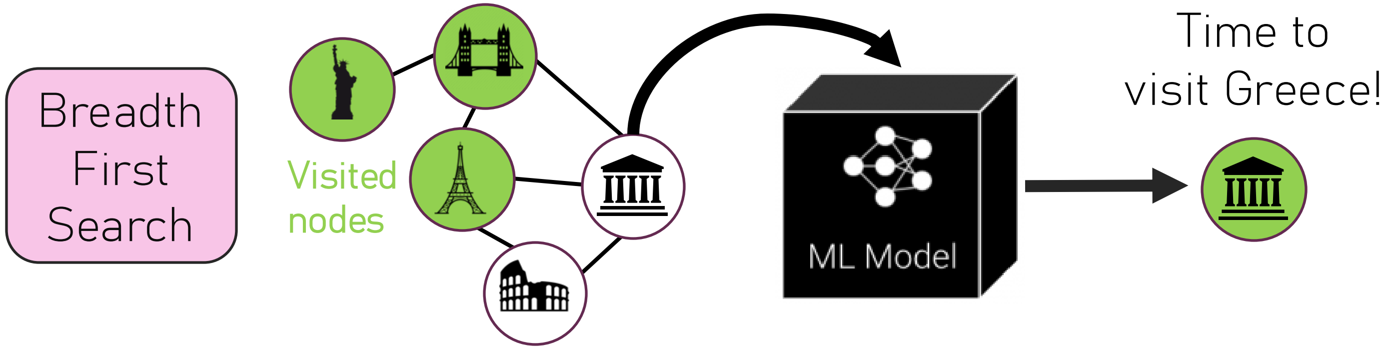
- Execute algorithms with DNNs
  - + OOD generalization (from algorithm exec)
  - + adapt to real-world inputs (e.g., images)
  - + (potentially) find new heuristics!



[1] Veličković et al. "Neural algorithmic reasoning." Patterns (2021).

NEURAL ALGORITHMIC REASONING

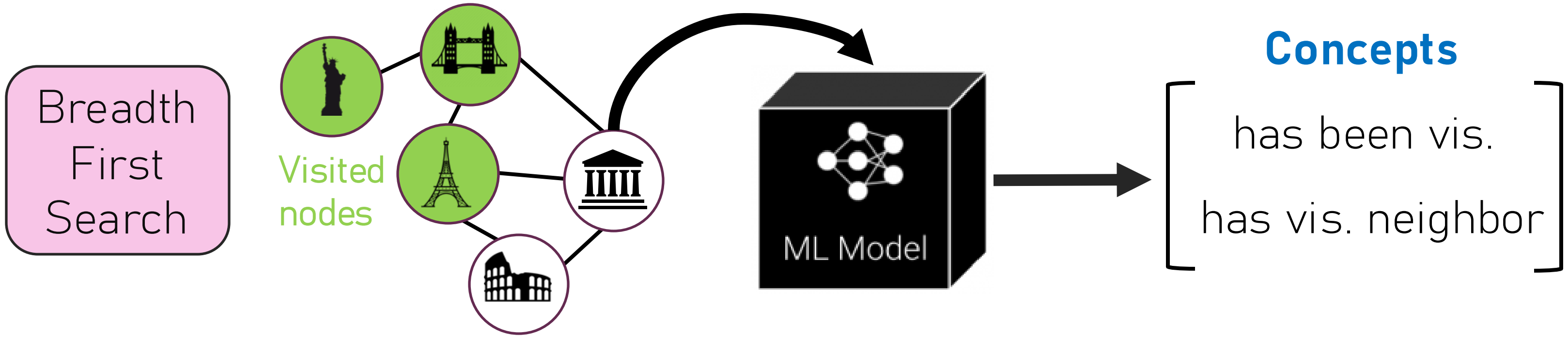
- Execute algorithms with DNNs
  - + OOD generalization (from algorithm exec)
  - + adapt to real-world inputs (e.g., images)
  - + (potentially) find new heuristics!



[1] Veličković et al. "Neural algorithmic reasoning." Patterns (2021).

CONCEPT-BASED NEURAL ALGORITHMIC REASONING

- Execute algorithms with DNNs
  - + OOD generalization (from algorithm exec)
  - + adapt to real-world inputs (e.g., images)
  - + (potentially) find new heuristics!



Breadth First Search

Visited nodes

ML Model

Concepts

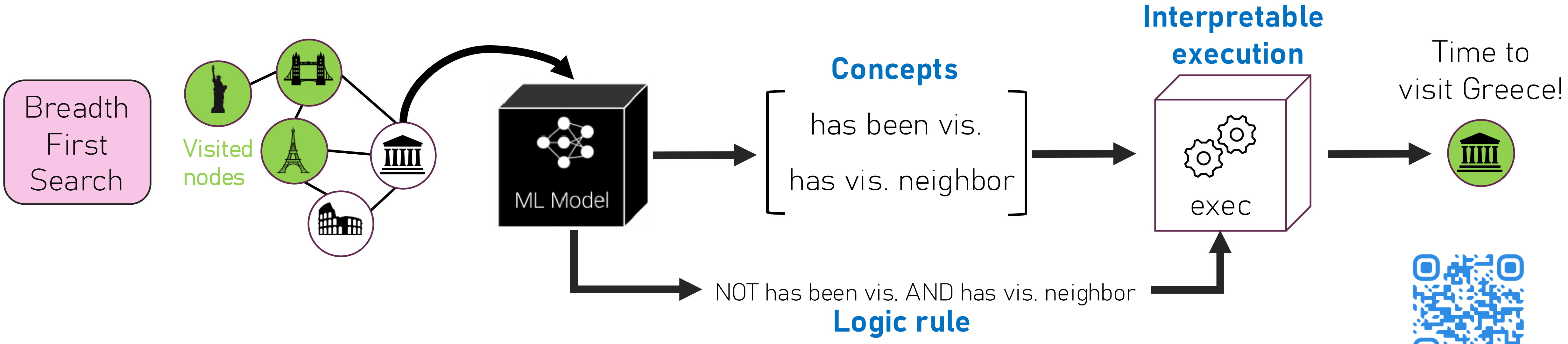
has been vis.

has vis. neighbor

[1] Georgiev et al. "Algorithmic concept-based explainable reasoning." AAAI (2022).

CONCEPT-BASED NEURAL ALGORITHMIC REASONING

- Execute algorithms with DNNs
    - + OOD generalization (from algorithm exec)
    - + adapt to real-world inputs (e.g., images)
    - + (potentially) find new heuristics!



[1] Georgiev et al. "Algorithmic concept-based explainable reasoning." AAAI (2022).

SHOULD INTERPRETABILITY BOTHER ABOUT CAUSALITY?

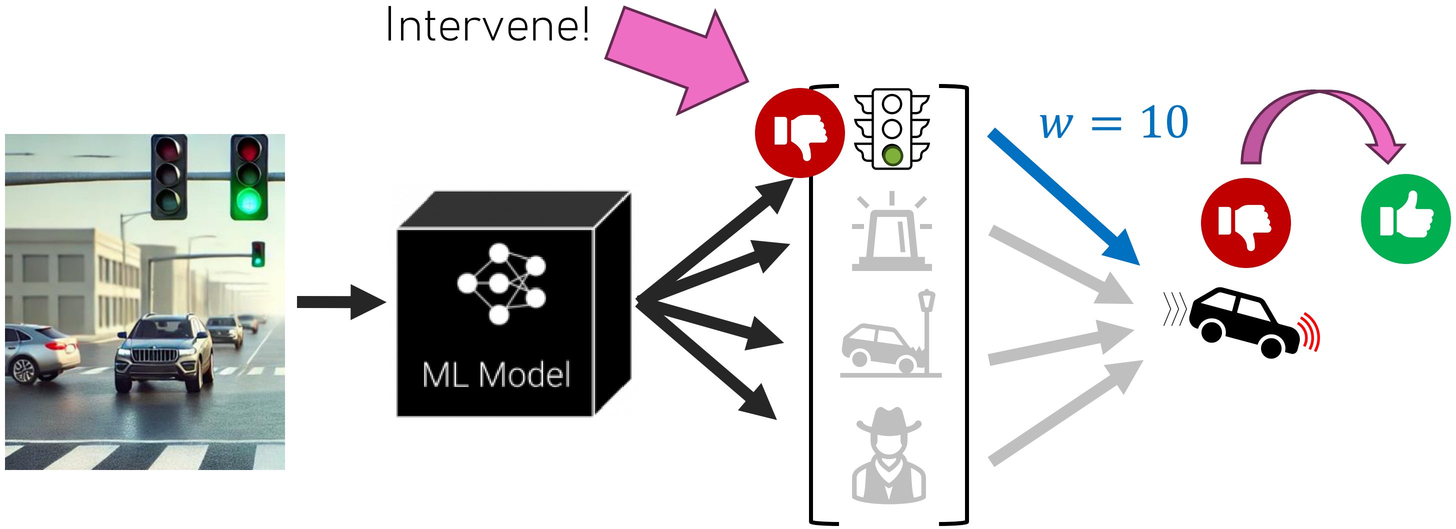
We'll focus on two main branches of concept-based reasoning:

Neural symbolic concept reasoning

Causal concept reasoning

SHOULD INTERPRETABILITY BOTHER ABOUT CAUSALITY?

Sometimes intervening on wrongly predicted concepts helps…

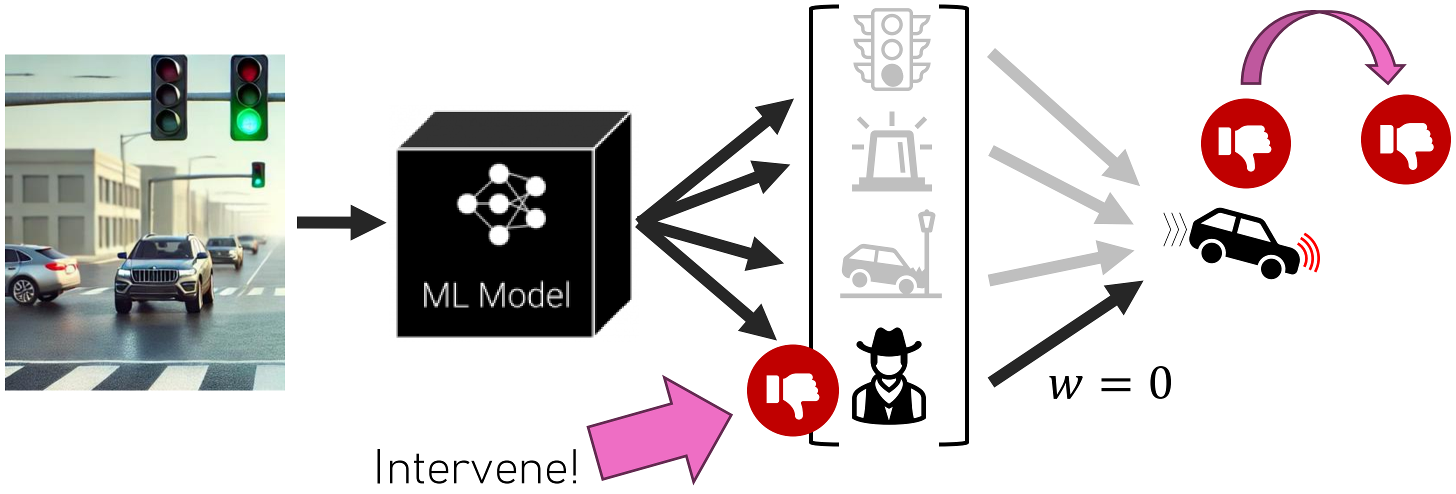


SHOULD INTERPRETABILITY BOTHER ABOUT CAUSALITY?

Sometimes intervening on wrongly predicted concepts helps…

and sometimes it doesn't! 😢

Causal analysis can provide us with insights!



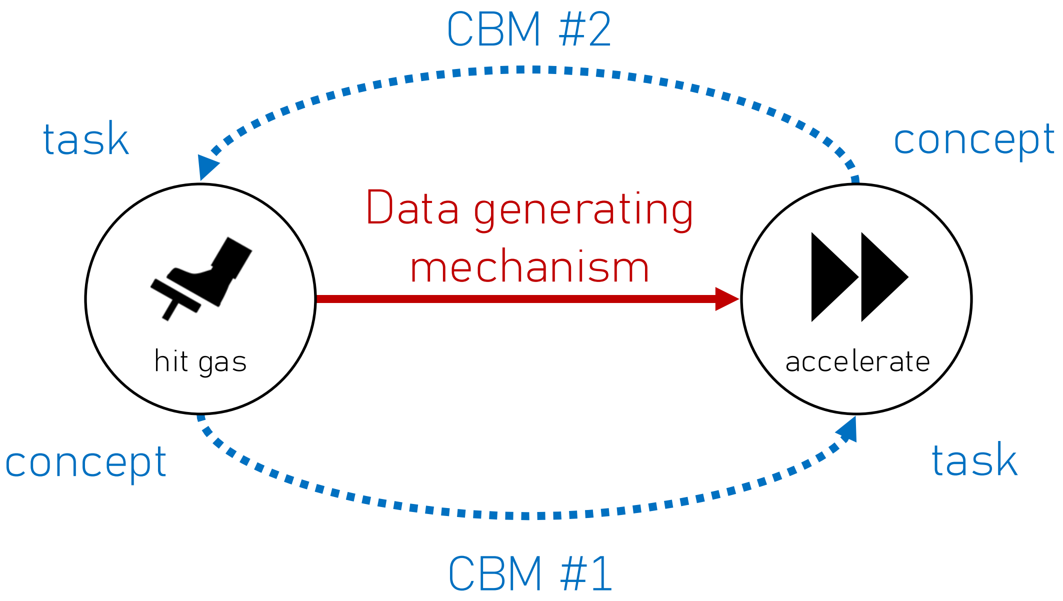
Yes, but…

CAUSAL OPACITY

- **Causal reliability**: discover causal mechanisms of the **data generating process**

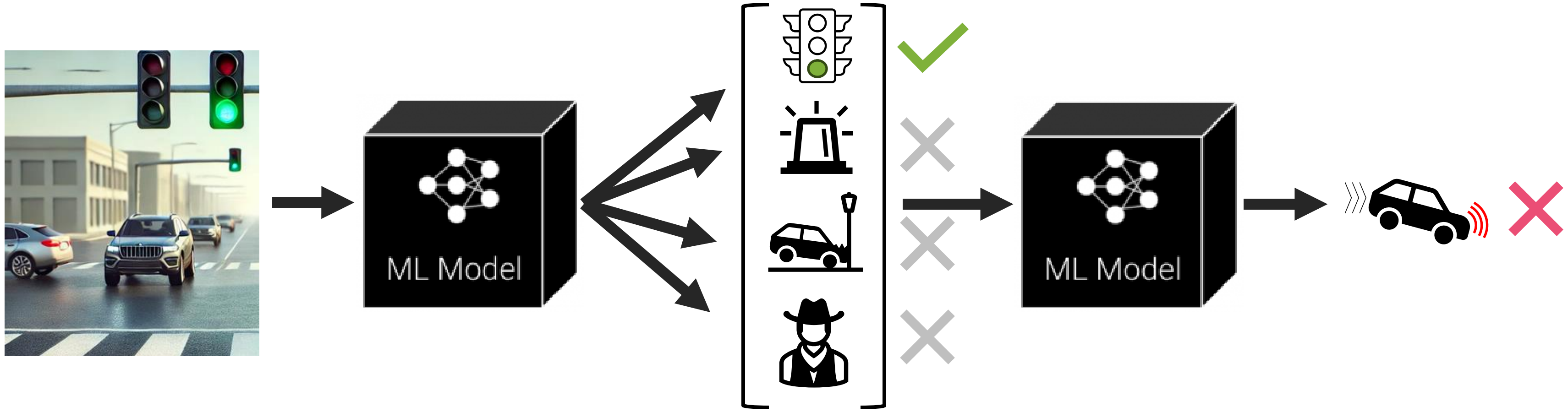
CAUSAL OPACITY

- Causal reliability: discover causal mechanisms of the data generating process
- Causal opacity: discover causal mechanism of a model's inference process



CONCEPT-BASED CAUSAL REASONING

CBMs can answer association queries (duh…)



Association

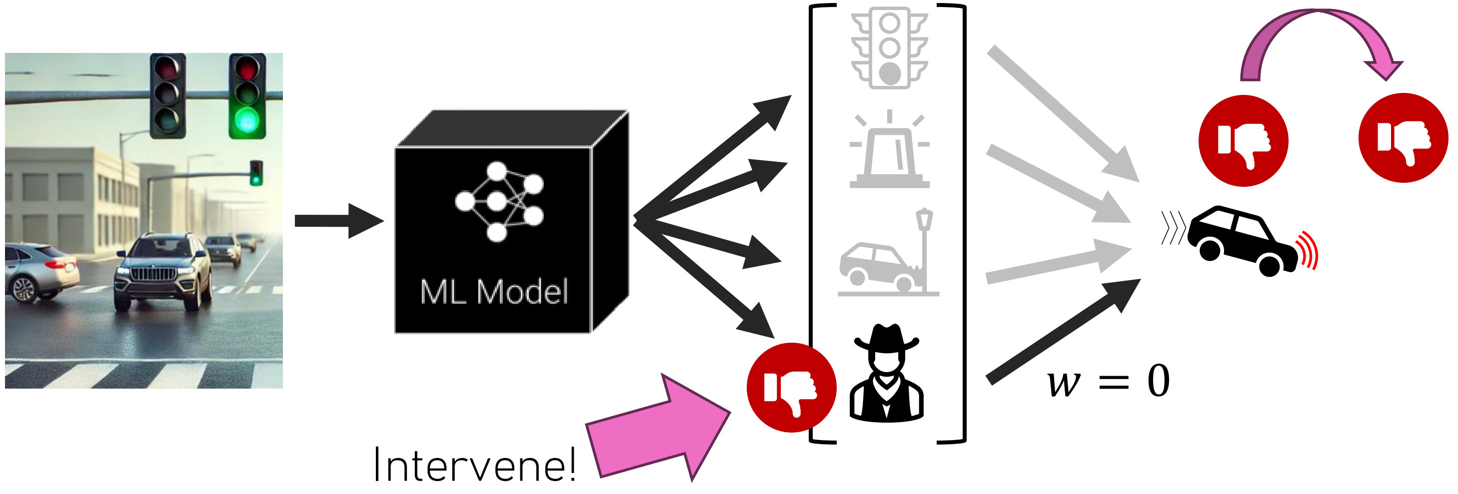
What if the model sees a green light?

$P(brake \mid light)$

CONCEPT-BASED CAUSAL REASONING

CBMs can answer association queries (duh…)

However, intervening on 🚦 influences the task, while intervening on 🕵 does not!



Intervention

What if I set the light color to red?

$P(brake \mid do(light))$

Association

What if the model sees a green light?

$P(brake \mid light)$

CONCEPT-BASED CAUSAL REASONING

CBMs can answer association queries (duh…)

However, intervening on 🚦 influences the task, while intervening on 🕵 does not!

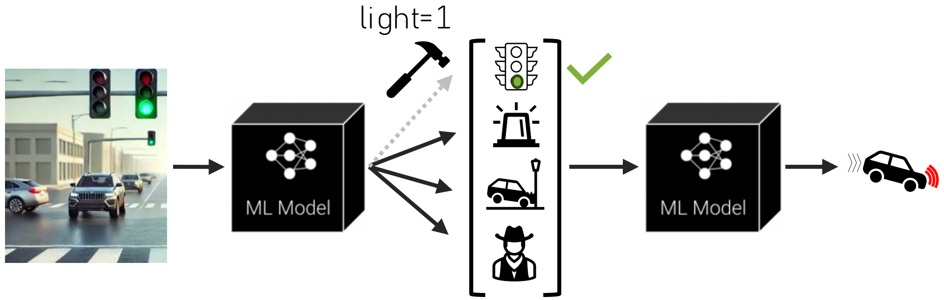
Can we measure the causal influence of a concept on the task?

What if the model sees a green light?
$P(brake \mid light)$

CAUSAL CONCEPT EFFECT

Proposed Solution

Step 1: Compute expected value of the task with $do(c_i = 1)$



$$\mathbb{E}[\,\text{brake} \mid do(\text{light} = 1)\,] = 0.2$$

Intervention

What if I set the light color to red?

$P(brake \mid do(light))$

Association

What if the model sees a green light?

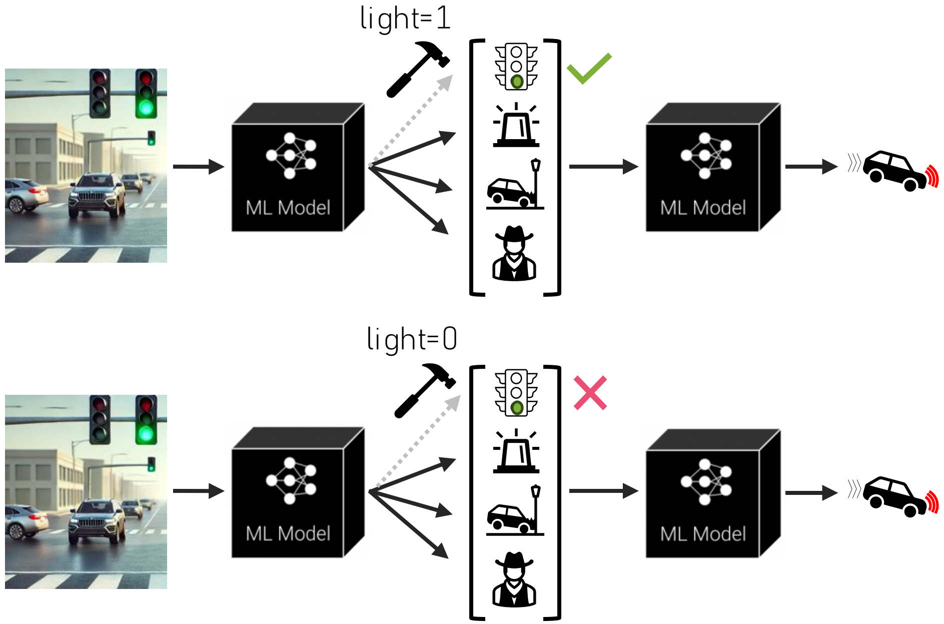
$P(brake \mid light)$

[1] Goyal et al. "Explaining classifiers with causal concept effect (cace)." arXiv (2019).

CAUSAL CONCEPT EFFECT

Proposed Solution

Step 2: Compute expected value of the task with $do(c_i = 0)$



$$\mathbb{E}[\,\text{brake} \mid do(\text{light} = 1)\,] = 0.2$$

$$\mathbb{E}[\,\text{brake} \mid do(\text{light} = 0)\,] = 1$$

Intervention

What if I set the light color to red?

$P(brake \mid do(light))$

Association

What if the model sees a green light?

$P(brake \mid light)$

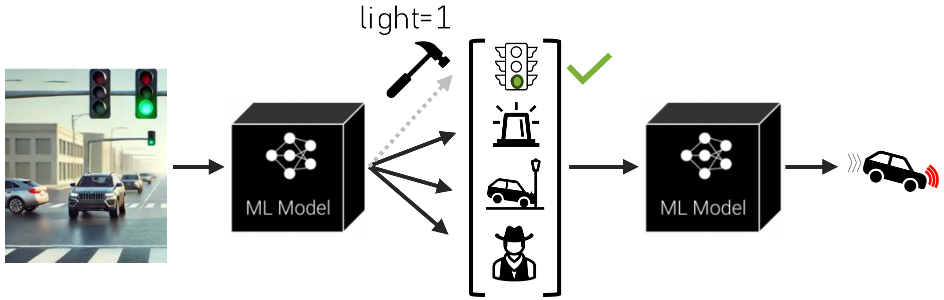
[1] Goyal et al. "Explaining classifiers with causal concept effect (cace)." arXiv (2019).

CAUSAL CONCEPT EFFECT

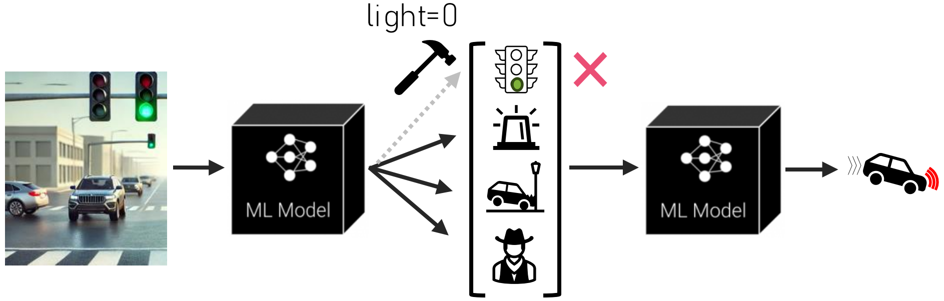
Proposed Solution

Step 3: Compute difference of expected values: absolute value is proportional to causal effect

$$\text{CaCE} = \mathbb{E}[\,\text{brake} \mid do(\text{light} = 1)\,] - \mathbb{E}[\,\text{brake} \mid do(\text{light} = 0)\,] = -0.8$$



$$\mathbb{E}[\,\text{brake} \mid do(\text{light} = 1)\,] = 0.2$$



$$\mathbb{E}[\,\text{brake} \mid do(\text{light} = 0)\,] = 1$$

Intervention

What if I set the light color to red?

$P(brake \mid do(light))$

Association

What if the model sees a green light?

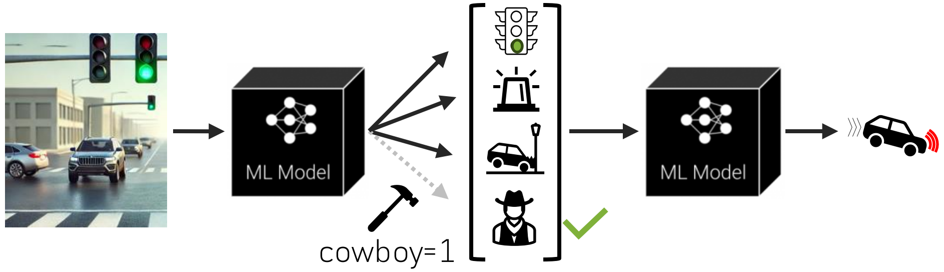
$P(brake \mid light)$

[1] Goyal et al. "Explaining classifiers with causal concept effect (cace)." arXiv (2019).

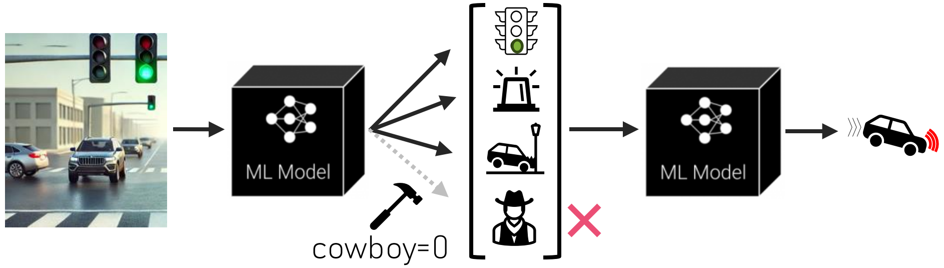
CAUSAL CONCEPT EFFECT

Proposed Solution

Step 3: Compute difference of expected values: absolute value is proportional to causal effect



$$\mathbb{E}[\,\text{brake} \mid do(\text{cowboy} = 1)\,] = 0.5$$



$$\mathbb{E}[\,\text{brake} \mid do(\text{cowboy} = 0)\,] = 0.5$$

Intervention

What if I set the light color to red?

$P(brake \mid do(light))$

Association

What if the model sees a green light?

$P(brake \mid light)$

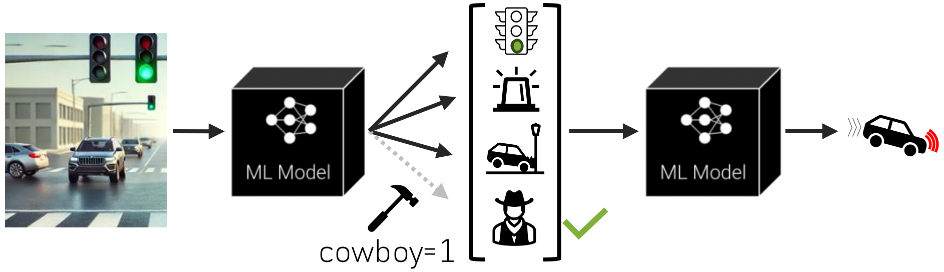
[1] Goyal et al. "Explaining classifiers with causal concept effect (cace)." arXiv (2019).

CAUSAL CONCEPT EFFECT

Proposed Solution

Step 3: Compute difference of expected values: absolute value is proportional to causal effect

$$\text{CaCE} = \mathbb{E}[\,\text{brake} \mid do(\text{cowboy} = 1)\,] - \mathbb{E}[\,\text{brake} \mid do(\text{cowboy} = 0)\,] = 0$$

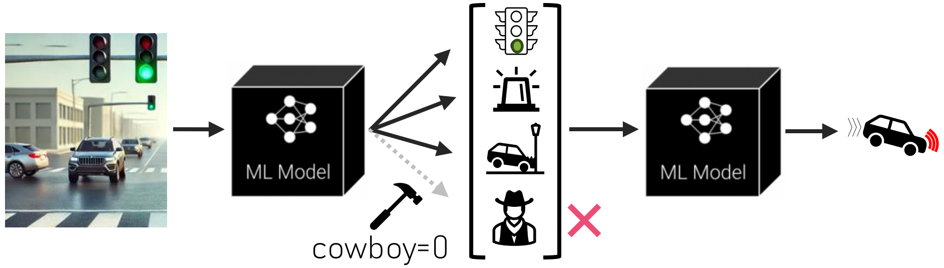


$$\mathbb{E}[\,\text{brake} \mid do(\text{cowboy} = 1)\,] = 0.5$$

Intervention

What if I set the light color to red?

$P(brake \mid do(light))$



$$\mathbb{E}[\,\text{brake} \mid do(\text{cowboy} = 0)\,] = 0.5$$

Association

What if the model sees a green light?

$P(brake \mid light)$

[1] Goyal et al. "Explaining classifiers with causal concept effect (cace)." arXiv (2019).

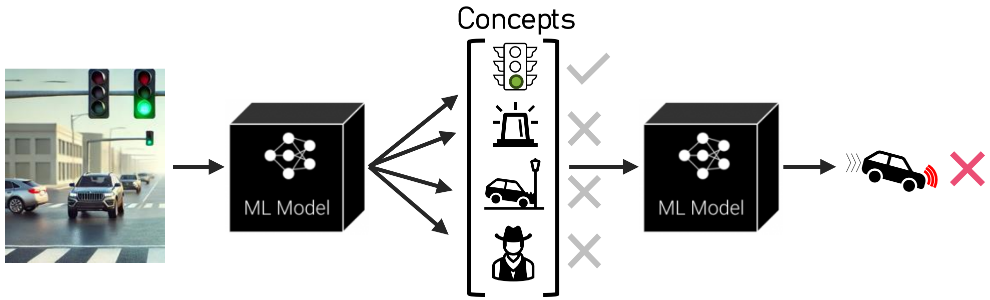
COUNTERFACTUAL CBMS

ICLR25

ICML22

Limitation Being Addressed

CBMs cannot answer counterfactual queries!



Counterfactual

What would have been predicted in the same circumstance had a car crash be seen?

$$P(brake \mid light, crash)$$

Intervention

What if I set the light color to red?

$$P(brake \mid do(light))$$

Association

What if the model sees a green light?

$$P(brake \mid light)$$

[1] Dominici et al. "Counterfactual Concept Bottleneck Models." ICLR (2025).
[2] Abid et al. "Meaningfully debugging model mistakes using conceptual counterfactual explanations." ICML (2022).

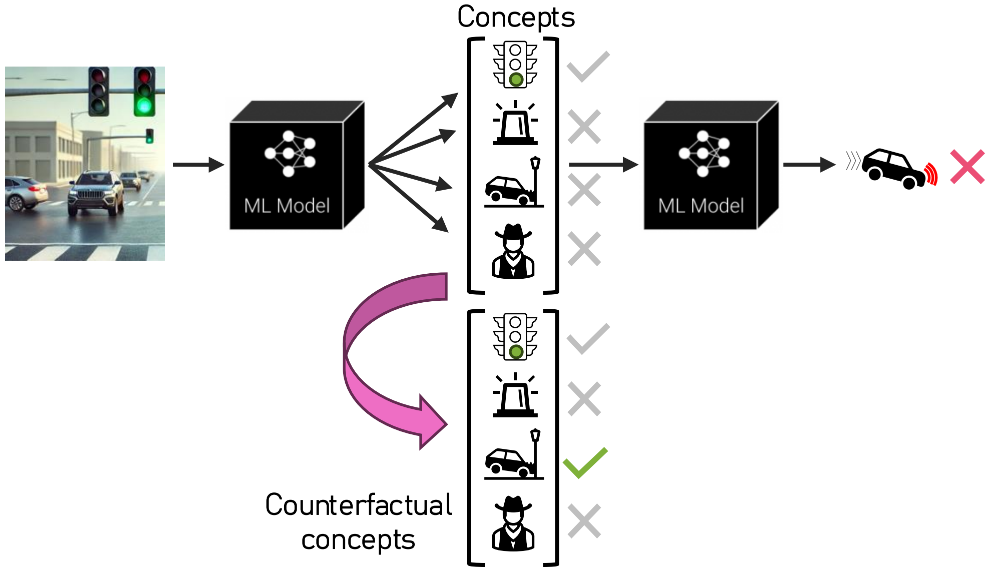
COUNTERFACTUAL CBMS

ICLR25

ICML22

Proposed Solution

Step 1: Generate **counterfactual concept activations**



Counterfactual

What would have been predicted in the same circumstance had a car crash be seen?

$P(brake \mid light, crash)$

Intervention

What if I set the light color to red?

$P(brake \mid do(light))$

Association

What if the model sees a green light?

$P(brake \mid light)$

[1] Dominici et al. "Counterfactual Concept Bottleneck Models." ICLR (2025).
[2] Abid et al. "Meaningfully debugging model mistakes using conceptual counterfactual explanations." ICML (2022).

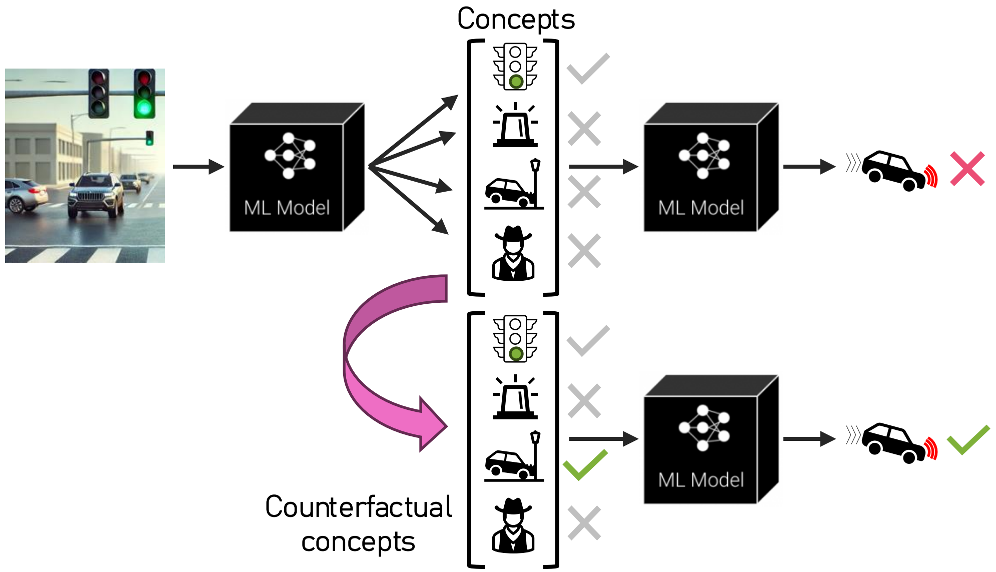
COUNTERFACTUAL CBMS

ICLR25

ICML22

Proposed Solution

Step 2: Compute causal effect on the task!



Counterfactual

What would have been predicted in the same circumstance had a car crash be seen?

$$P(brake \mid light, crash)$$

Intervention

What if I set the light color to red?

$$P(brake \mid do(light))$$

Association

What if the model sees a green light?

$$P(brake \mid light)$$

[1] Dominici et al. "Counterfactual Concept Bottleneck Models." ICLR (2025).

[2] Abid et al. "Meaningfully debugging model mistakes using conceptual counterfactual explanations." ICML (2022).

DIRECT COUNTERFACTUAL DEPENDENCE

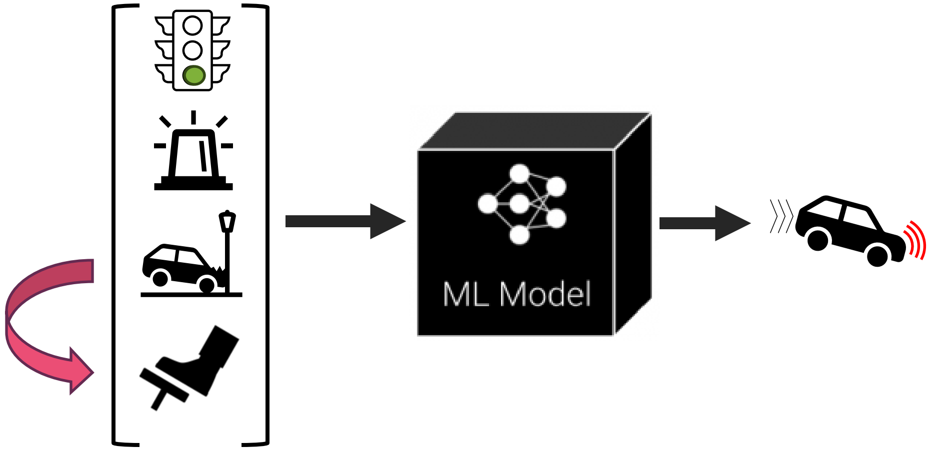
So far, we have been making 2 strong assumptions…

DIRECT COUNTERFACTUAL DEPENDENCE

So far, we have been making 2 strong assumptions:

- Concepts are mutually independent

Intervening on “car crash” does not increase the likelihood of hitting the brakes!



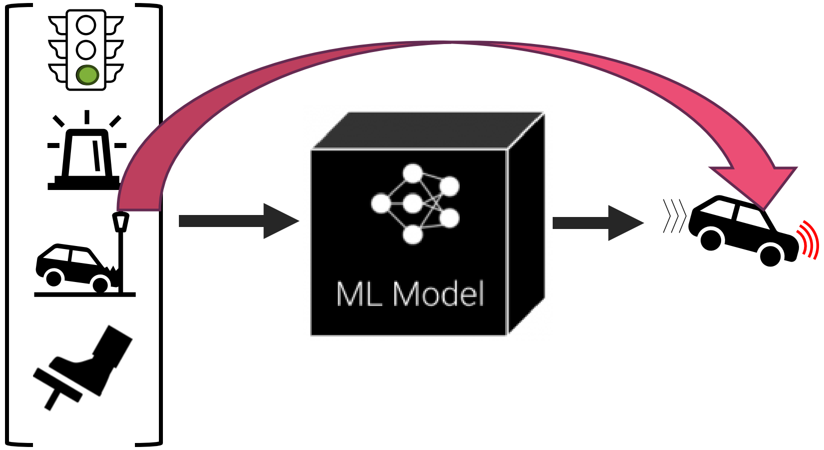
DIRECT COUNTERFACTUAL DEPENDENCE

So far, we have been making 2 strong assumptions:

- Concepts are mutually independent
- Concepts are direct causes of the task

Intervening on "car crash" does not increase the likelihood of hitting the brakes!

Intervening on "car crash" directly causes the car to brake!



CONCEPT GRAPH MODELS

Limitation Being Addressed

CBMs (as most XAI methods) assume direct counterfactual dependence!

ICLR25

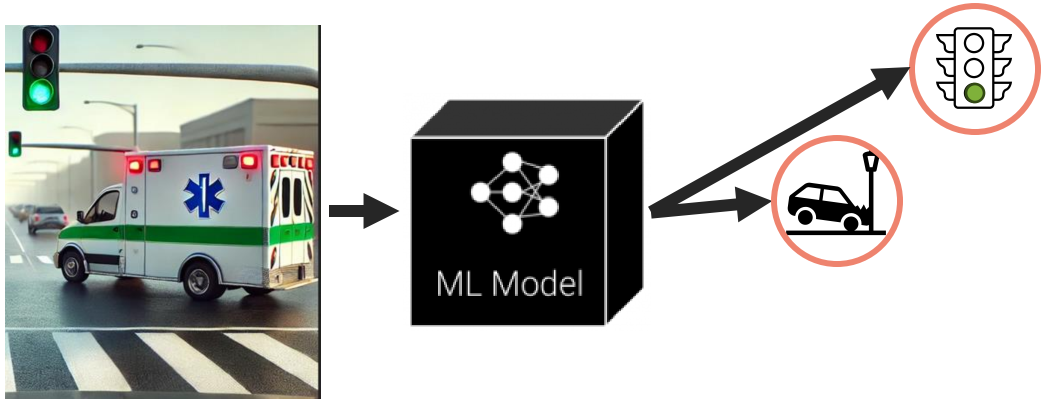
CLeaR24

[1] Dominici et al. "Causal Concept Graph Models: Beyond Causal Opacity in Deep Learning." ICLR 2025.
[2] Moreira et al. "Diconstruct: Causal concept-based explanations through black-box distillation." CLeaR 2024.

CONCEPT GRAPH MODELS

Proposed Solution

Enforce inference through a concept graph!



ICLR25

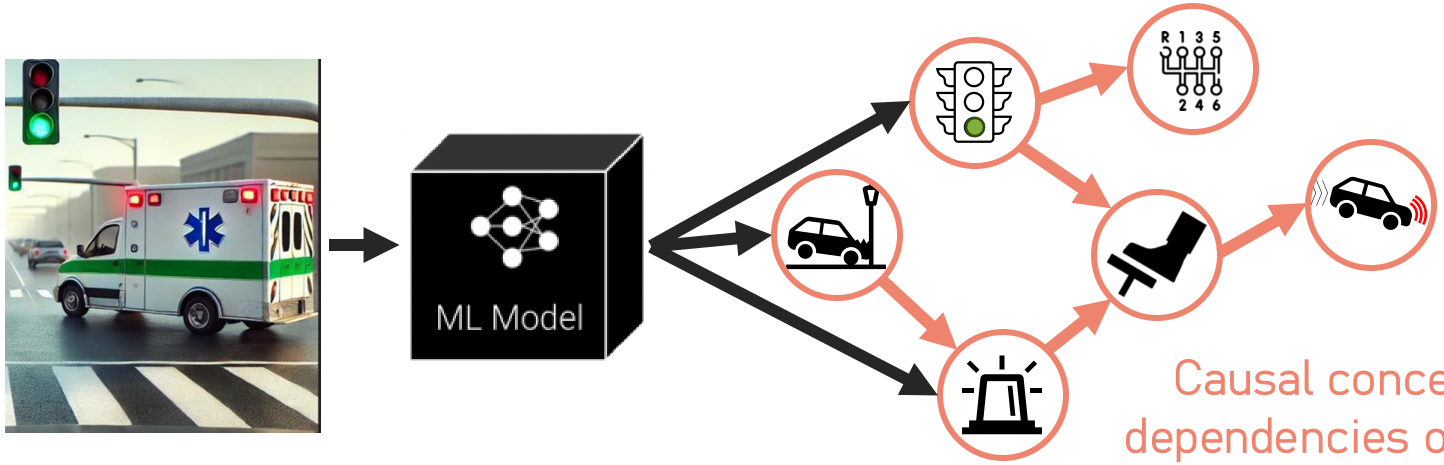
CLeaR24

[1] Dominici et al. "Causal Concept Graph Models: Beyond Causal Opacity in Deep Learning." ICLR 2025.
[2] Moreira et al. "Diconstruct: Causal concept-based explanations through black-box distillation." CLeaR 2024.

CONCEPT GRAPH MODELS

Proposed Solution

Enforce inference through a concept graph!



Causal concept dependencies of the inference mechanism

ICLR25

CLeaR24

---

[1] Dominici et al. "Causal Concept Graph Models: Beyond Causal Opacity in Deep Learning." ICLR 2025.

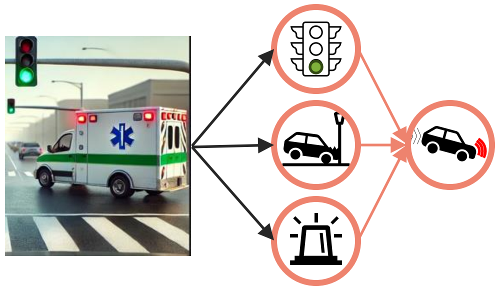
[2] Moreira et al. "Diconstruct: Causal concept-based explanations through black-box distillation." CLeaR 2024.

CONCEPT GRAPH MODELS

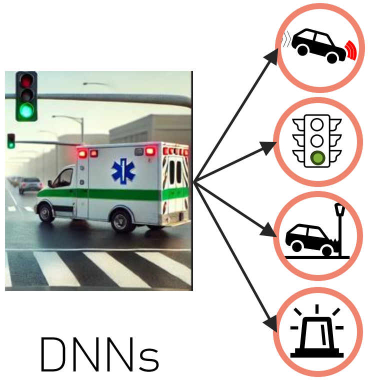
Proposed Solution

The concept graph can be:

- Given as a prior



CBMs



DNNs

ICLR25

CLeaR24

---

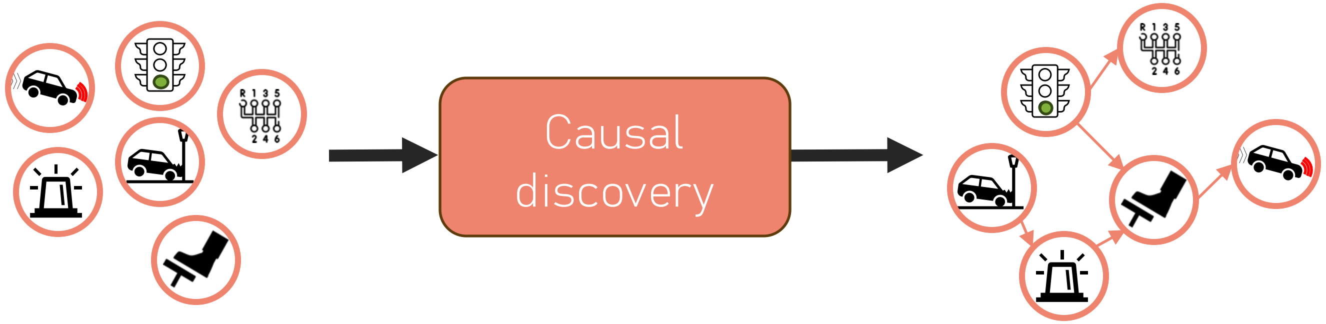
[1] Dominici et al. "Causal Concept Graph Models: Beyond Causal Opacity in Deep Learning." ICLR 2025.

[2] Moreira et al. "Diconstruct: Causal concept-based explanations through black-box distillation." CLeaR 2024.

CONCEPT GRAPH MODELS

Proposed Solution

The concept graph can be:
- Given as a prior
- Extracted from data with causal discovery techniques



Causal discovery

ICLR25

CLeaR24

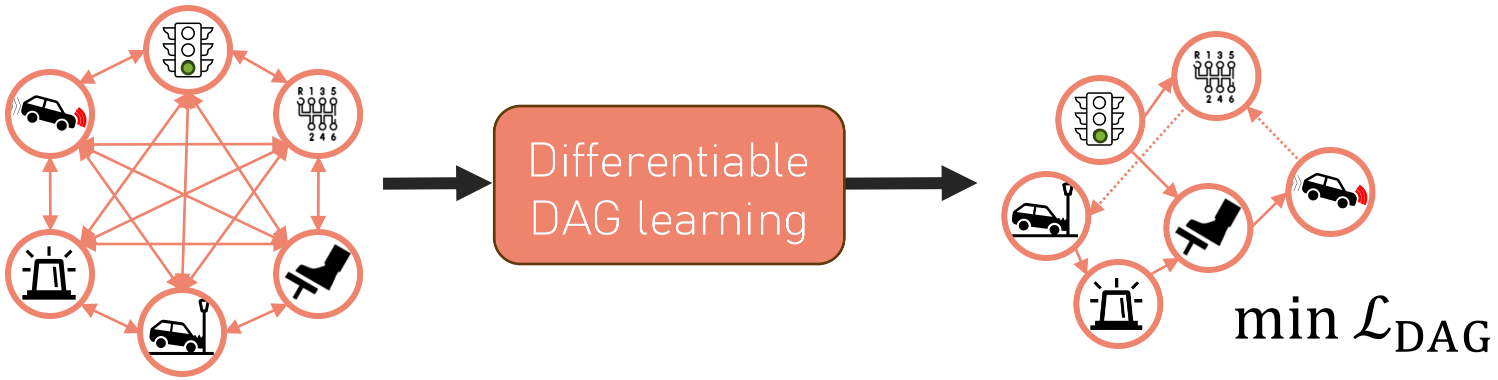
[1] Dominici et al. "Causal Concept Graph Models: Beyond Causal Opacity in Deep Learning." ICLR 2025.
[2] Moreira et al. "Diconstruct: Causal concept-based explanations through black-box distillation." CLeaR 2024.

CONCEPT GRAPH MODELS

Proposed Solution

The concept graph can be:

- Given as a prior
- Extracted from data with causal discovery techniques
- Obtained with differentiable DAG learning



Differentiable DAG learning

$\min \mathcal{L}_{\text{DAG}}$

ICLR25

CLeaR24

---

[1] Dominici, Gabriele et al. "Causal Concept Graph Models: Beyond Causal Opacity in Deep Learning." ICLR 2025.

[2] Moreira, Ricardo, et al. "Diconstruct: Causal concept-based explanations through black-box distillation." CLeaR 2024.

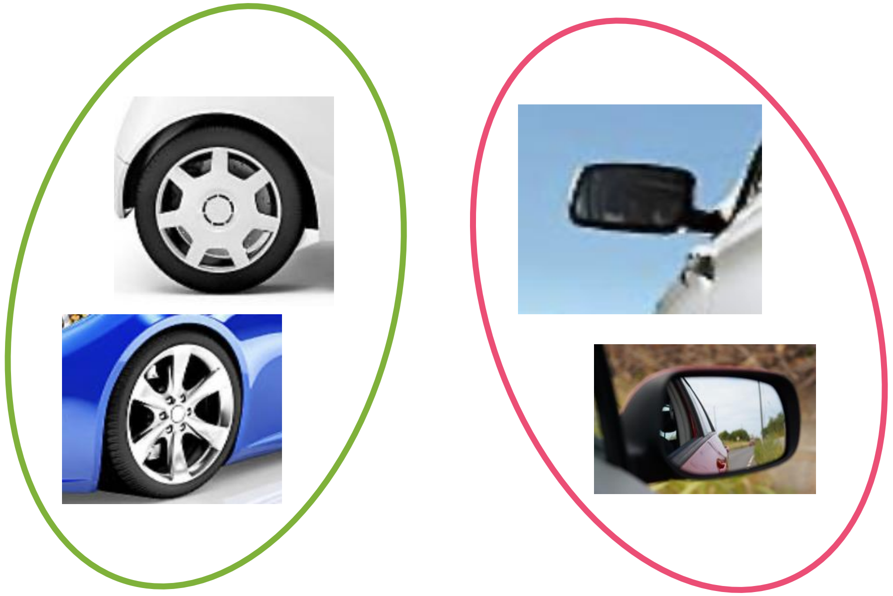
TUTORIAL OUTLINE

1. Introduction
2. Supervised Concept Learning
3. Concept Interventions
4. Q&A + Break
5. Unsupervised Concept Learning
6. Reasoning With Concepts
7. **Future Directions**
8. Q&A

AMAZING CONCEPTS & WHERE TO FIND THEM

Concept interpretability is not the first nor the only area focusing on concepts!

- Prototypes (clustering)

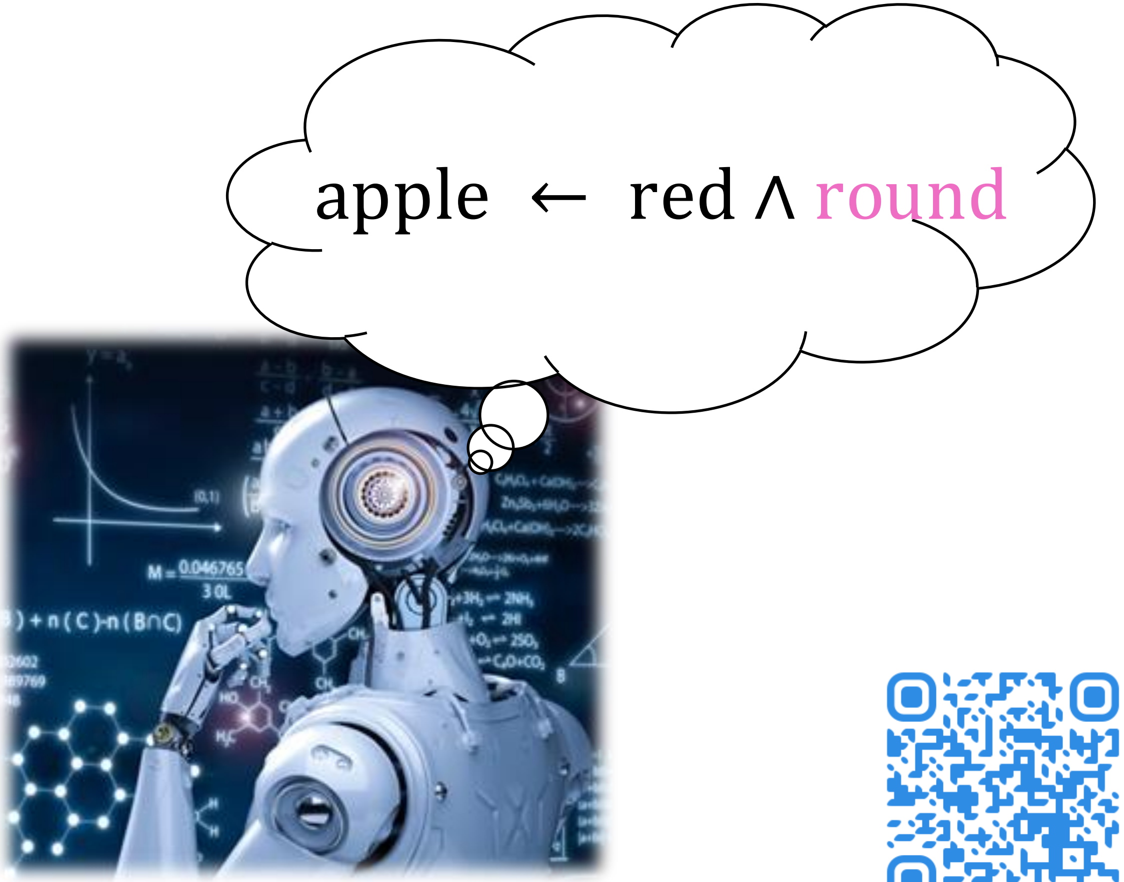


[1] Goguen "What is a concept? " International Conference on Conceptual Structures, 2005

AMAZING CONCEPTS & WHERE TO FIND THEM

Concept interpretability is not the first nor the only area focusing on concepts!

- Prototypes (clustering)
- Symbols (logic, neural-symbolic AI)



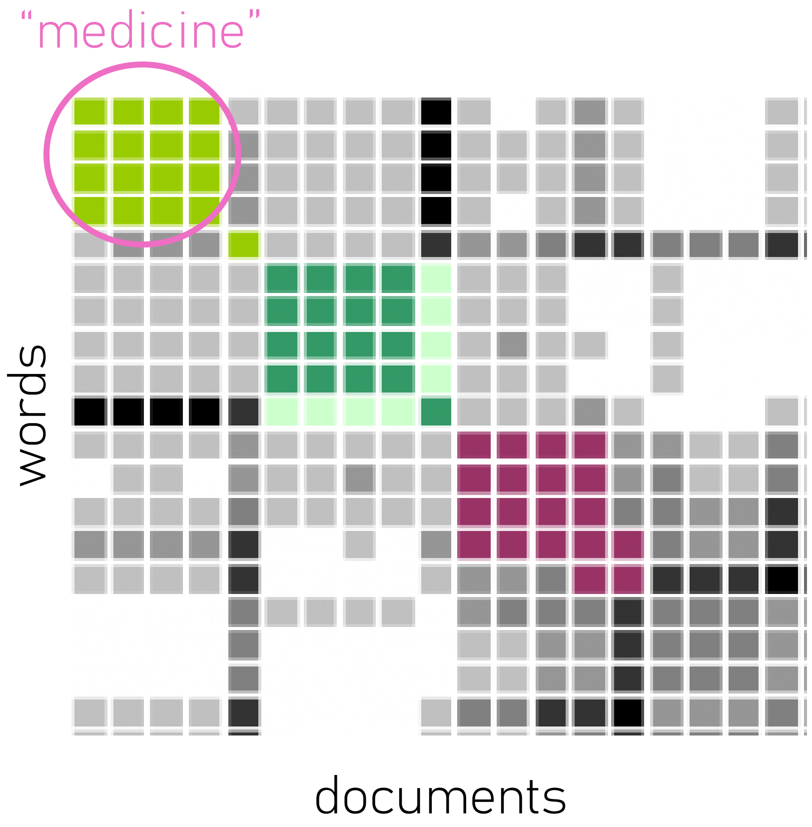
$$\text{apple} \leftarrow \text{red} \wedge \text{round}$$

[1] Goguen "What is a concept? " International Conference on Conceptual Structures, 2005

AMAZING CONCEPTS & WHERE TO FIND THEM

Concept interpretability is not the first nor the only area focusing on concepts!

- Prototypes (clustering)
- Symbols (logic, neural-symbolic AI)
- Topic models (semantic analysis)

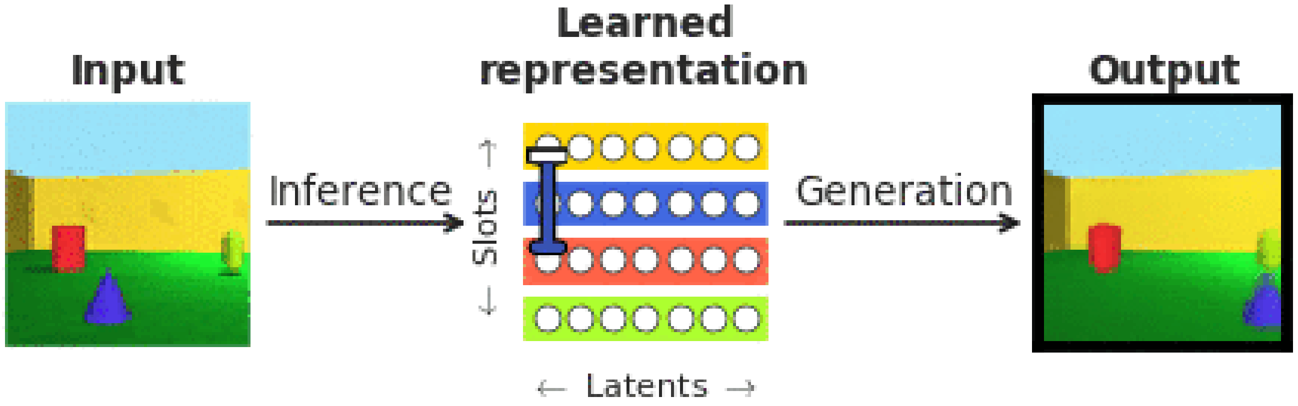


[1] Goguen "What is a concept? " International Conference on Conceptual Structures. 2005

AMAZING CONCEPTS & WHERE TO FIND THEM

Concept interpretability is not the first nor the only area focusing on concepts!

- Prototypes (clustering)
- Symbols (logic, neural-symbolic AI)
- Topic models (semantic analysis)
- Factors of variation (disentanglement learning)



[1] Goguen "What is a concept? " International Conference on Conceptual Structures. 2005.

AMAZING CONCEPTS & WHERE TO FIND THEM

Concept interpretability is not the first nor the only area focusing on concepts!

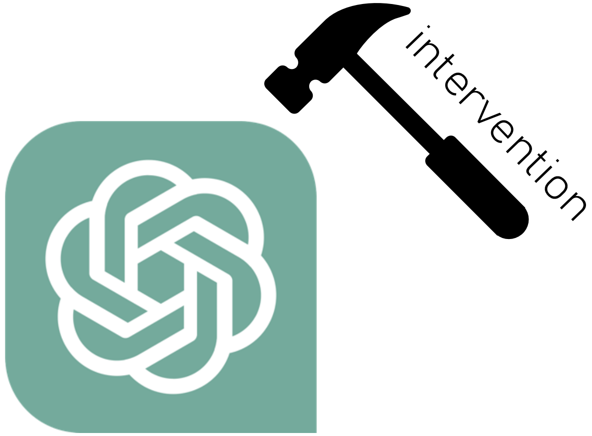
… but it has a few key differences:

- Focus on intervenability & different forms of transparency (semantic, functional, causal)
- For this reason, often different assumptions hold (e.g., concepts don't have to be independent as in disentanglement learning!)

OPEN CHALLENGES

Label-free models are currently not as reliable as supervised ones

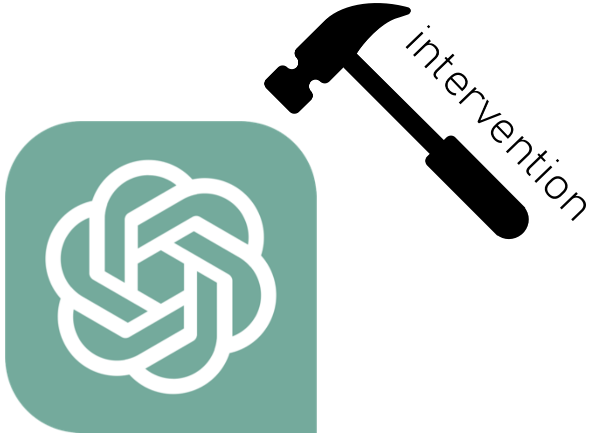
- How to effectively intervene in label-free settings?



OPEN CHALLENGES

Label-free models are currently not as reliable as supervised ones

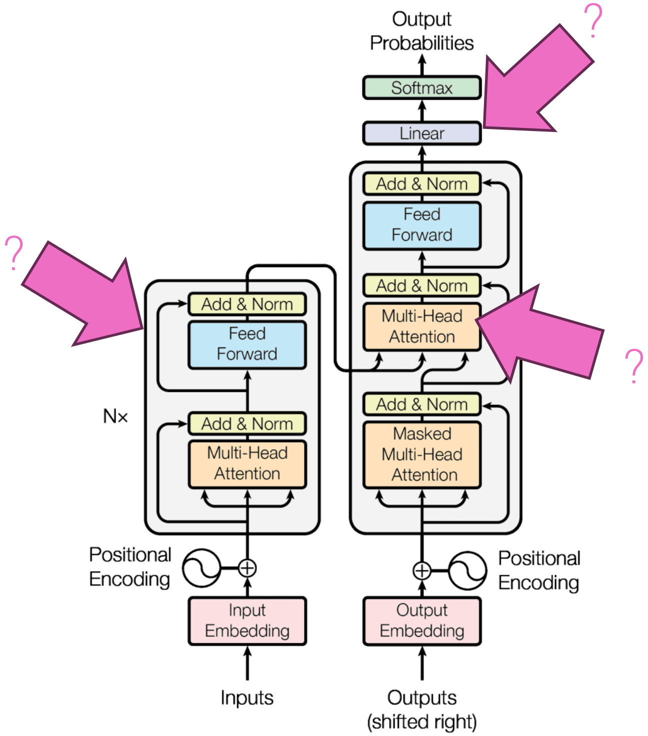
- **How** to effectively **intervene** in label-free settings?
- **How** to construct robust **annotations** without pre-trained domain-specific models?



OPEN CHALLENGES

Concept-based models are currently not designed nor integrated to scale to large models

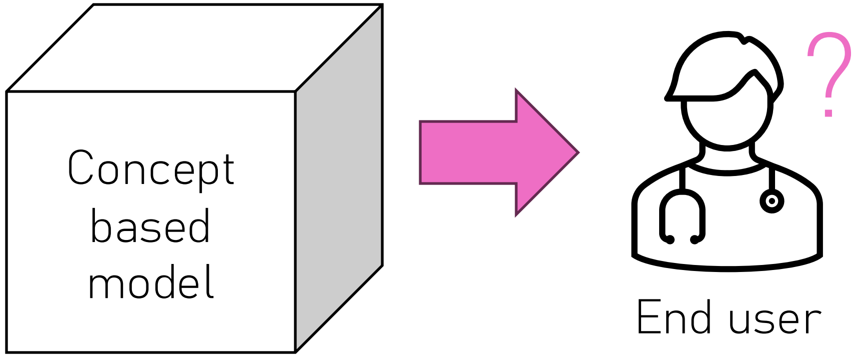
- Where (autoregressive, sentence, or paragraph) should we look for/place concepts in large models?
- Should large models reason based on concepts?



OPEN CHALLENGES

Concept-based models are currently not designed nor integrated to scale to large models

- Where (autoregressive, sentence, or paragraph) should we look for/place concepts in large models?
- Should large models reason based on concepts?
- Which guidelines should we follow to deploy concept-based models in the wild?



OPEN CHALLENGES

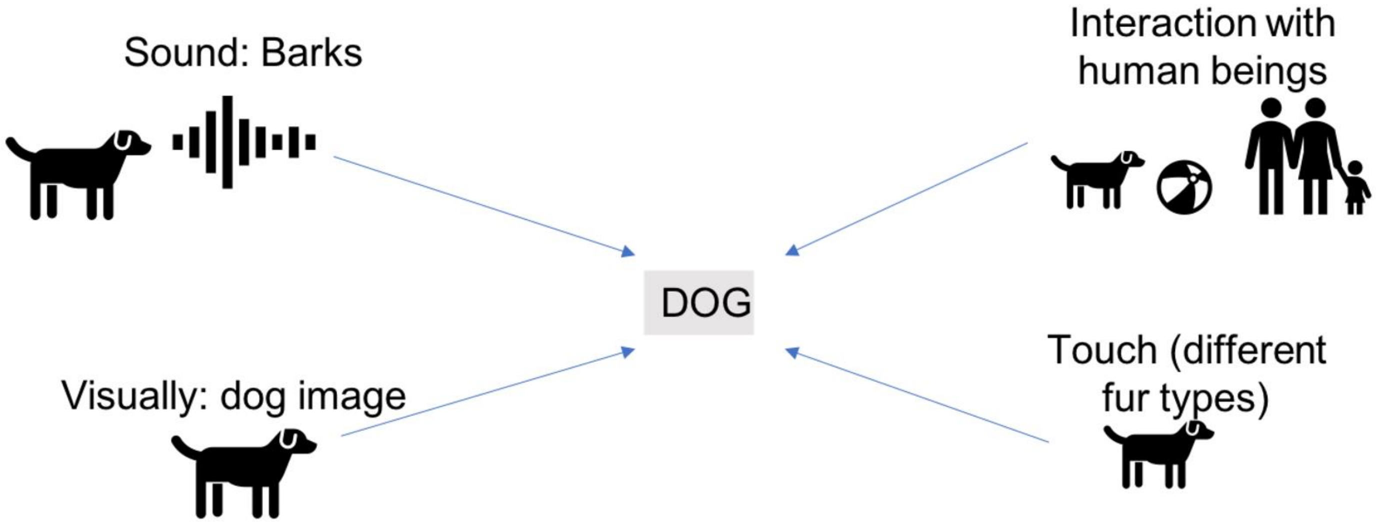
Some concepts are intrinsically hard to represent or intervene on

- How to deal with abstract (e.g., moral) or subjective concepts (e.g., aesthetics)?

OPEN CHALLENGES

Some concepts are intrinsically hard to represent or intervene on

- How to deal with abstract (e.g., moral) or subjective concepts (e.g., aesthetics)?
- How to construct and intervene on multi-modal concepts?

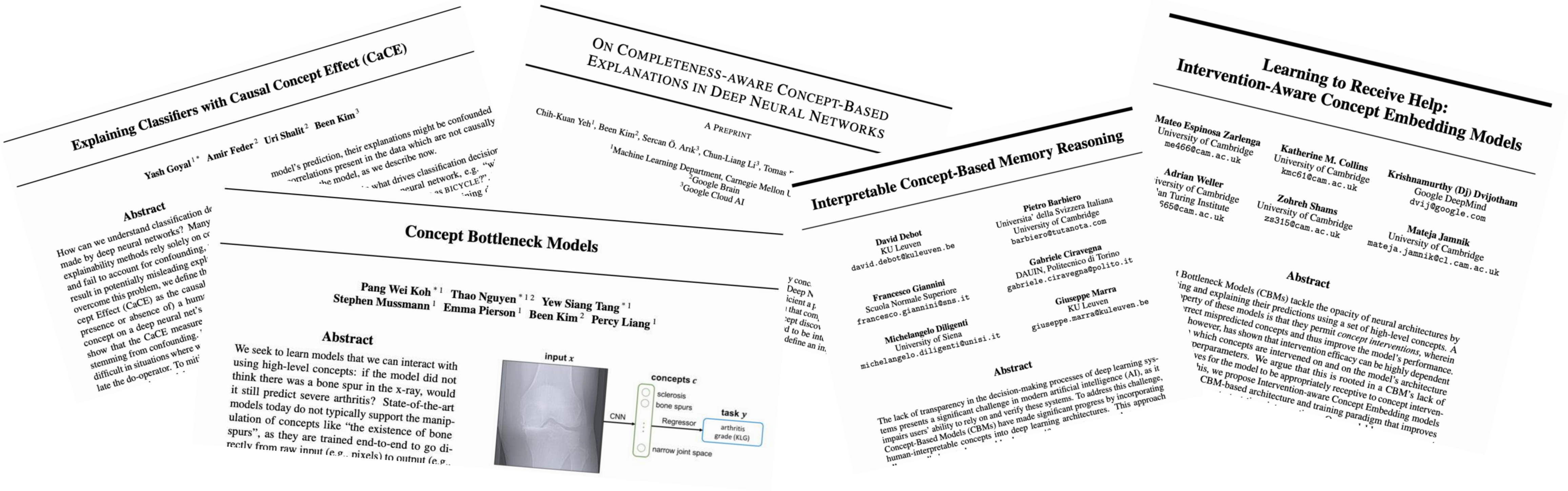


RESOURCES

conceptlearning.github.io/

Cornerstone papers highlighted in this presentation

Extended **bibliography** on the tutorial website and in the slide deck's appendix



Explaining Classifiers with Causal Concept Effect (CaCE)

Yash Goyal[1]* Amir Feder[2] Uri Shalit[2] Been Kim[3]

Abstract

How can we understand classification de
made by deep neural networks? Many
explainability methods rely solely on co
and fail to account for confounding,
result in potentially misleading expl
overcome this problem, we define th
cept Effect (CaCE) as the causal
presence or absence of) a huma
concept on a deep neural net's
show that the CaCE measure
stemming from confounding.
difficult in situations where w
late the do-operator. To mit

model's prediction, their explanations might be confounded
correlations present in the data which are not causally
the model, as we describe now.
what drives classification decisior
neural network, e.g. "w
as BICYCLE?".

On Completeness-aware Concept-Based Explanations in Deep Neural Networks

A Preprint

Chih-Kuan Yeh[1], Been Kim[2], Sercan Ö. Arık[3], Chun-Liang Li[3], Tomas

[1]Machine Learning Department, Carnegie Mellon U
[2]Google Brain
[3]Google Cloud AI

y conc
Deep N
icient a p
that com
ept discov
d to be int
define an in

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.,

---

Interpretable Concept-Based Memory Reasoning

David Debot
KU Leuven
david.debot@kuleuven.be

Pietro Barbiero
Universita' della Svizzera Italiana
University of Cambridge
barbiero@tutanota.com

Francesco Giannini
Scuola Normale Superiore
francesco.giannini@sns.it

Gabriele Ciravegna
DAUIN, Politecnico di Torino
gabriele.ciravegna@polito.it

Michelangelo Diligenti
University of Siena
michelangelo.diligenti@unisi.it

Giuseppe Marra
KU Leuven
giuseppe.marra@kuleuven.be

Abstract

The lack of transparency in the decision-making processes of deep learning systems presents a significant challenge in modern artificial intelligence (AI), as it impairs users' ability to rely on and verify these systems. To address this challenge, Concept-Based Models (CBMs) have made significant progress by incorporating human-interpretable concepts into deep learning architectures. This approach

---

Learning to Receive Help: Intervention-Aware Concept Embedding Models

Mateo Espinosa Zarlenga
University of Cambridge
me466@cam.ac.uk

Katherine M. Collins
University of Cambridge
kmc61@cam.ac.uk

Krishnamurthy (Dj) Dvijotham
Google DeepMind
dvij@google.com

Adrian Weller
iversity of Cambridge
an Turing Institute
565@cam.ac.uk

Zohreh Shams
University of Cambridge
zs315@cam.ac.uk

Mateja Jamnik
University of Cambridge
mateja.jamnik@cl.cam.ac.uk

Abstract

t Bottleneck Models (CBMs) tackle the opacity of neural architectures by
ing and explaining their predictions using a set of high-level concepts. A
perty of these models is that they permit *concept interventions*, wherein
rrect mispredicted concepts and thus improve the model's performance.
however, has shown that intervention efficacy can be highly dependent
which concepts are intervened on and on the model's architecture
perparameters. We argue that this is rooted in a CBM's lack of
ves for the model to be appropriately receptive to concept interven-
his, we propose Intervention-aware Concept Embedding models
CBM-based architecture and training paradigm that improves

RESOURCES

PyC

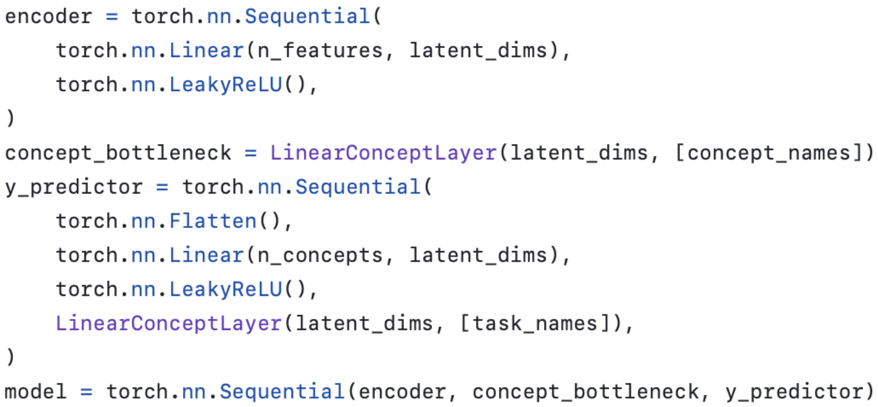
@github

@medium

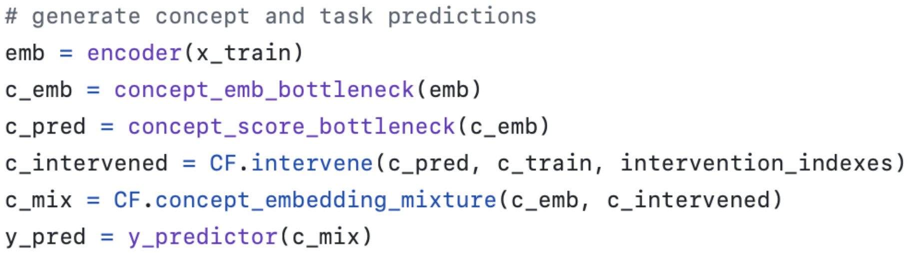
Some XAI libraries implement concept-based techniques (check out tutorial website!)

We are working on PyTorch Concepts (PyC), a library dedicated to concept-based interpretability

- APIs are designed to implement existing models, but also to support the development of new ones
- Currently supports concept-based: data types, layers, interventions, metrics, models
- The PyC team is publishing hands-on tutorials on Medium!



```
encoder = torch.nn.Sequential(
    torch.nn.Linear(n_features, latent_dims),
    torch.nn.LeakyReLU(),
)
concept_bottleneck = LinearConceptLayer(latent_dims, [concept_names])
y_predictor = torch.nn.Sequential(
    torch.nn.Flatten(),
    torch.nn.Linear(n_concepts, latent_dims),
    torch.nn.LeakyReLU(),
    LinearConceptLayer(latent_dims, [task_names]),
)
model = torch.nn.Sequential(encoder, concept_bottleneck, y_predictor)
```



```
# generate concept and task predictions
emb = encoder(x_train)
c_emb = concept_emb_bottleneck(emb)
c_pred = concept_score_bottleneck(c_emb)
c_intervened = CF.intervene(c_pred, c_train, intervention_indexes)
c_mix = CF.concept_embedding_mixture(c_emb, c_intervened)
y_pred = y_predictor(c_mix)
```

[1] https://github.com/pyc-team/pytorch_concepts
[2] https://medium.com/@pyc.devteam

A FEW THINGS TO BRING BACK HOME!

Thank you for your time! Before leaving, remember that concept-based interpretability:

- is connected to other AI areas, but it focuses on specific research questions (intervenability and different forms of opacity)
- can make things easier (human interaction)… or worse (need for annotations)
- is a relatively young research field, so there's a lot of work to do for all of us!

Read our Medium stories to implement your first concept-based model in <15 minutes!

PyC

conceptlearning.github.io/