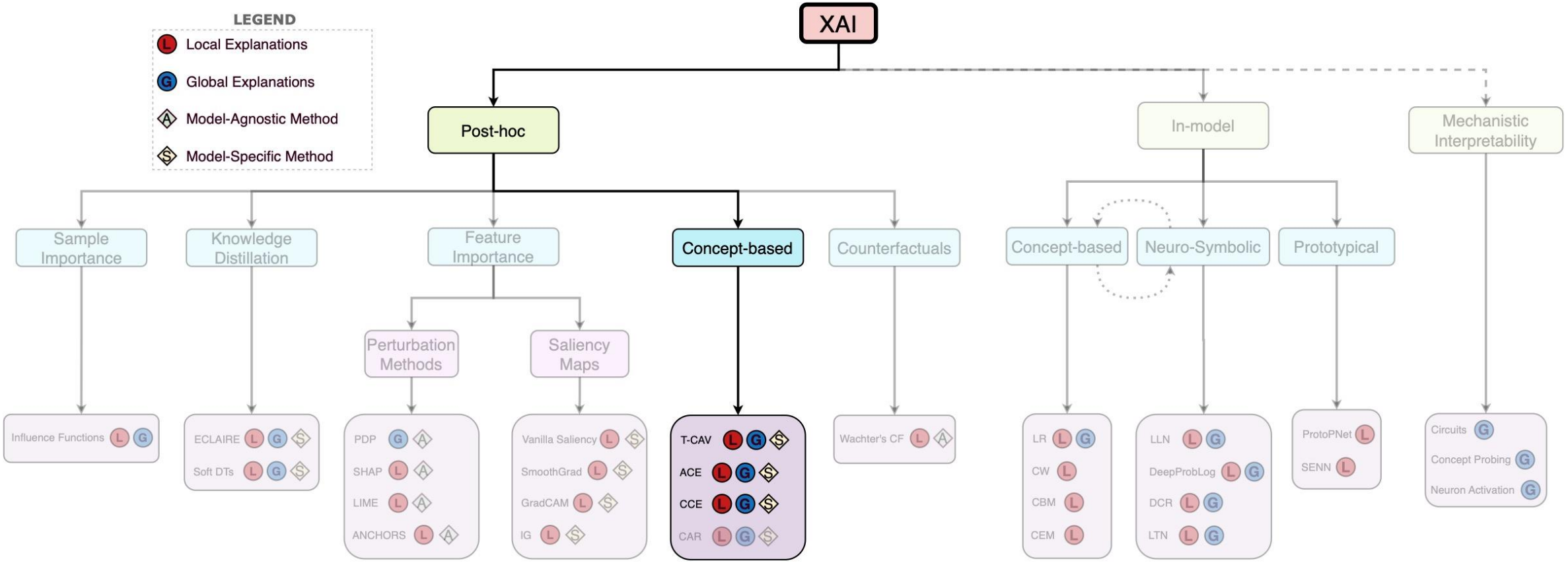
EXPLAINABLE ARTIFICIAL INTELLIGENCE

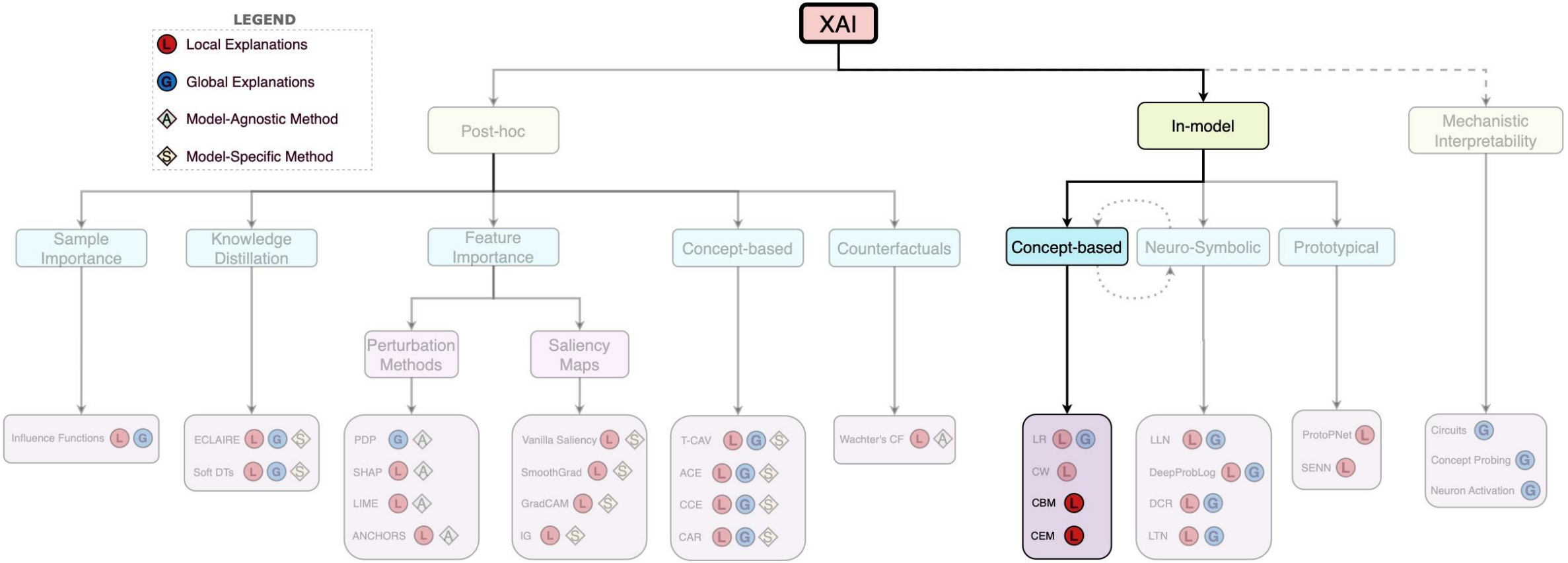
L193 – Lecture 4 – Lent 2025

UNIVERSITY OF CAMBRIDGE

WHERE WE LEFT OFF LAST TIME



TODAY: IN-MODEL EXPLAINABILITY



OUR XAI STORY SO FAR

We have focused on exploring post-hoc XAI methods

Taken from memecenter.com

OUR XAI 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!

OUR XAI 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 (Krishna et al., [1])!

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

OUR XAI STORY SO FAR

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

2. They are unable to capture causal relationships between input features, concepts, and output labels

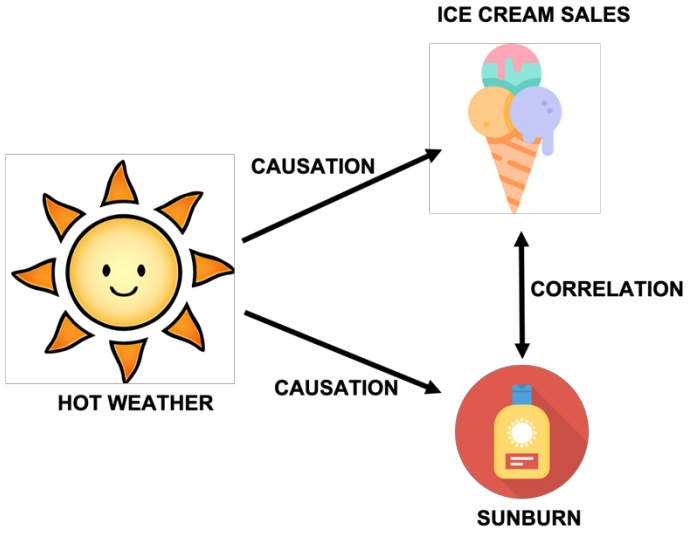


Image taken from https://www.royriachi.com/2019/02/correlational-and-causal-relationships.html

OUR XAI STORY SO FAR

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

3. Explanations are prone to confirmation bias (Bertrand et al., [1])!

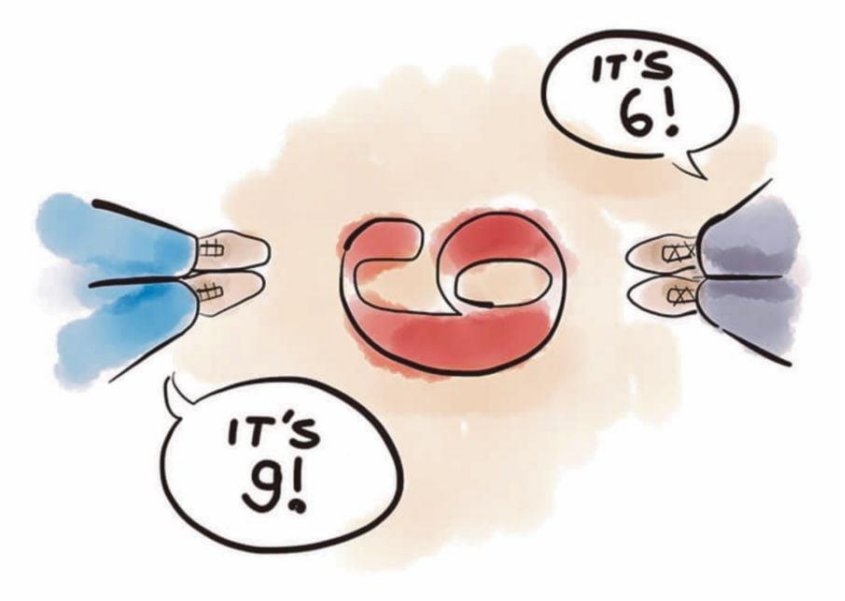


Image taken from "Confirmation Bias and the new Malaysia" by Datuk Steven Wong (New Straits Times)

[1] Bertrand, Astrid, et al. "How cognitive biases affect XAI-assisted decision-making: A systematic review." Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society. 2022.

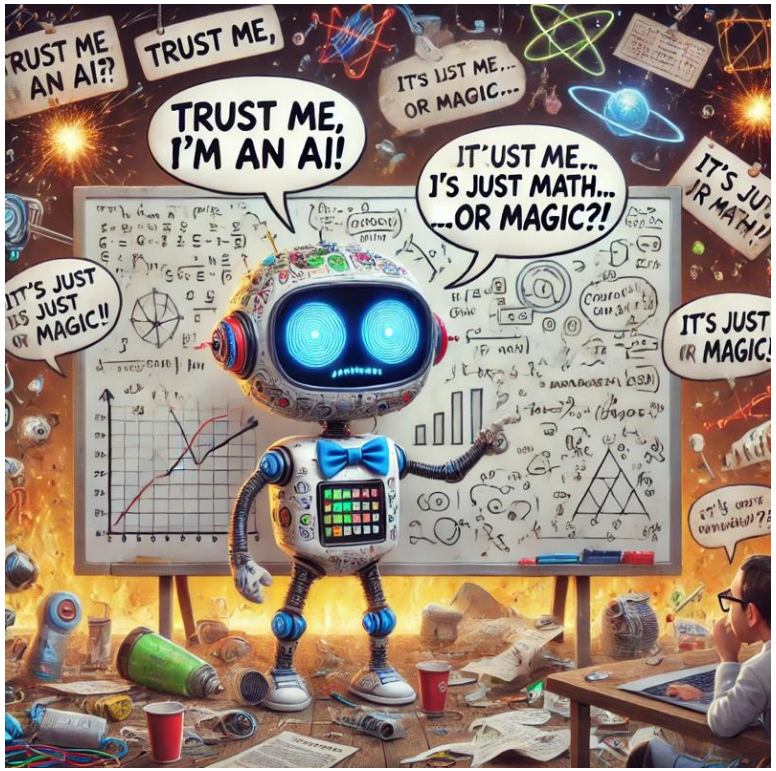
SO, WHO WE'RE GOING TO CALL?

In-model Explainable Neural Architectures!

IN-MODEL EXPLAINABLE MODELS

GOING IN-MODEL

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



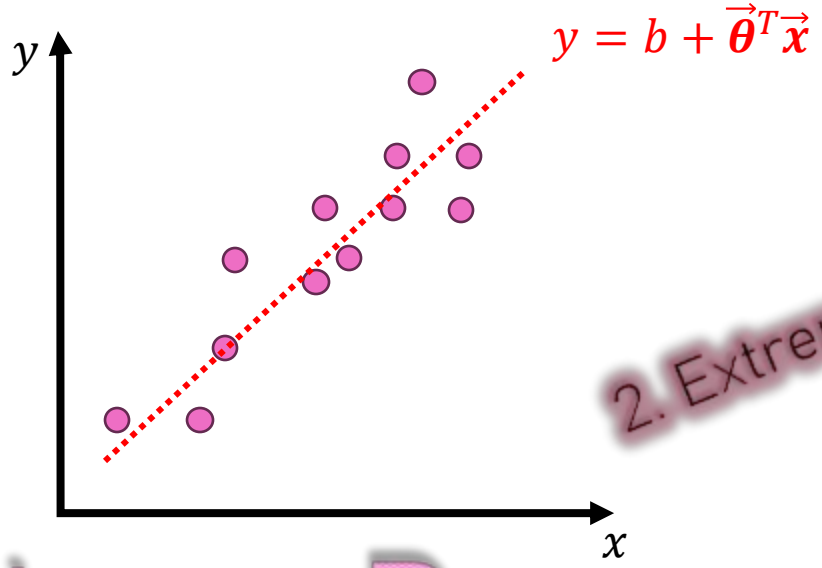
GOING IN-MODEL

There are a few main ways to achieve this:

GOING IN-MODEL

There are a few main ways to achieve this:

1. Linearity (e.g., logistic regression)



1. Interpretable!

2. Extremely simple!

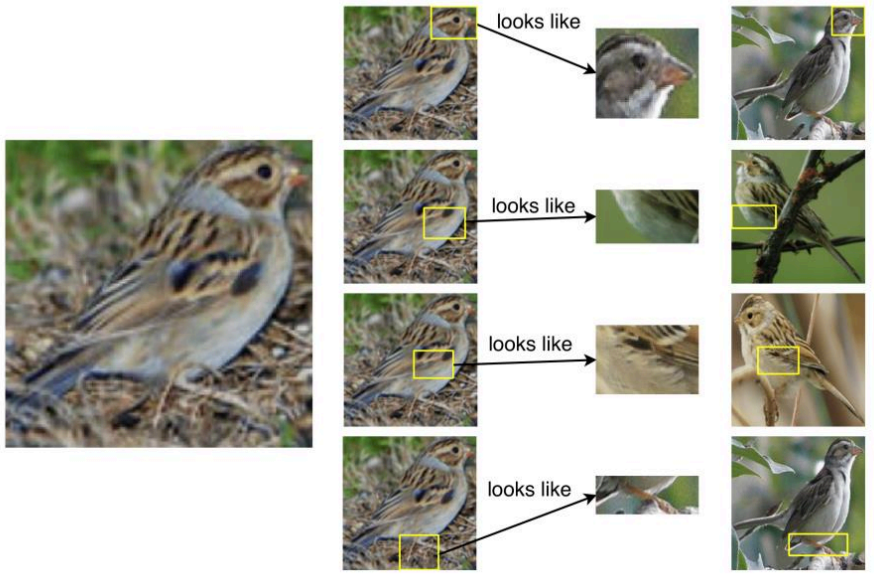
The Linear Regressor

3. Widely loved, understood, and revered!

GOING IN-MODEL

There are a few main ways to achieve this:

1. Linearity (e.g., logistic regression)
2. Prototypical explanations (e.g., "this looks like that")



GOING IN-MODEL

There are a few main ways to achieve this:

1. Linearity (e.g., logistic regression)
2. Prototypical explanations (e.g., "this looks like that")
3. And... concepts!

Today, we will focus on concept-based interpretable architectures but we will discuss other alternatives in future lectures!

CONCEPT BOTTLENECK MODELS (CBMS)

KOH ET AL. (ICML 2020)

Almost all concept-based interpretable architectures can be framed in terms of a Concept Bottleneck Model [1], or a CBM for short!

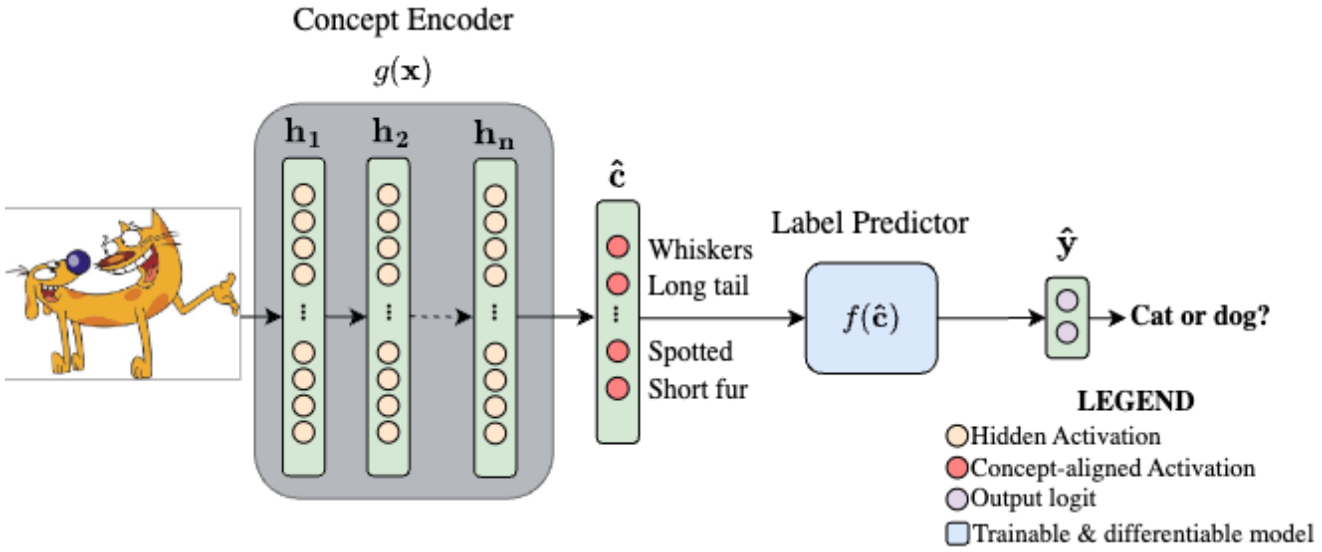
[1] Koh, Pang Wei, et al. "Concept bottleneck models." *International Conference on Machine Learning*. PMLR, 2020.

CONCEPT BOTTLENECK MODELS (CBMS)

KOH ET AL. (ICML 2020)

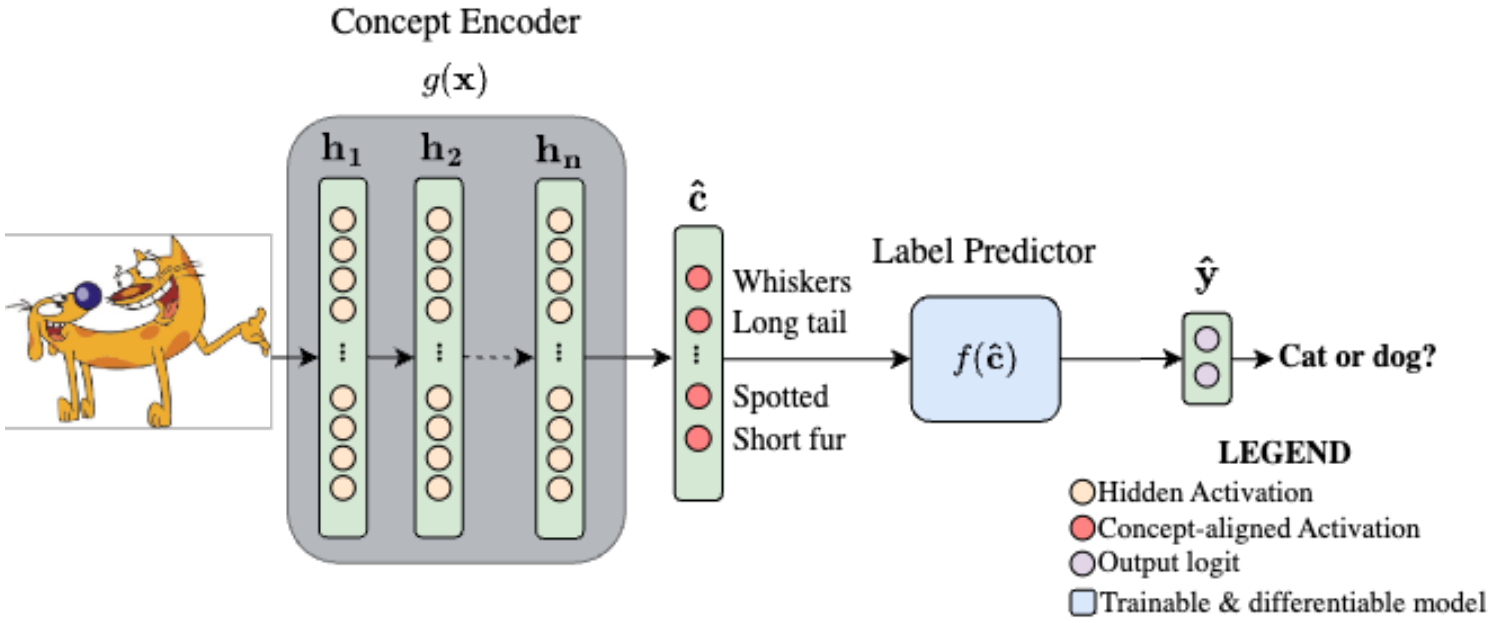
CBMs force some of their latent spaces to be aligned to known concepts by composing two functions:

1. The first part will predict concept activations from the input features (*concept encoder* $g(\boldsymbol{x}) = \hat{\boldsymbol{c}}$)
2. The second part will predict an task label from the predicted concepts (*label predictor* $f(\hat{\boldsymbol{c}}) = \hat{y}$)



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

UNDERLYING ASSUMPTIONS



Assumptions for Concept Bottleneck Models (CBMs):

1. Each sample is annotated with a task label $y \in \{0, 1, \cdots, L-1\}$
2. Each sample is annotated with a vector $\boldsymbol{c} \in \{0,1\}^k$ of k binary concepts

TRAINING A CBM

How would you train such a model given a concept-annotated dataset $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, \boldsymbol{c}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$?

(1) Independently

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

$\mathbb{E}_{(\boldsymbol{c}, y)\sim\mathcal{D}}[\mathrm{CE}(f(\boldsymbol{c}), y)]$

(2) Sequentially

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

Freeze g

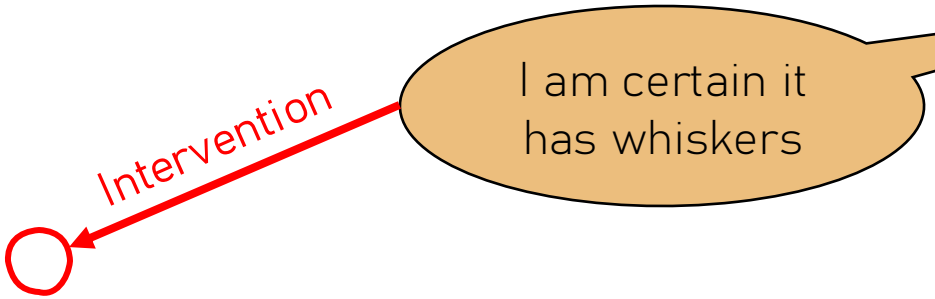
(b) $\mathbb{E}_{(\boldsymbol{x}, y)\sim\mathcal{D}}[\mathrm{CE}(f(g(x)), y)]$

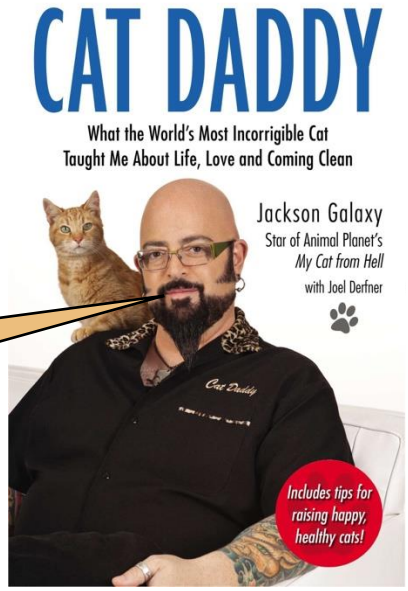
(3) Jointly

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

Question: what does λ control for in the joint training?

CONCEPT INTERVENTIONS

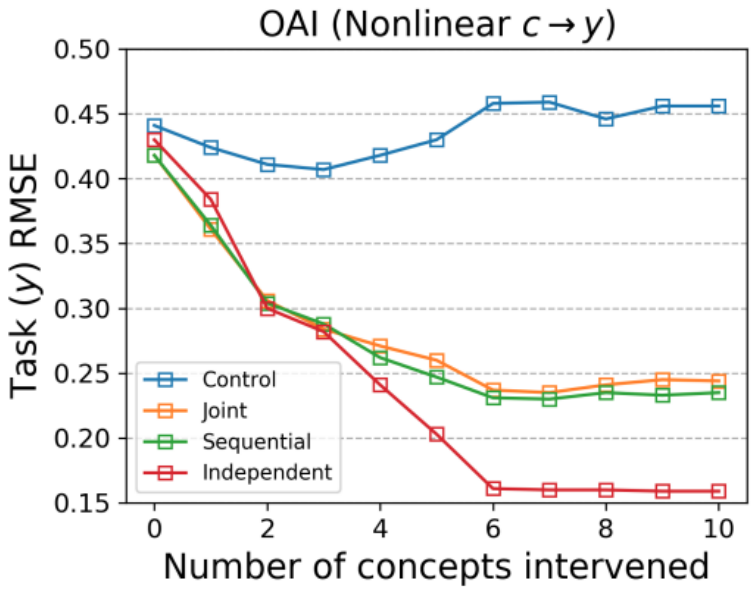


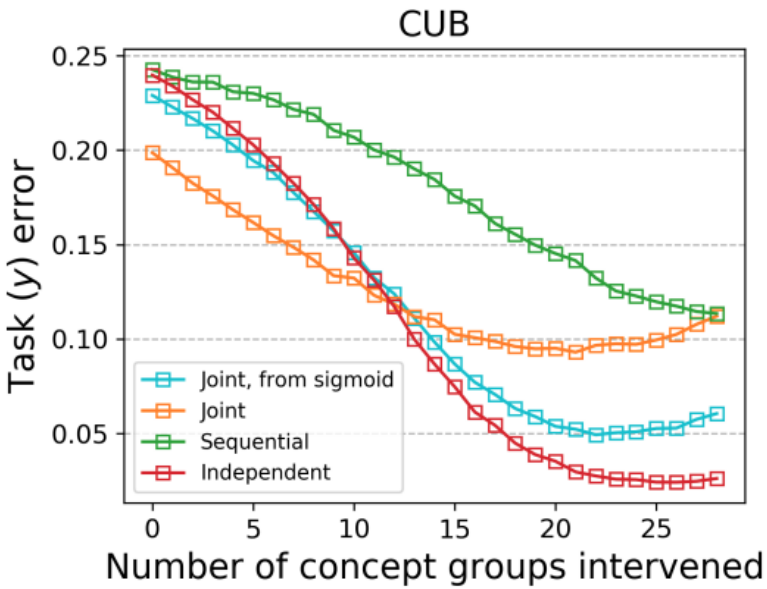


Our expert guest for today

CONCEPT INTERVENTIONS

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





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

ARE CBMS ALL WE NEED?

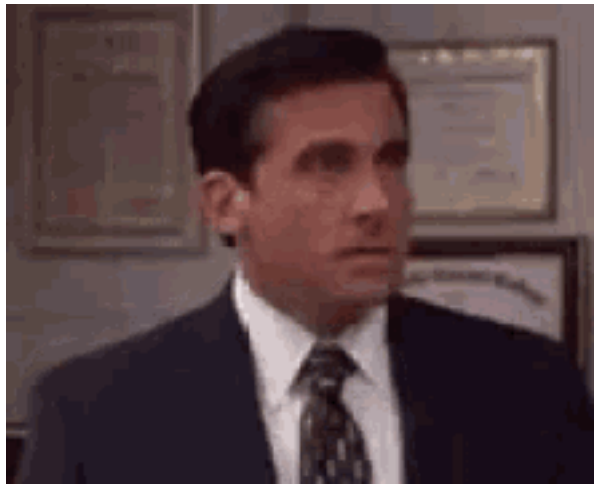
CBMs are great in a lot of ways:

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.

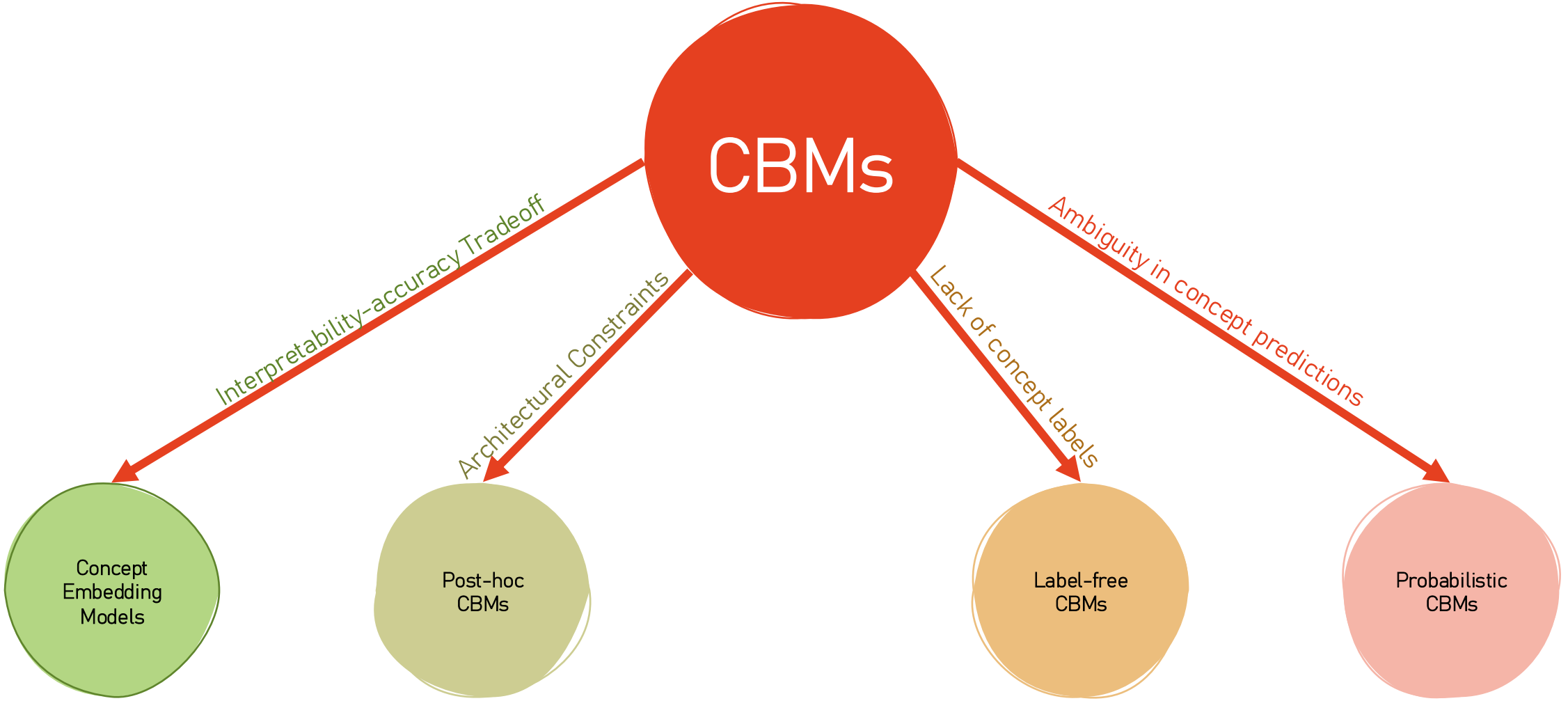
So, are we done?

Short Answer: No

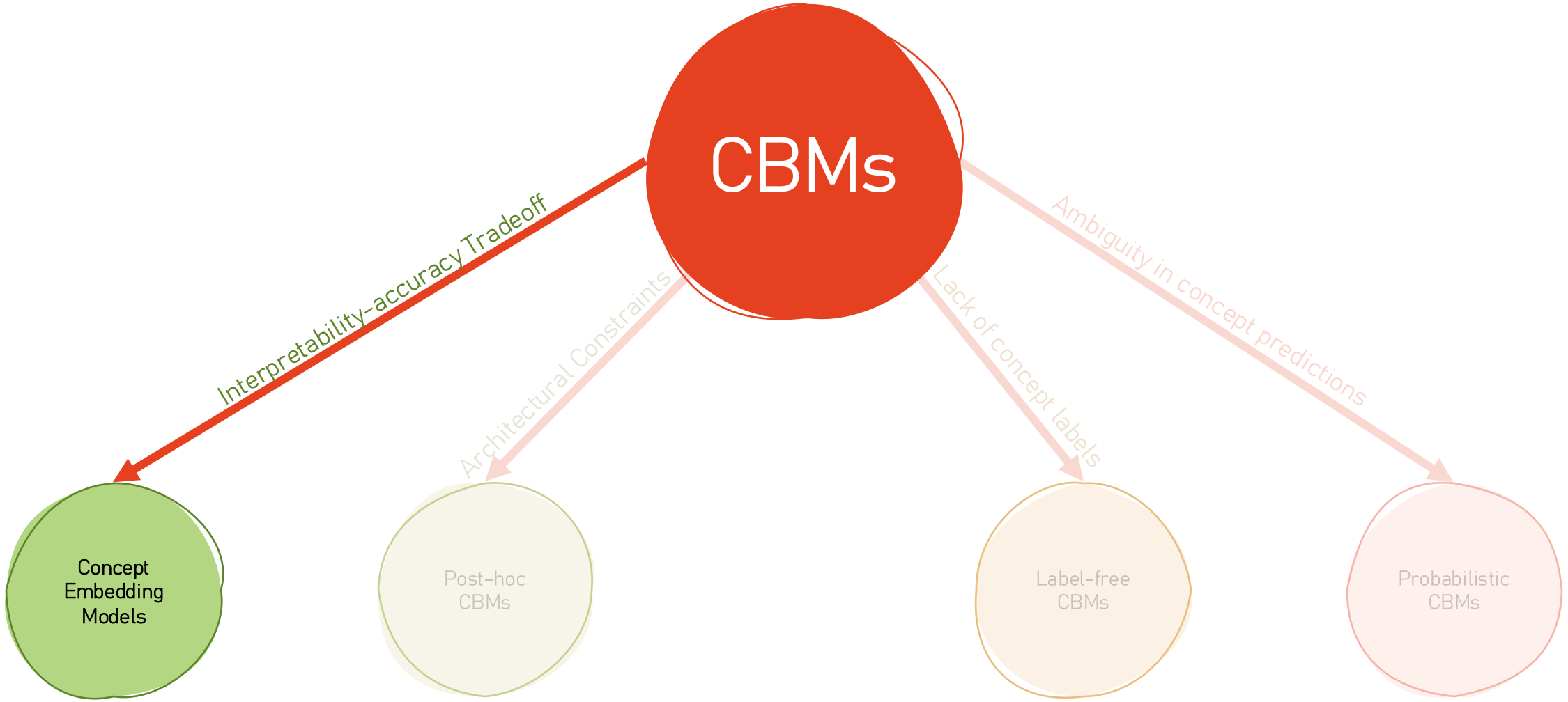
Long Answer:



INTRODUCING CBM'S FRIENDS



SPEED-DATING WITH CBM'S FRIENDS

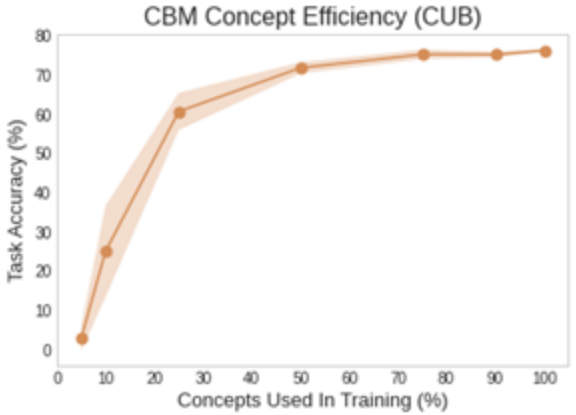


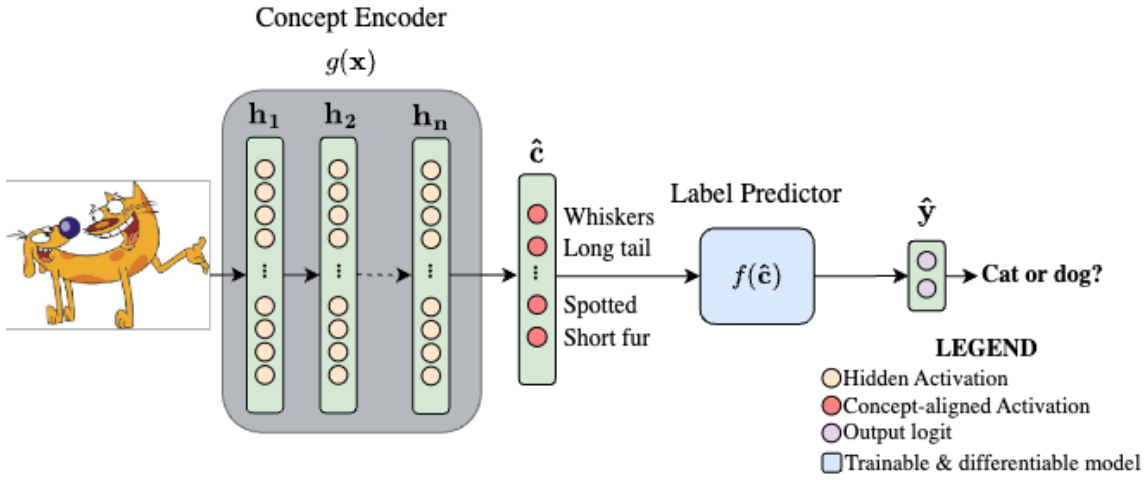
CONCEPT EMBEDDING MODELS

ESPINOSA ZARLENGA & BARBIERO ET AL. (NEURIPS 2022)

Limitation Being Addressed

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





Q#1: which training regime do you think would perform better when concepts are incomplete?

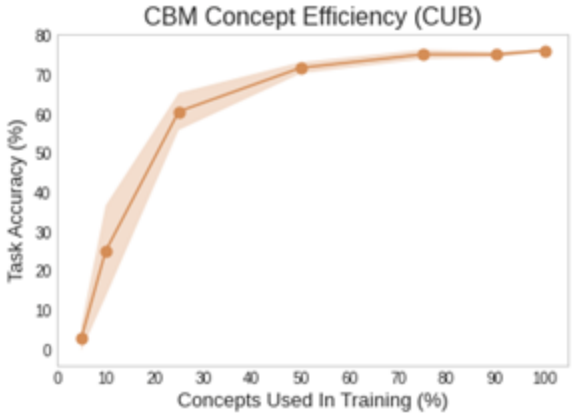
[1] Espinosa Zarlenga and Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

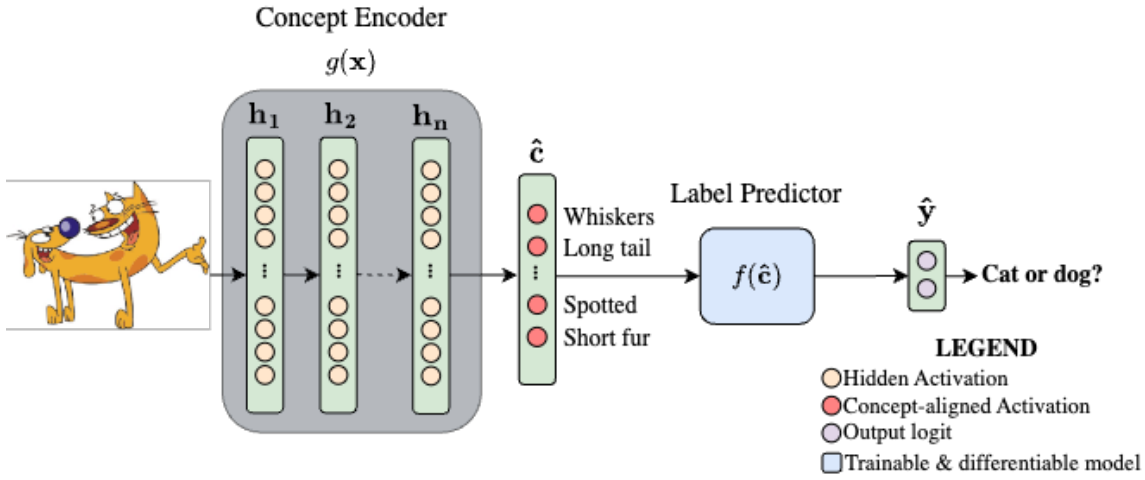
CONCEPT EMBEDDING MODELS

ESPINOSA ZARLENGA & BARBIERO ET AL. (NEURIPS 2022)

Limitation Being Addressed

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





Q#2: Why can't we just add a bypass from the input to the output?

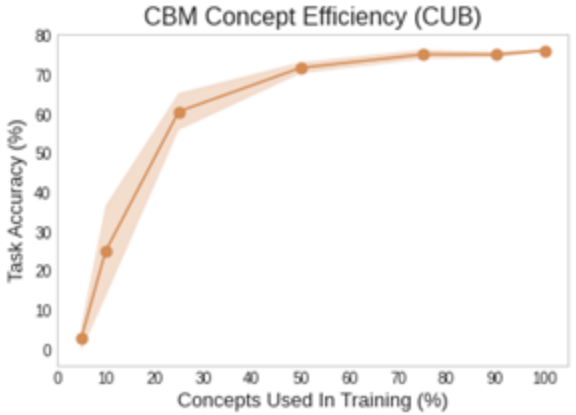
[1] Espinosa Zarlenga and Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

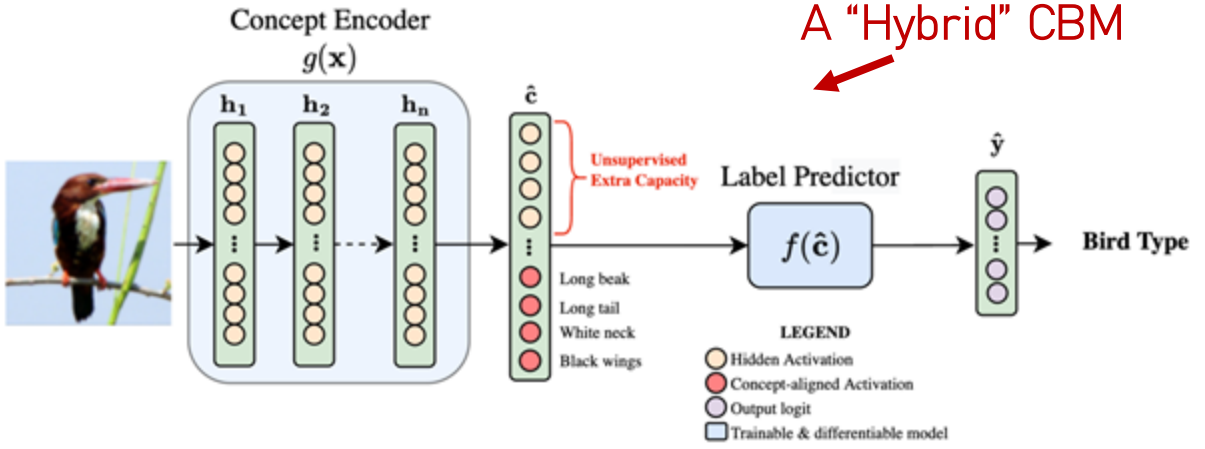
CONCEPT EMBEDDING MODELS

ESPINOSA ZARLENGA & BARBIERO ET AL. (NEURIPS 2022)

Limitation Being Addressed

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





Q#2: Why can't we just add a bypass from the input to the output?

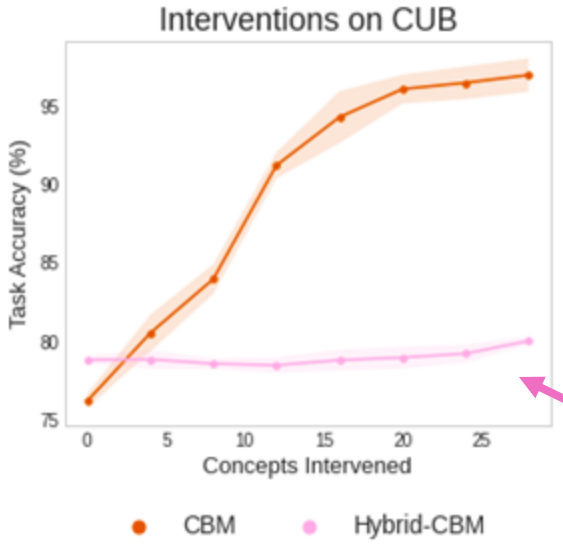
[1] Espinosa Zarlenga and Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

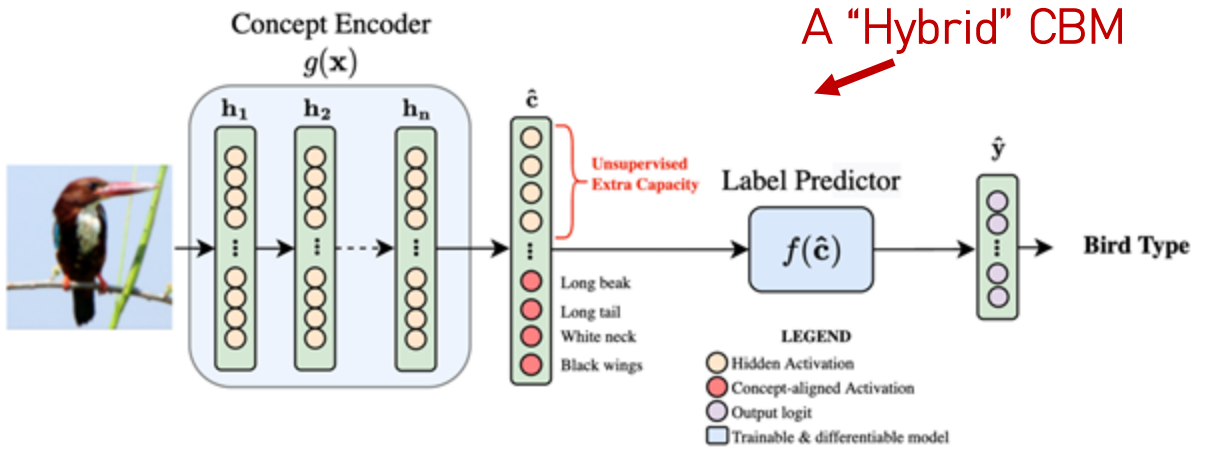
CONCEPT EMBEDDING MODELS

ESPINOSA ZARLENGA & BARBIERO ET AL. (NEURIPS 2022)

Limitation Being Addressed

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





Interventions do not necessarily work with Hybrid CBMs!

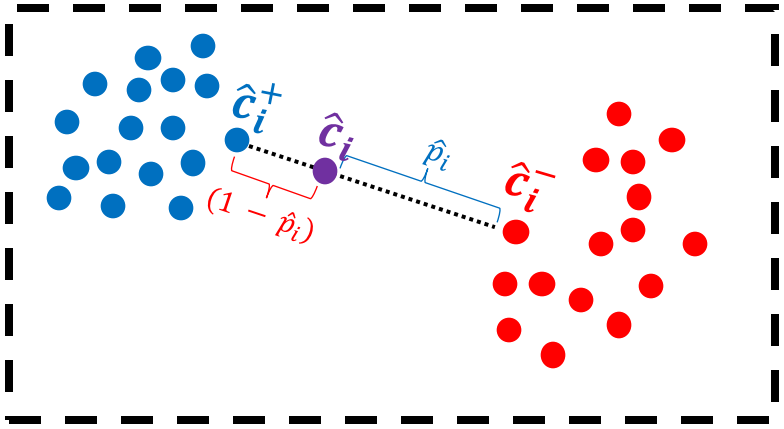
[1] Espinosa Zarlenga and Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

ESPINOSA ZARLENGA & BARBIERO ET AL. (NEURIPS 2022)

Proposed Solution

Learn two high-dimensional embeddings for each concept representing the concept when it is "on" and when it is "off"



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

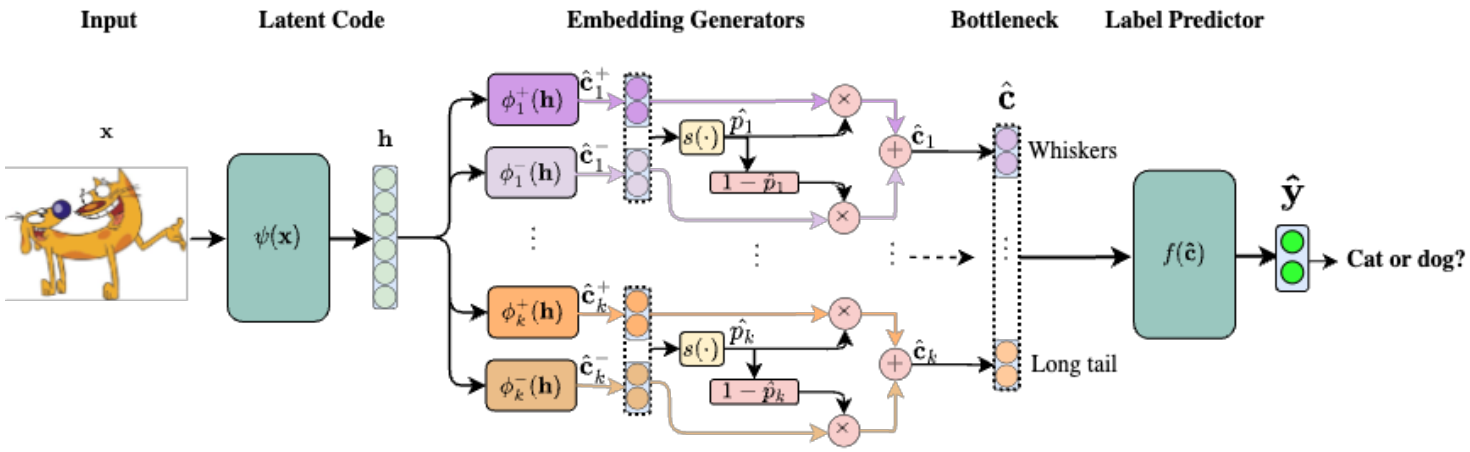
[1] Espinosa Zarlenga and Barbiero et al. "Concept embedding models: Beyond the accuracy-explainability trade-off." NeurIPS (2022)

CONCEPT EMBEDDING MODELS

ESPINOSA ZARLENGA & BARBIERO ET AL. (NEURIPS 2022)

Proposed Solution

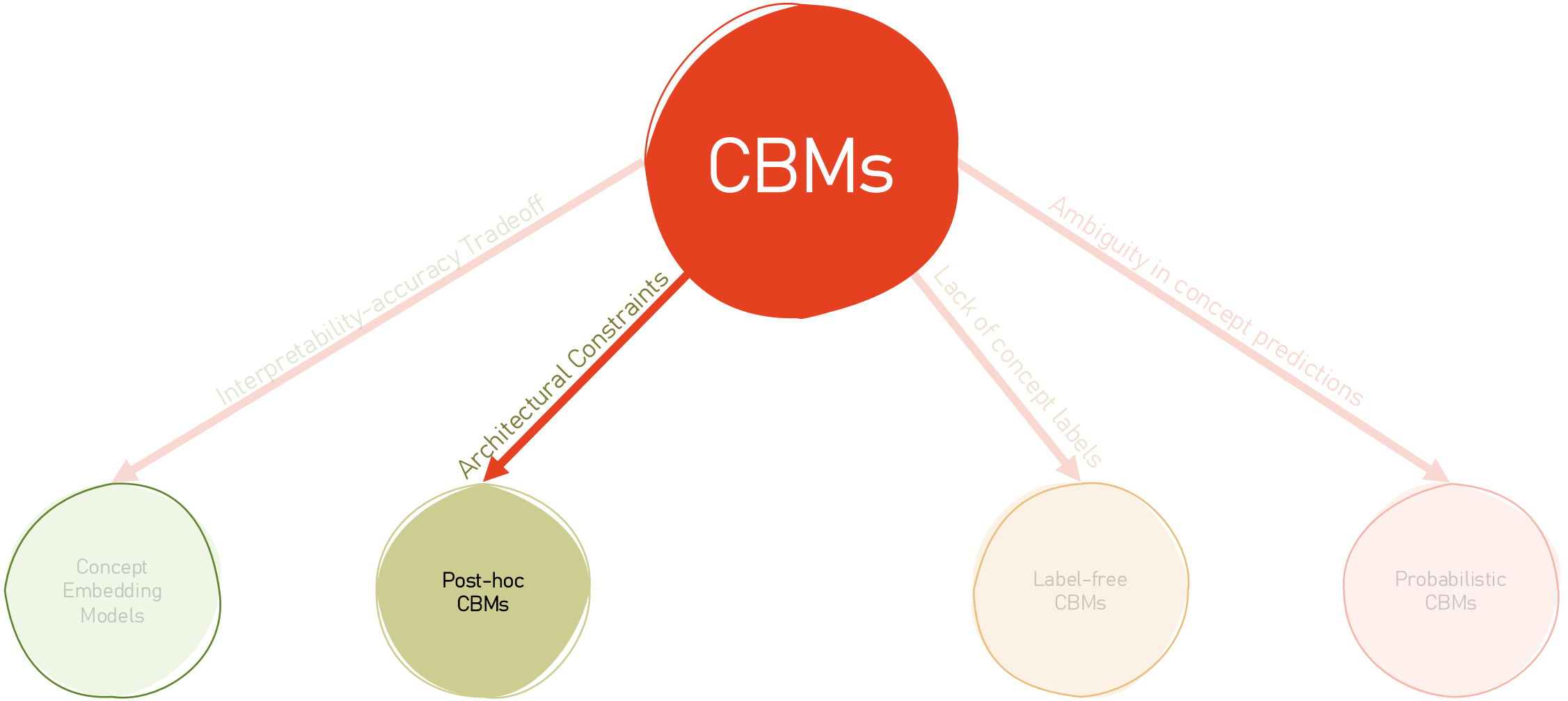
Learn two high-dimensional embeddings for each concept representing the concept when it is "on" and when it is "off"



Mix the two embeddings based on their predicted probability: $\hat{\boldsymbol{c}}_i = \hat{p}_i \boldsymbol{c}_i^{(+)} + (1 - \hat{p}_i)\boldsymbol{c}_i^{(-)}$

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

SPEED-DATING WITH CBM'S FRIENDS

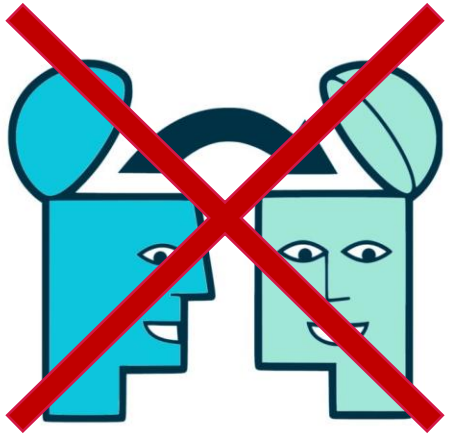


POST-HOC CBMS

YUKSEKGONUL ET AL. (ICLR 2023)

Limitation Being Addressed

Training a CBMs requires training from scratch, leading to significant constraints and architectural changes, and it requires all training samples to be concept annotated!



[1] Yuksekgonul, Mert, Maggie Wang, and James Zou. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

YUKSEKGONUL ET AL. (ICLR 2023)

Proposed Solution

Can we align a layer in a pre-trained model to concept scores obtained using T-CAV?

This would allow us to just finetune a single part of a pre-trained model using concept annotations in potentially distinct datasets!

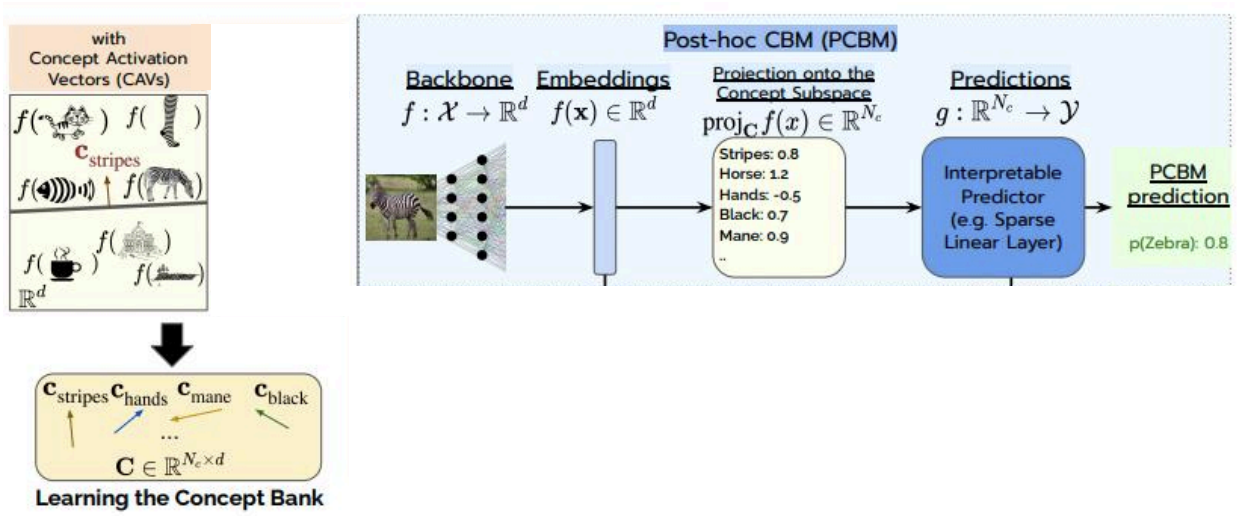
[1] Yuksekgonul, Mert, Maggie Wang, and James Zou. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

YUKSEKGONUL ET AL. (ICLR 2023)

Proposed Solution

Align a layer in a pre-trained model to concept scores obtained using T-CAV



Make the final prediction with an interpretable predictor

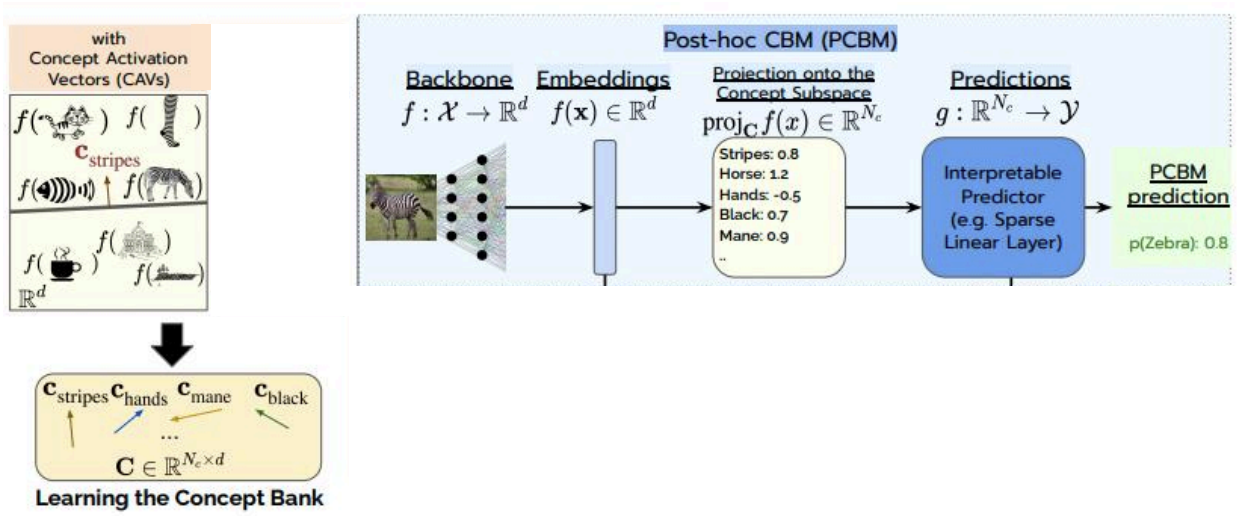
[1] Yuksekgonul, Mert, Maggie Wang, and James Zou. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

YUKSEKGONUL ET AL. (ICLR 2023)

Proposed Solution

Align a layer in a pre-trained model to concept scores obtained using T-CAV



Make the final prediction with an interpretable predictor

Question: do you see any issues with this architecture? (hint: think of our previous discussion)

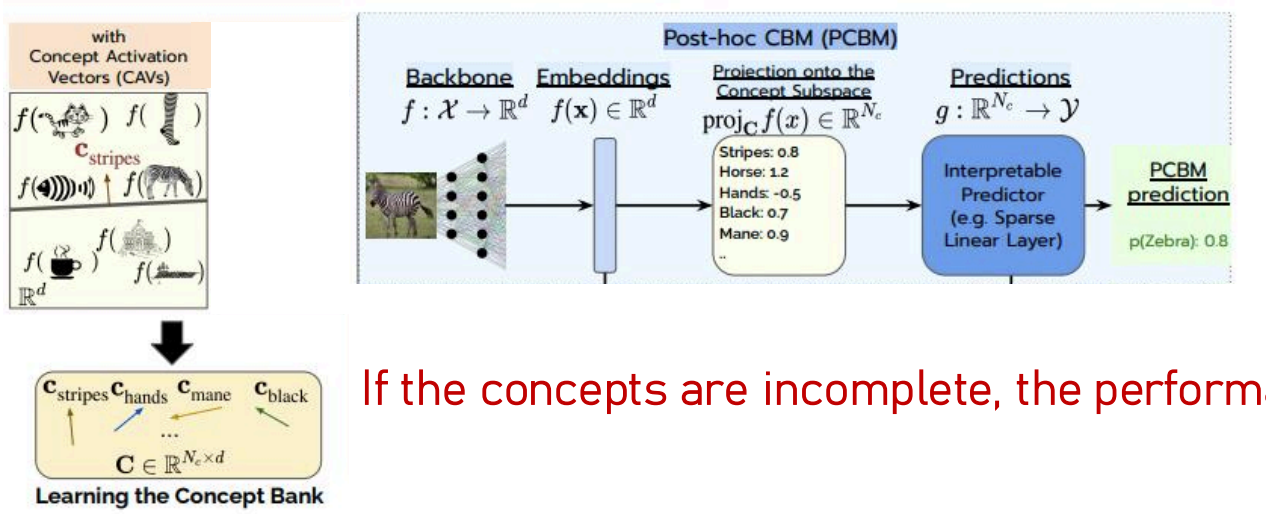
[1] Yuksekgonul, Mert, Maggie Wang, and James Zou. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

YUKSEKGONUL ET AL. (ICLR 2023)

Proposed Solution

Align a layer in a pre-trained model to concept scores obtained using T-CAV



Make the final prediction with an interpretable predictor

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

Question: do you see any issues with this architecture? (hint: think of our previous discussion)

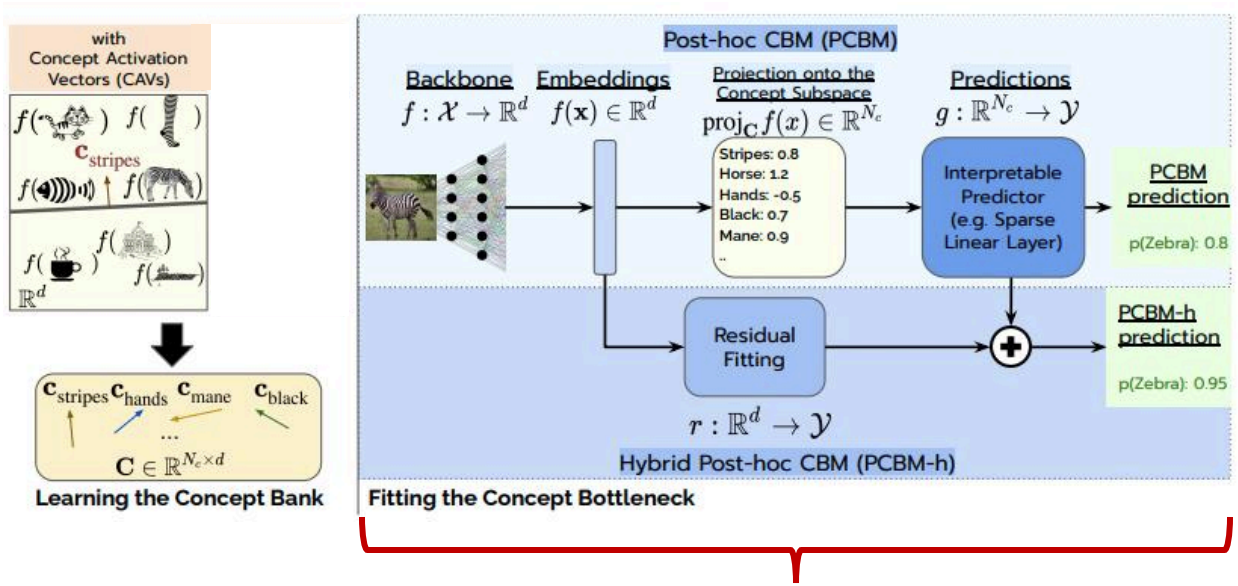
[1] Yuksekgonul, Mert, Maggie Wang, and James Zou. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

YUKSEKGONUL ET AL. (ICLR 2023)

Proposed Solution

Align a layer in a pre-trained model to concept scores obtained using T-CAV



A residual model can be fitted if the concept bank is incomplete!

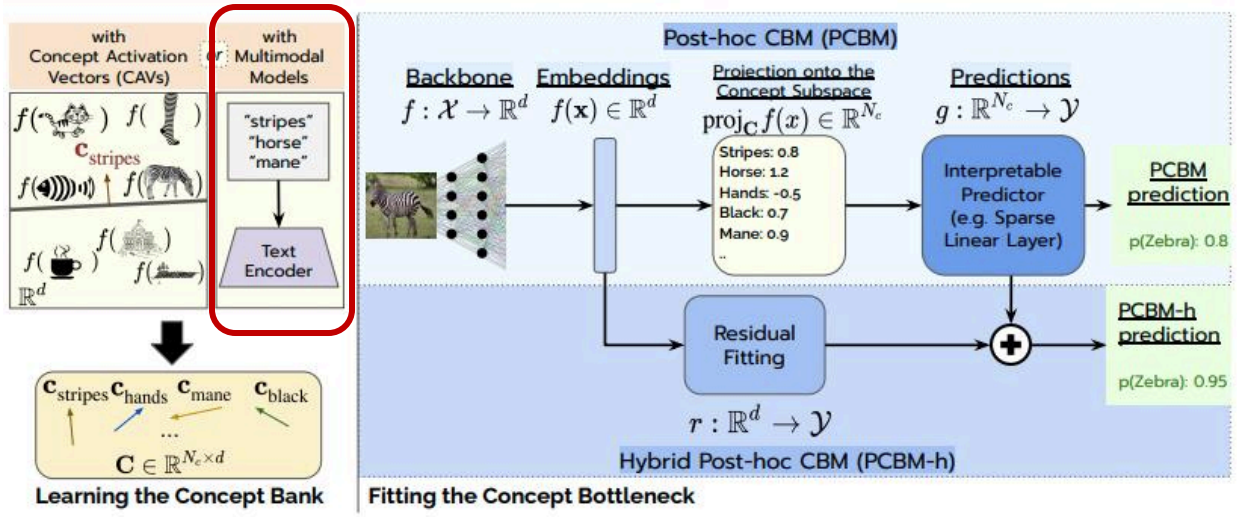
[1] Yuksekgonul, Mert, Maggie Wang, and James Zou. "Post-hoc concept bottleneck models." ICLR (2023).

POST-HOC CBMS

YUKSEKGONUL ET AL. (ICLR 2023)

Proposed Solution

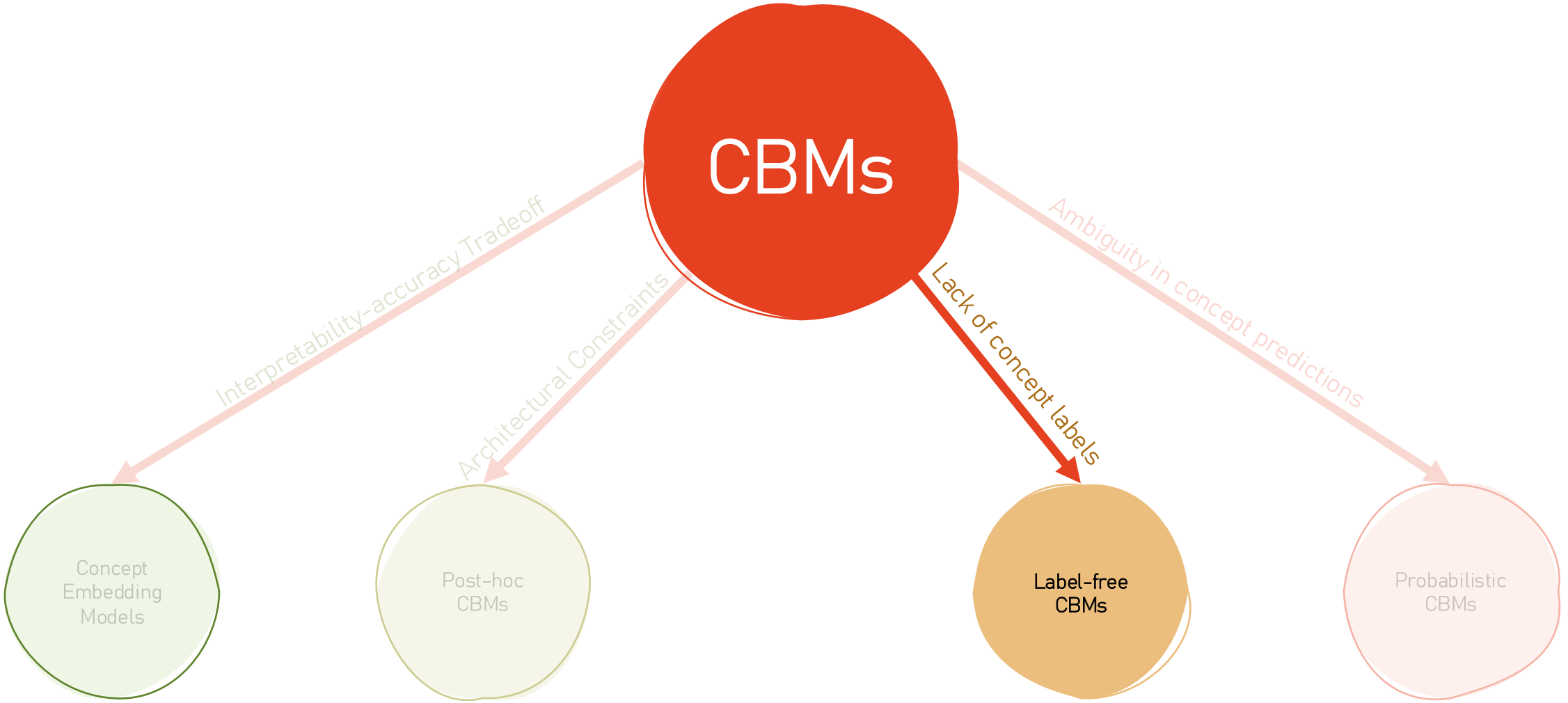
Align a layer in a pre-trained model to concept scores obtained using T-CAV



CAVs can be learnt using language-based concepts together with a multimodal model to learn CAVs!

[1] Yuksekgonul, Mert, Maggie Wang, and James Zou. "Post-hoc concept bottleneck models." ICLR (2023).

SPEED-DATING WITH CBM'S FRIENDS

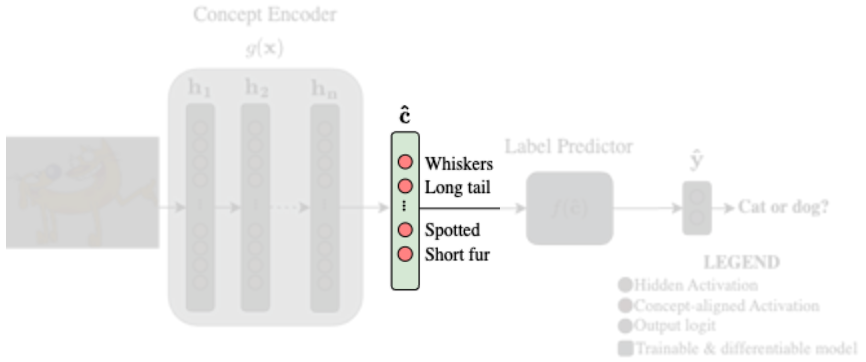


LABEL-FREE CBMS

OIKARINEN ET AL. (ICLR 2023)

Limitation Being Addressed

CBMs and CEMs 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!

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

LABEL-FREE CBMS

OIKARINEN ET AL. (ICLR 2023)

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}:"

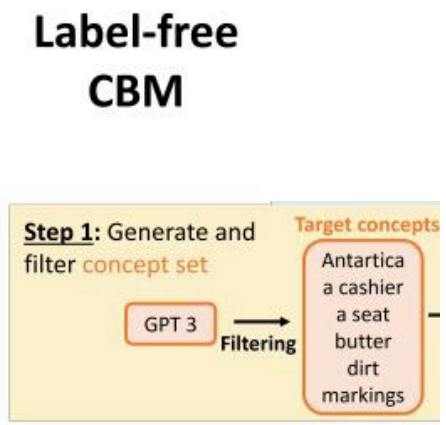
[1] Oikarinen, Tuomas, et al. "Label-Free Concept Bottleneck Models." ICLR (2023).

LABEL-FREE CBMS

OIKARINEN ET AL. (ICLR 2023)

Proposed Solution

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



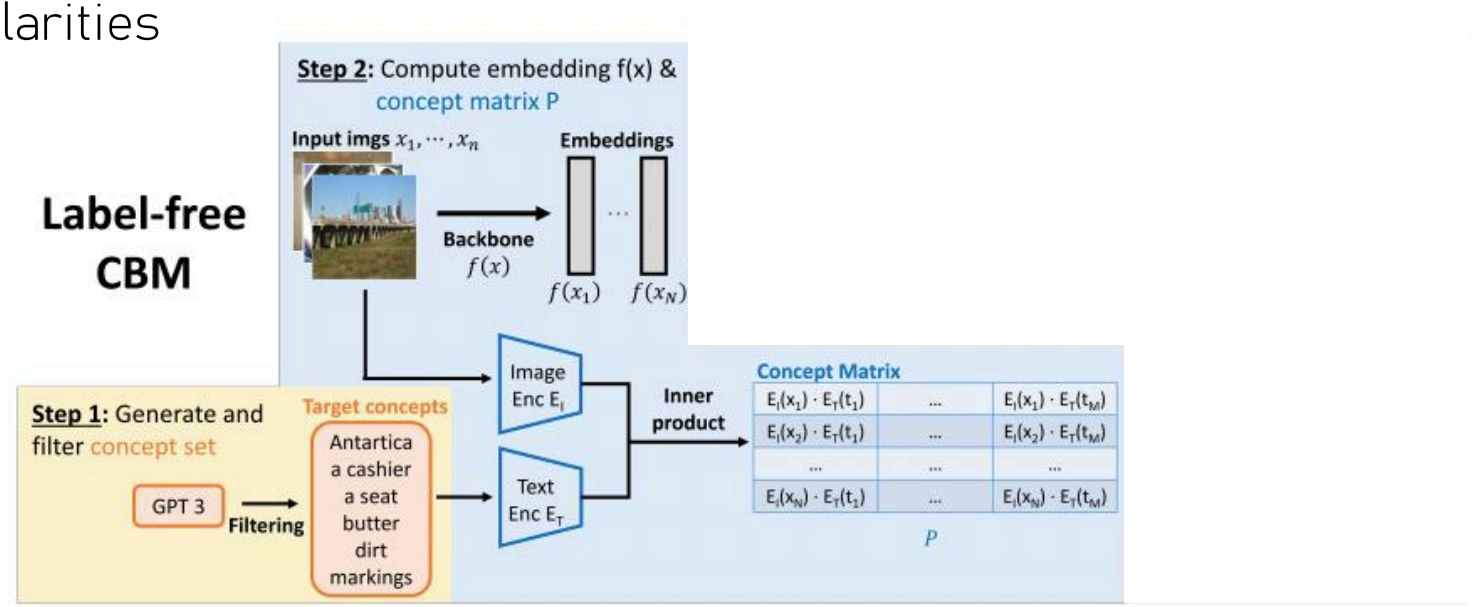
[1] Oikarinen, Tuomas, et al. "Label-Free Concept Bottleneck Models." ICLR (2023).

LABEL-FREE CBMS

OIKARINEN ET AL. (ICLR 2023)

Proposed Solution

Step 2: Map samples to an embedding space using a VLM (e.g., CLIP) and compute concept similarities



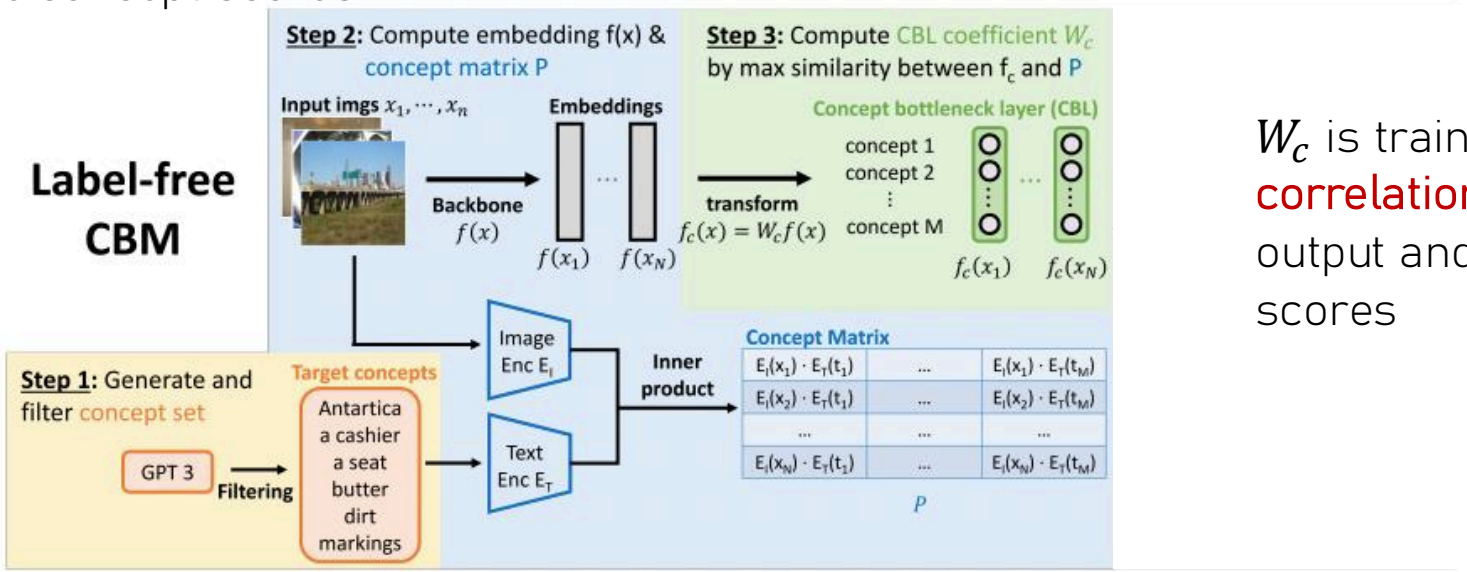
[1] Oikarinen, Tuomas, et al. "Label-Free Concept Bottleneck Models." ICLR (2023).

LABEL-FREE CBMS

OIKARINEN ET AL. (ICLR 2023)

Proposed Solution

Step 3: Learn a **linear mapping** between a backbone's embeddings and a **vector whose activations are aligned** with the concept scores



W_c is trained to **maximise the correlation** between its i-th output and the i-th concept's scores

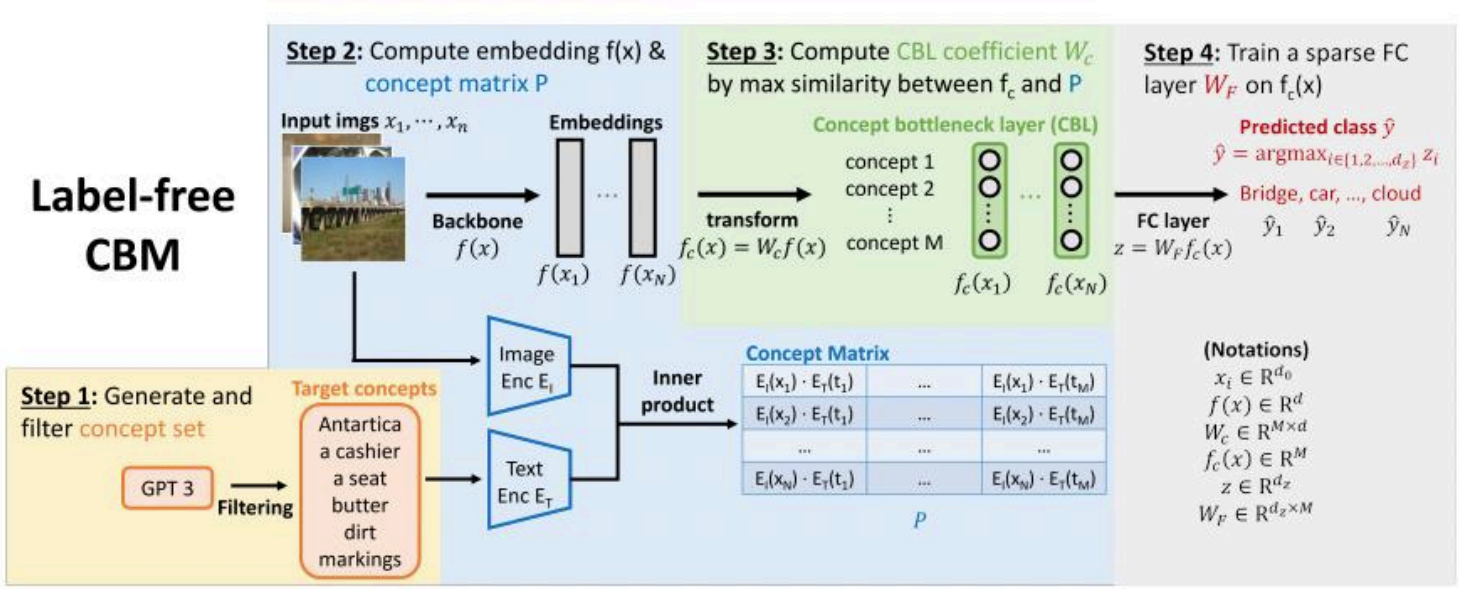
[1] Oikarinen, Tuomas, et al. "Label-Free Concept Bottleneck Models." ICLR (2023).

LABEL-FREE CBMS

OIKARINEN ET AL. (ICLR 2023)

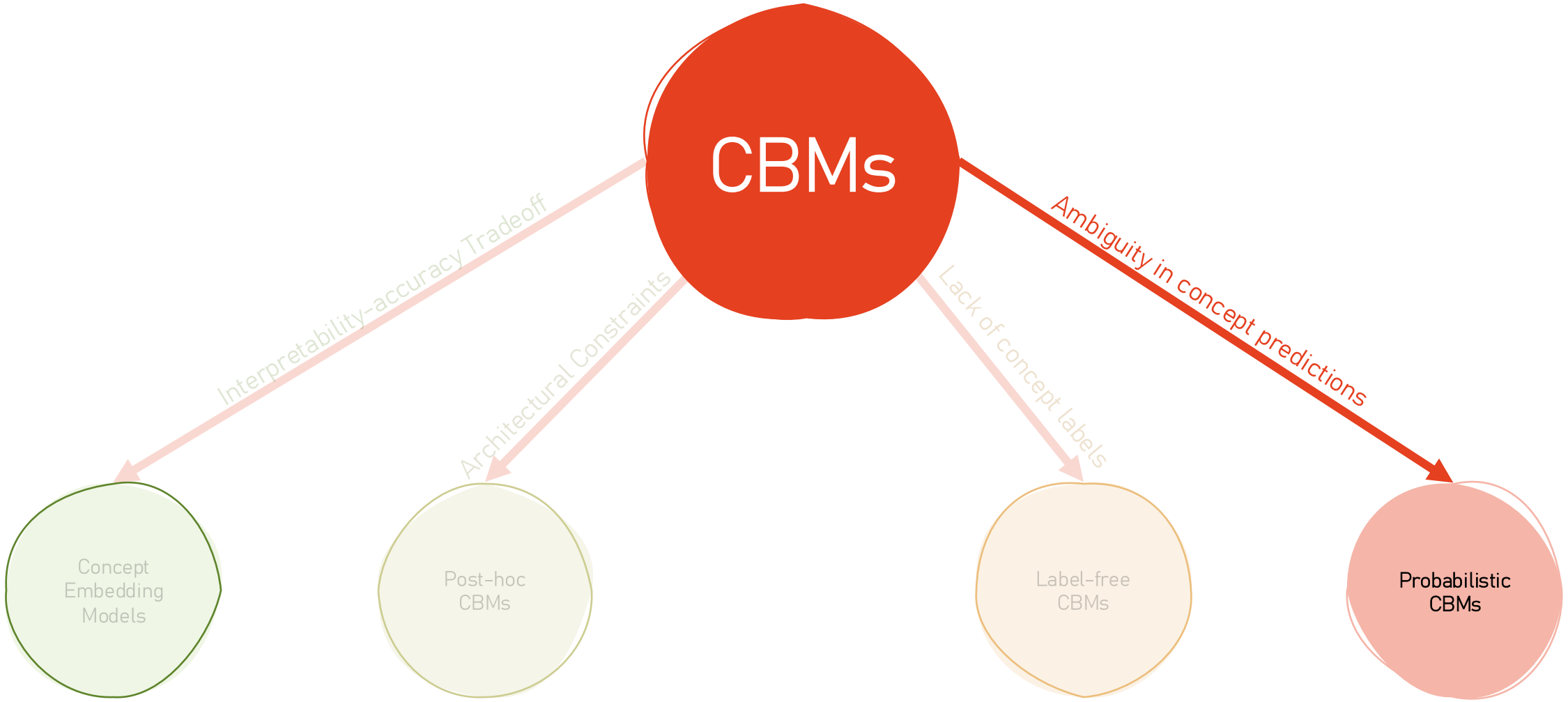
Proposed Solution

Step 4: Train a spare (interpretable) model to map predicted concept scores to tasks



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

SPEED-DATING WITH CBM'S FRIENDS

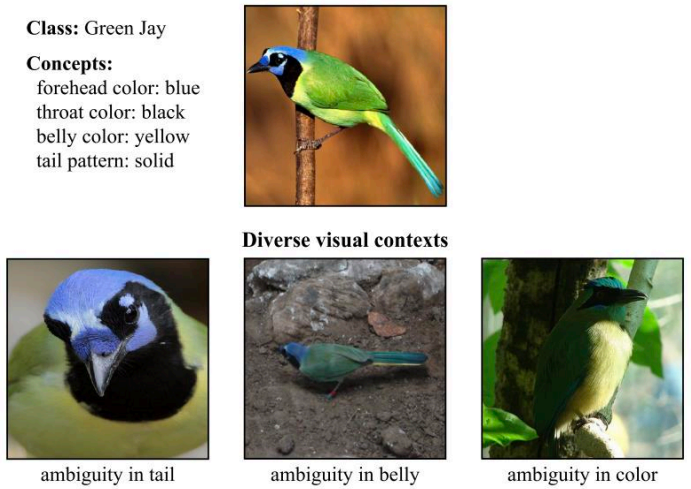


PROBABILISTIC CBMS

KIM ET AL. (ICML 2023)

Limitation Being Addressed

CBMs must predict concepts for all samples even they are ambiguous



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

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

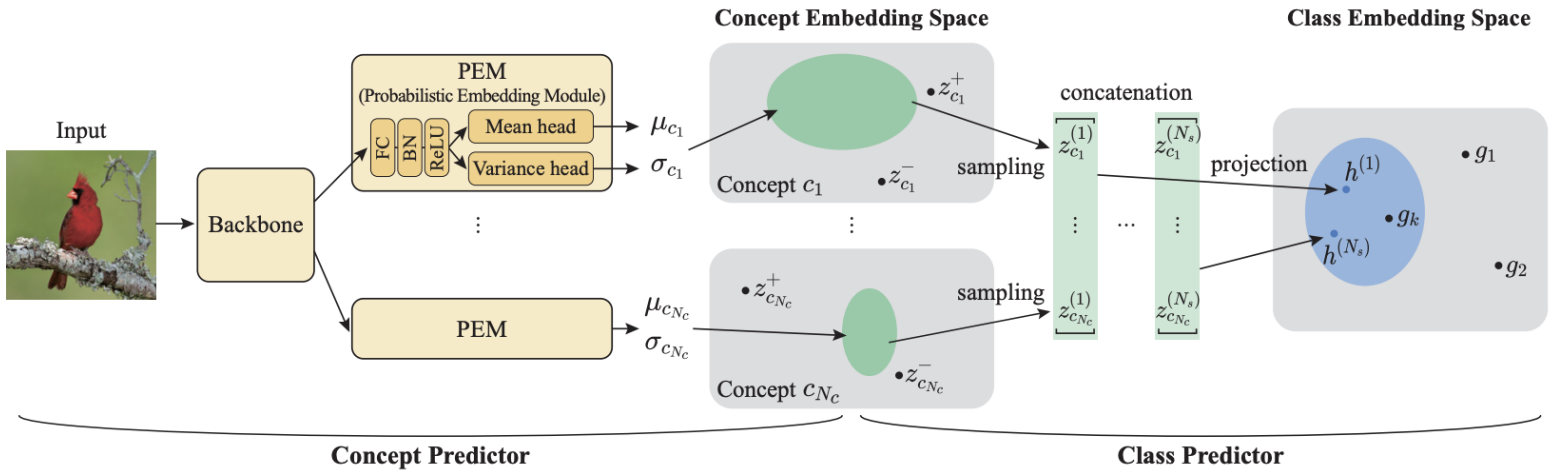
PROBABILISTIC CBMS

KIM ET AL. (ICML 2023)

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



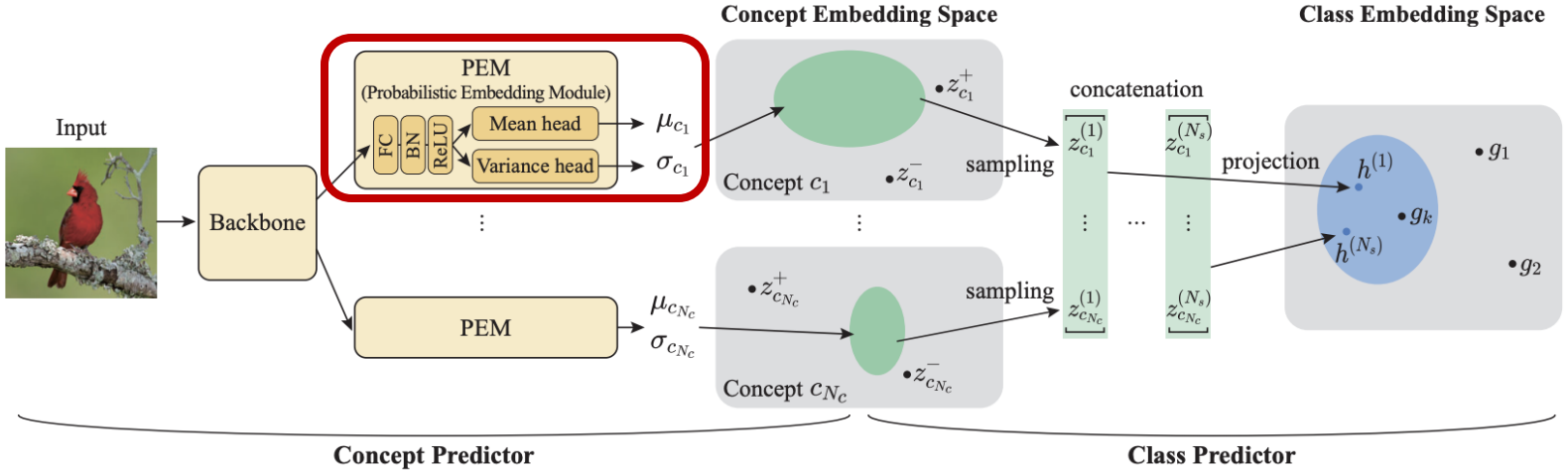
[1] Kim, Eunji, et al. "Probabilistic Concept Bottleneck Models." ICML (2023).

PROBABILISTIC CBMS

KIM ET AL. (ICML 2023)

Proposed Solution

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



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

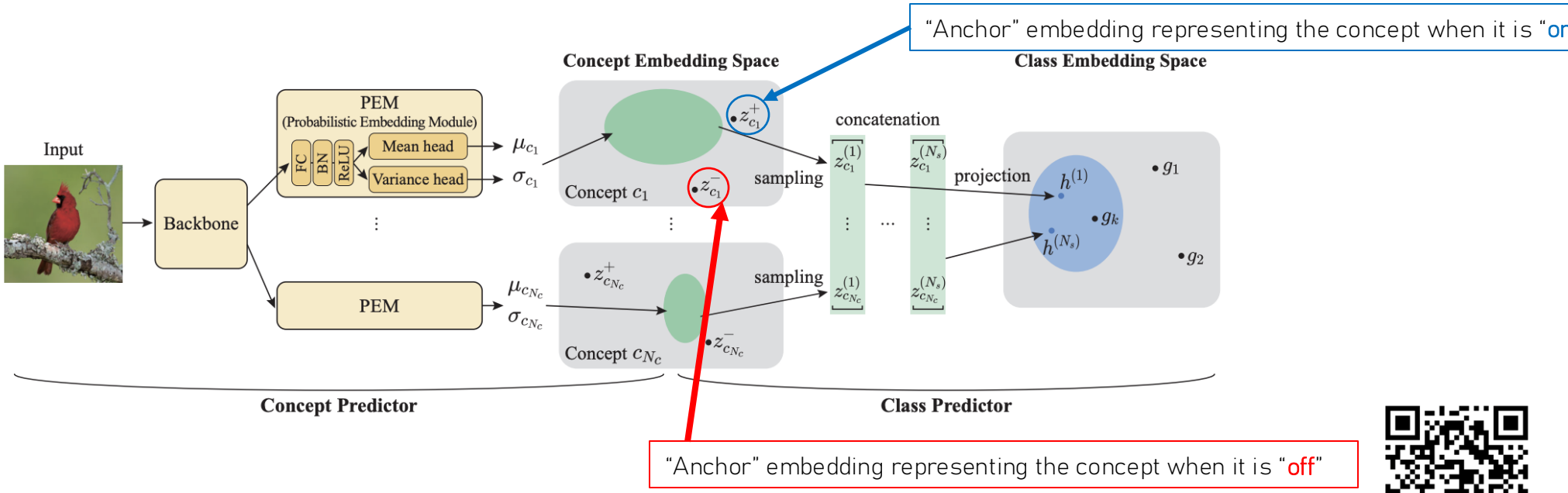
[1] Kim, Eunji, et al. "Probabilistic Concept Bottleneck Models." ICML (2023).

PROBABILISTIC CBMS

KIM ET AL. (ICML 2023)

Proposed Solution

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



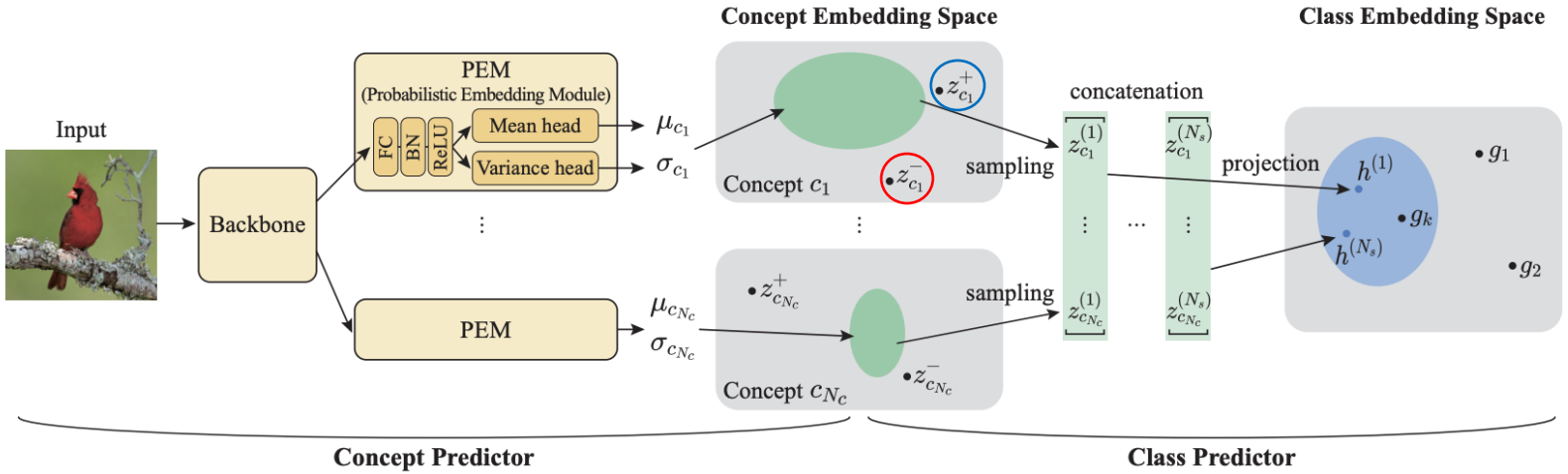
[1] Kim, Eunji, et al. "Probabilistic Concept Bottleneck Models." ICML (2023).

PROBABILISTIC CBMS

KIM ET AL. (ICML 2023)

Proposed Solution

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



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

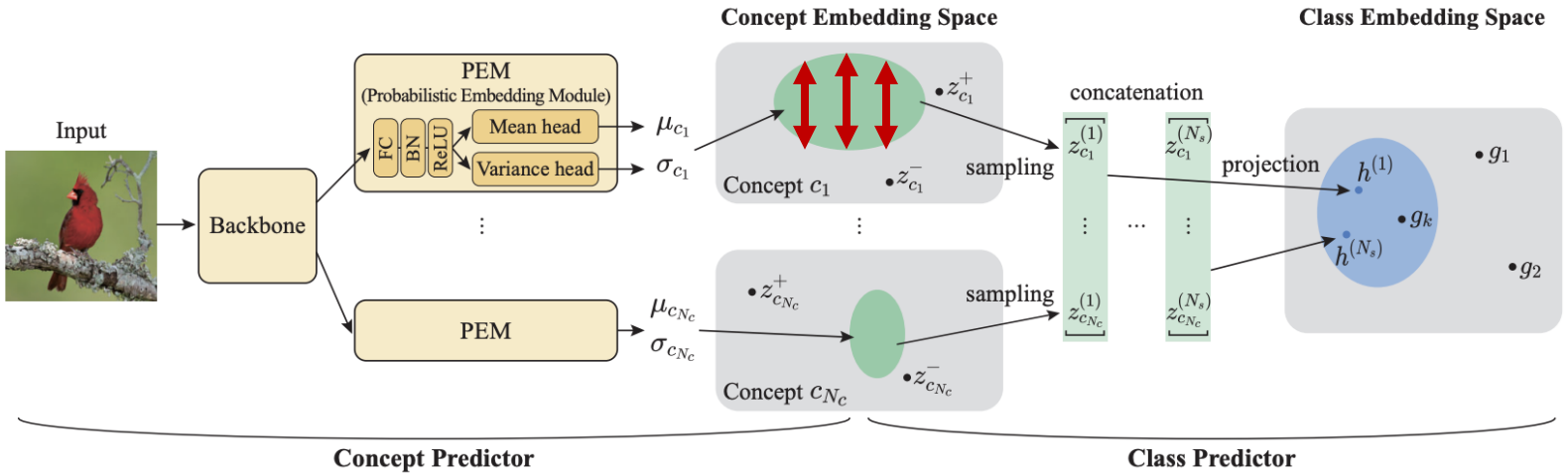
[1] Kim, Eunji, et al. "Probabilistic Concept Bottleneck Models." ICML (2023).

PROBABILISTIC CBMS

KIM ET AL. (ICML 2023)

Proposed Solution

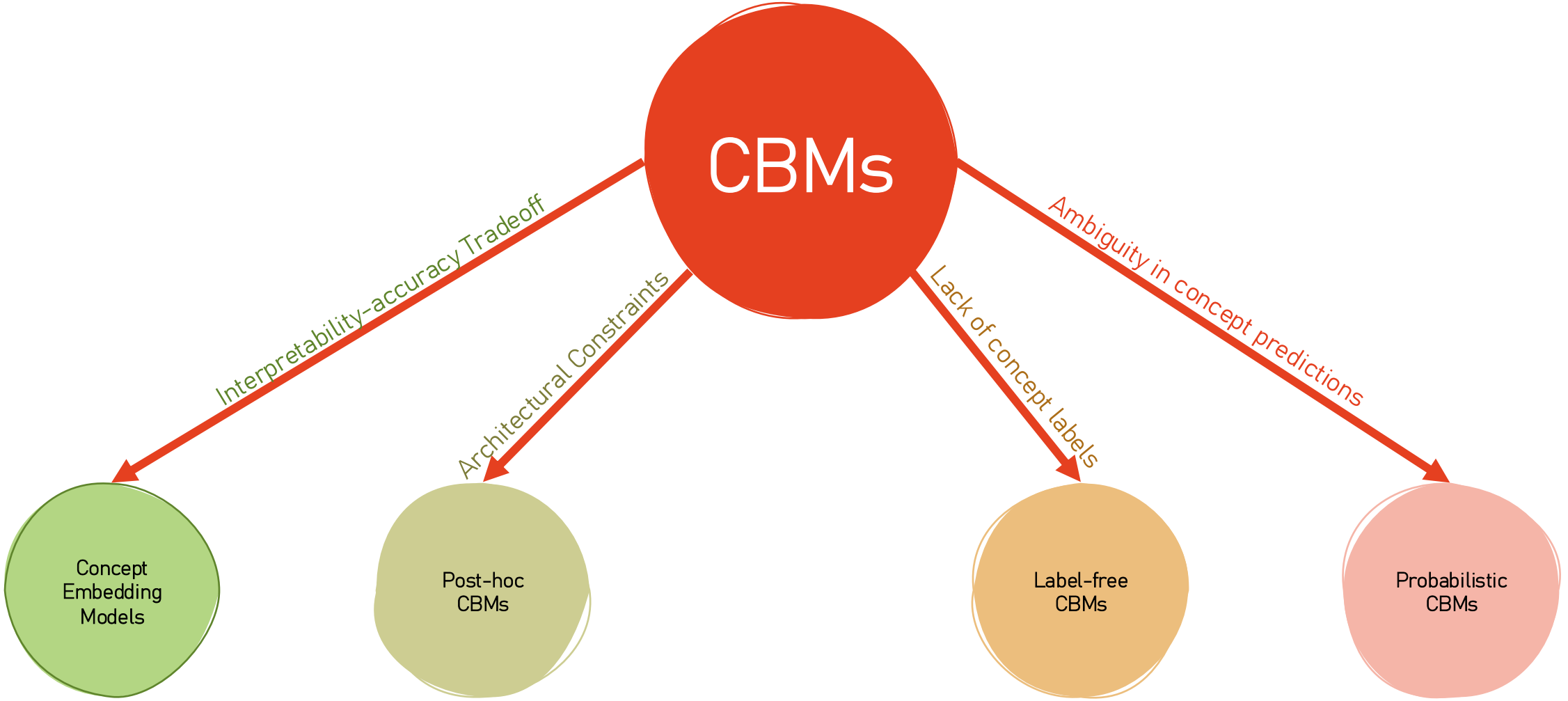
A concept's distribution's volume can be used to quantify its uncertainty!



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

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

END OF OUR SPEED DATING!



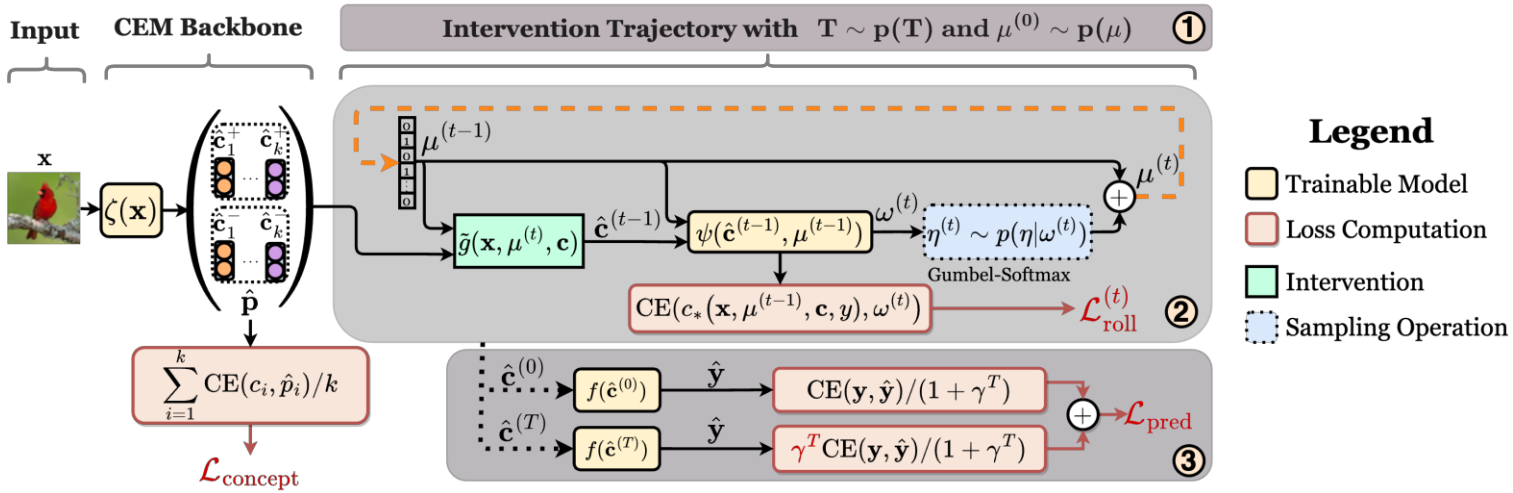
NEW DIRECTIONS

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

NEW DIRECTIONS

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

1. Improving the effect of concept interventions (a topic of personal importance)



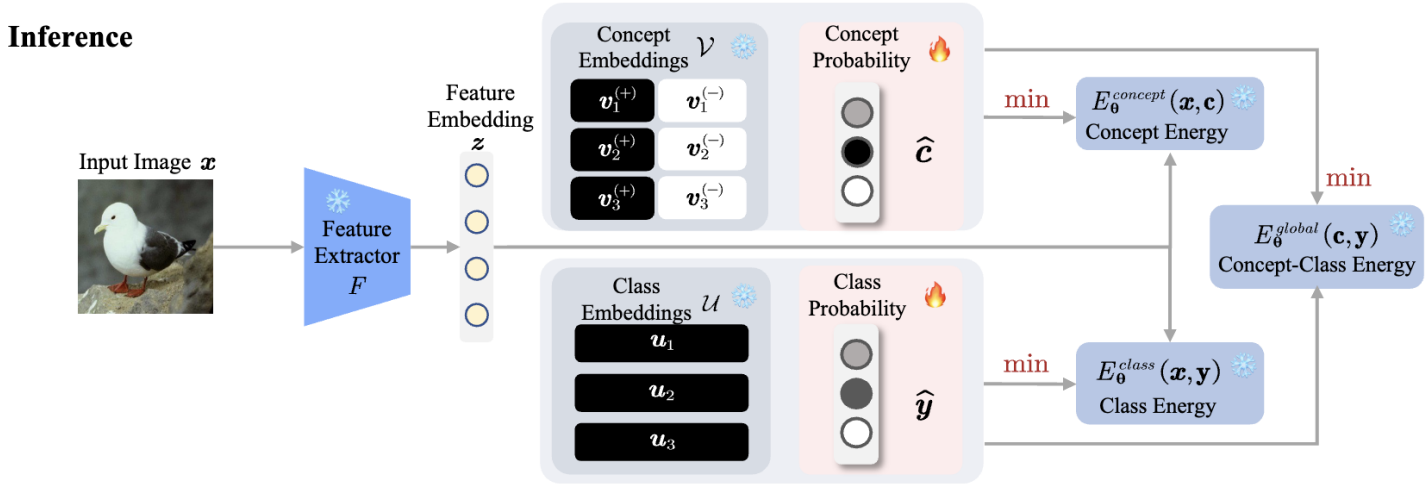
Intervention-aware Concept Embedding Models (Espinosa Zarlenga et al., 2023)

[1] Espinosa Zarlenga et al. "Learning to Receive Help: Intervention-aware Concept Embedding Models" NeurIPS (2023)

NEW DIRECTIONS

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

1. Generalising Concept Interventions (a topic of personal importance)
2. Understanding how to better model concept-to-concept relationships



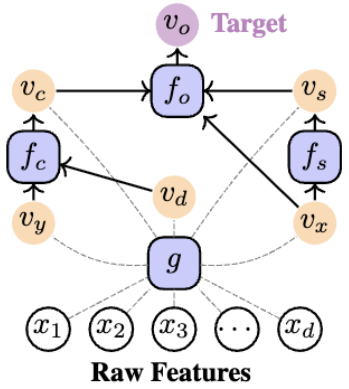
Energy-based Concept Bottleneck Models (Xu et al., 2024)

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

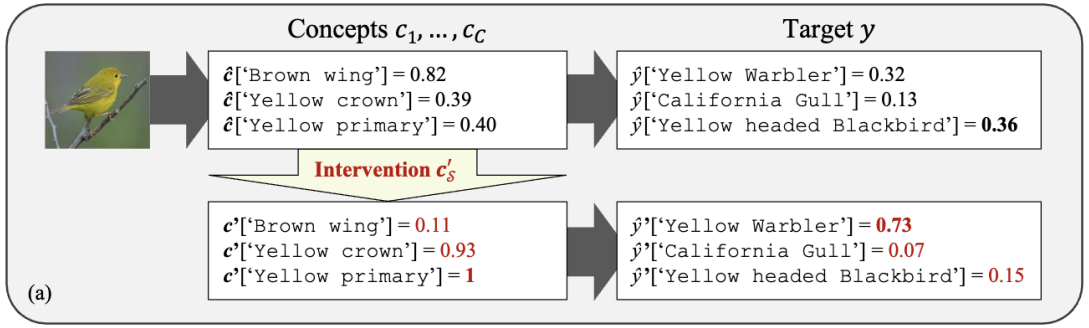
NEW DIRECTIONS

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

1. Generalising Concept Interventions (a topic of personal importance)
2. Understanding how to better model concept-to-concept relationships
3. Exploring the relationship between concepts, tasks, and causality



Causal Concept Graph Models (Dominici et al.)



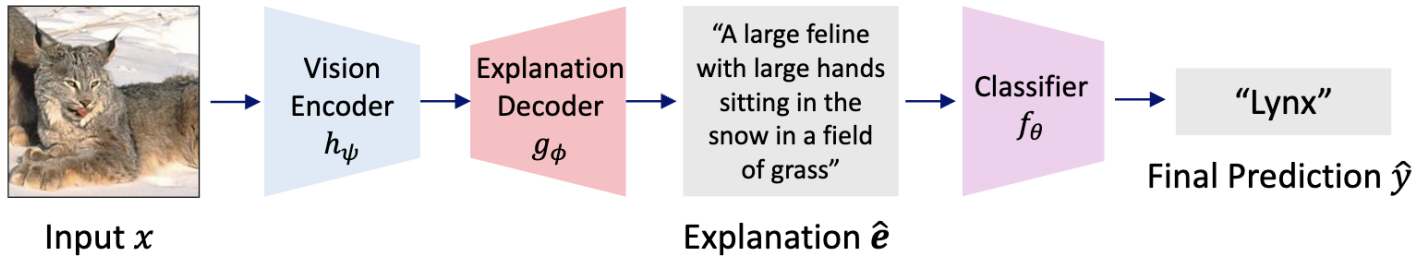
Stochastic CBMs (Vandenhirtz et al.)

[1] Dominici and Barbiero et al. "Causal Concept Embedding Models: Beyond Causal Opacity in Deep Learning." ICLR (2025)
[2] Vandenhirtz and Laguna et al. "Stochastic concept bottleneck models." NeurIPS (2024)..

NEW DIRECTIONS

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

1. Generalising Concept Interventions (a topic of personal importance)
2. Understanding how to better model concept-to-concept relationships
3. Exploring the relationship between concepts, tasks, and causality
4. Producing entirely language-based bottlenecks (very recent!)



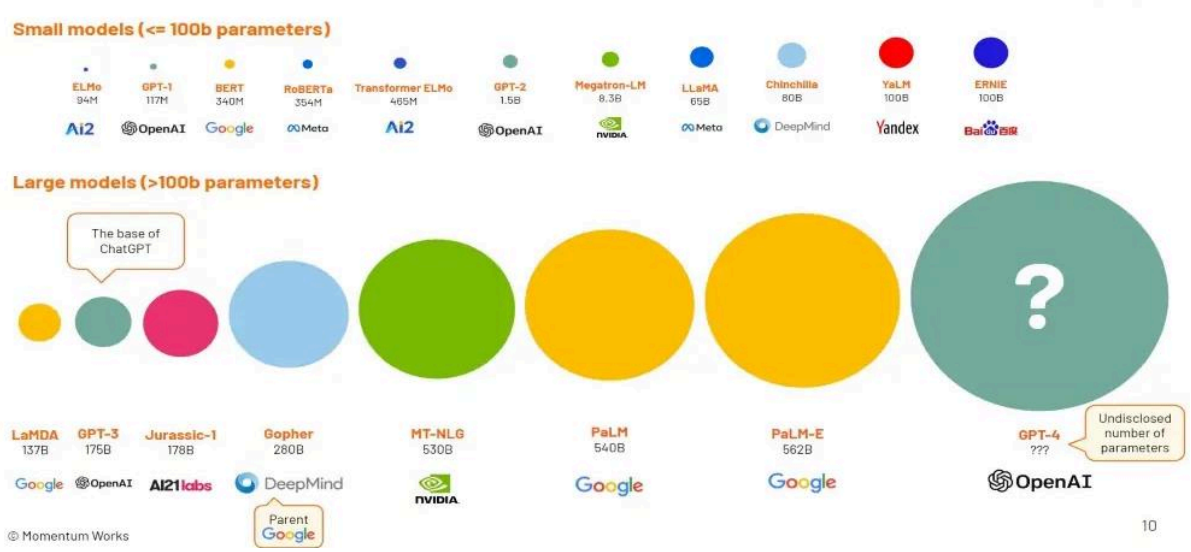
Explanation Bottleneck Models (Yamaguchi et al.)

[1] Yamaguchi, Shin'ya, and Kosuke Nishida. "Explanation Bottleneck Models." AAAI (2025).

ARE CBMS THE SOLUTION?

Maybe… but they still have some clear open questions:

1. CBM-based models require serious architectural changes or unrealistic data annotations which may not scale to very large models.

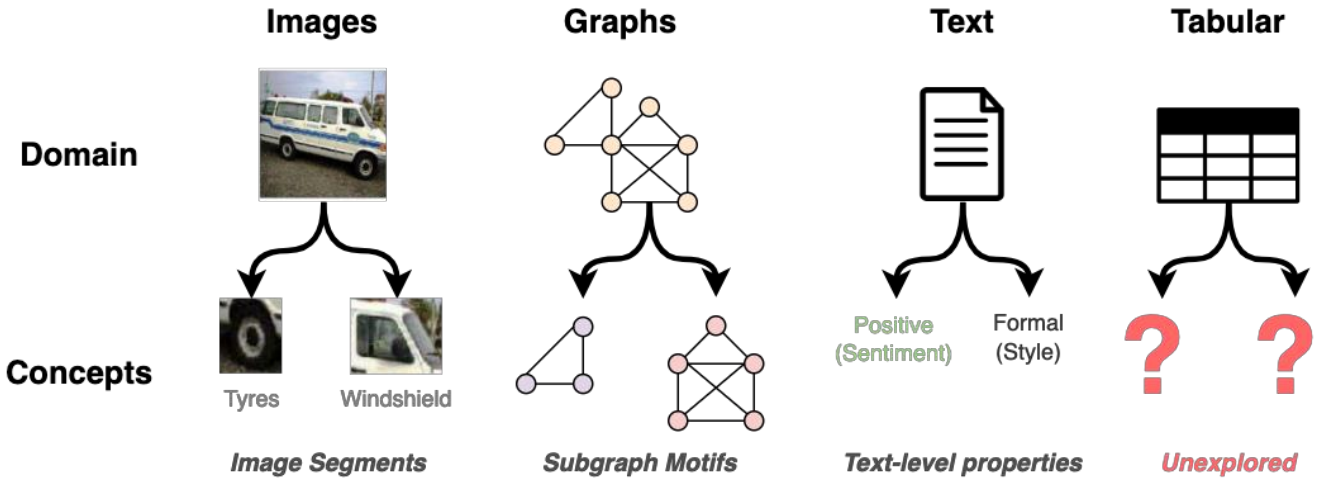


[1] Image taken from Momentum Works at "https://thelowdown.momentum.asia/the-emergence-of-large-language-models-llms/"

ARE CBMS THE SOLUTION?

Maybe… but they still have some clear open questions:

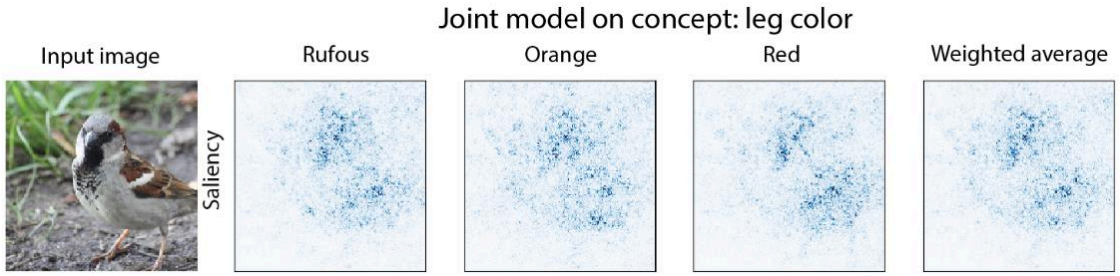
1. CBM-based models require serious architectural changes or unrealistic data annotations which may not scale to very large models.
2. Concepts are ill-defined in a lot of domains (e.g., tabular data, speech, etc)



ARE CBMS THE SOLUTION?

Maybe… but they still have some clear open questions:

1. CBM-based models require serious architectural changes or unrealistic data annotations which may not scale to very large models.
2. Concepts are ill-defined in a lot of domains (e.g., tabular data, speech, etc)
3. Concepts are still predicted with black-box models, leading to accidental leakage

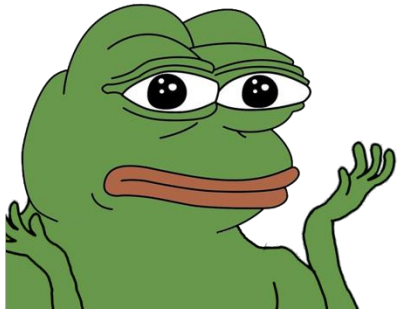


[1] Margeloiu et al. "Do concept bottleneck models learn as intended?." arXiv preprint arXiv:2105.04289 (2021).
[2] Mahinpei et al. "Promises and pitfalls of black-box concept learning models." arXiv preprint arXiv:2106.13314 (2021).

ARE CBMS THE SOLUTION?

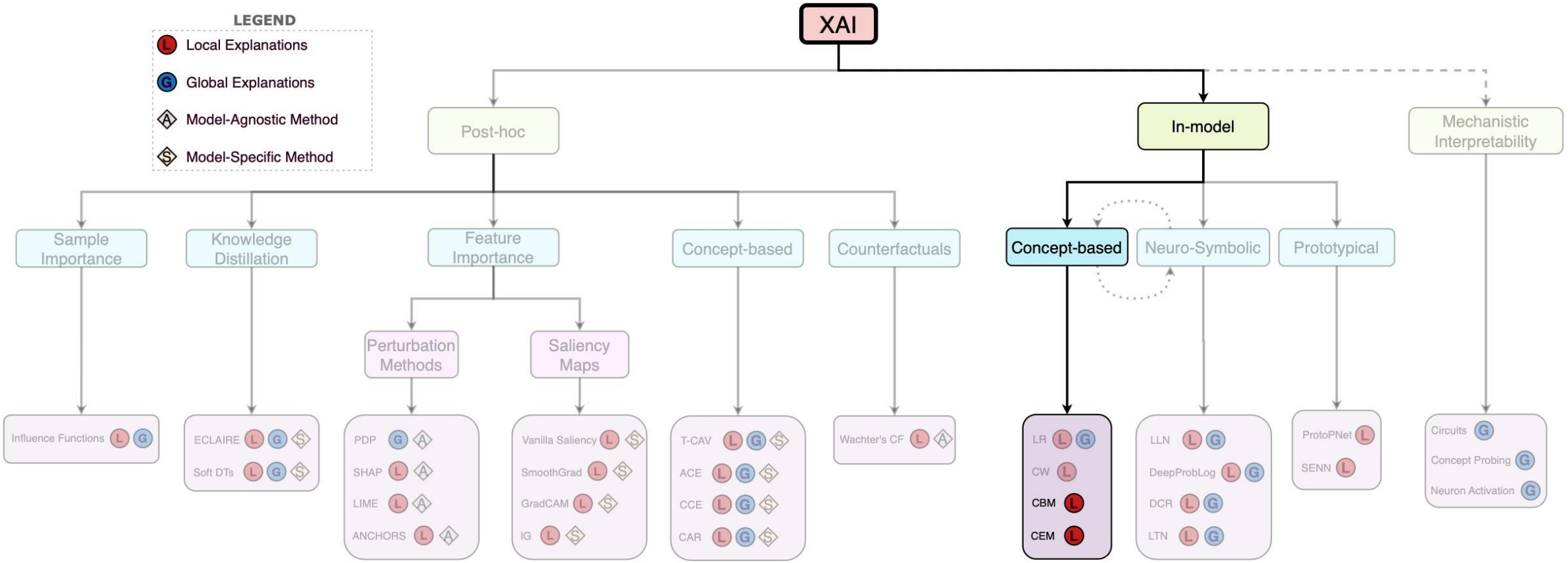
Maybe… but they still have some clear open questions:

What can we do about it?

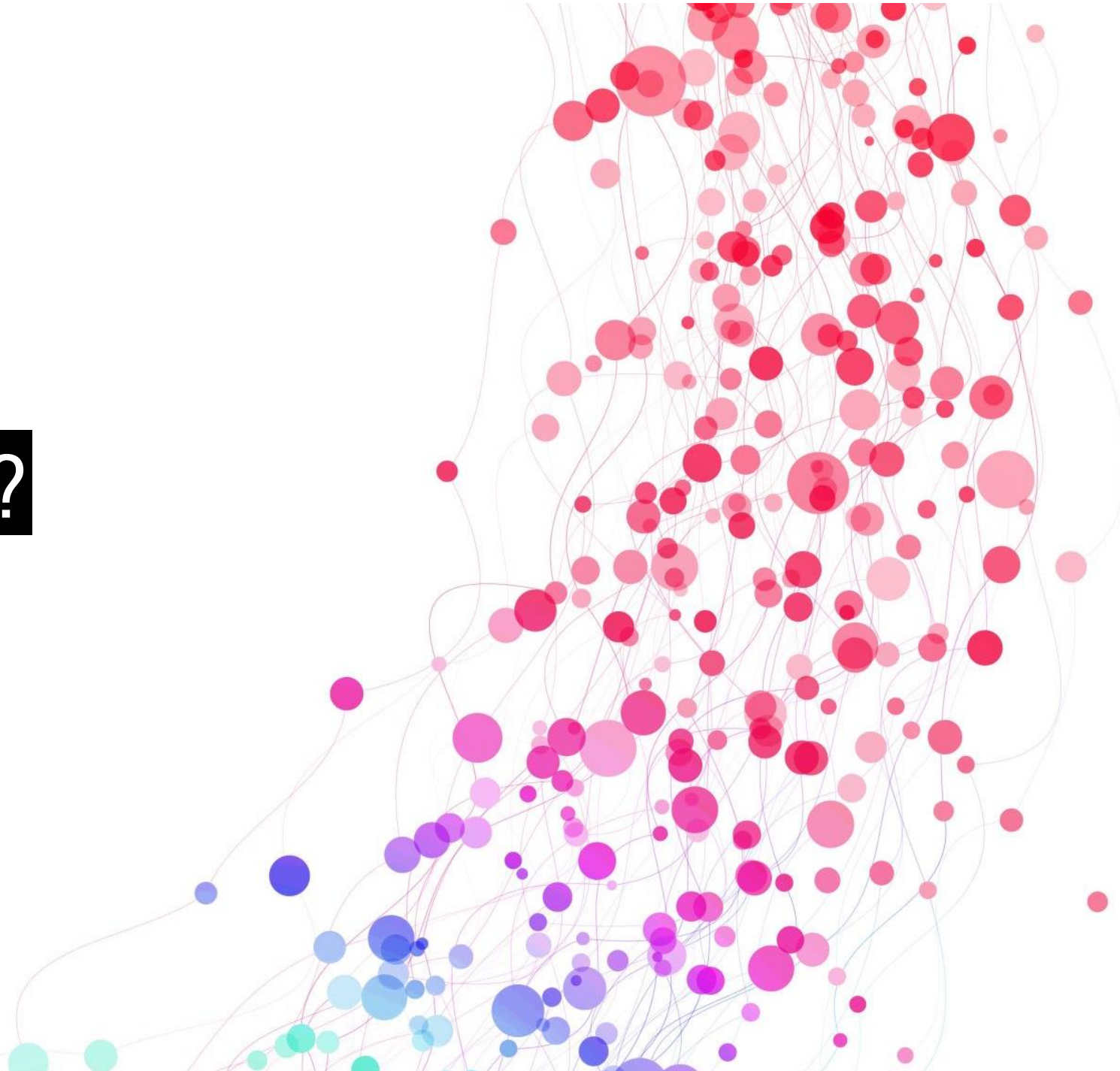


More on this in two weeks!

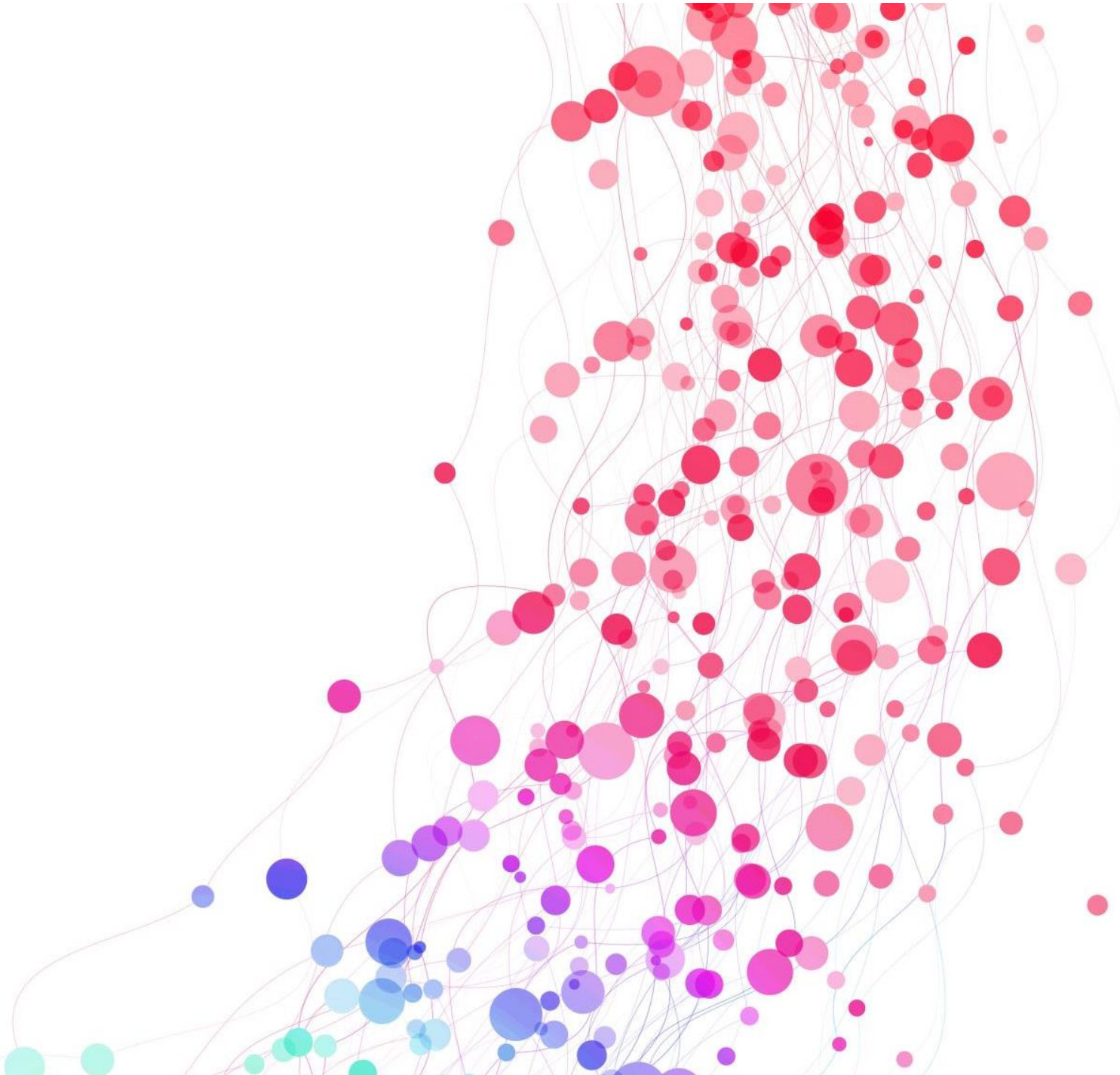
RECAP FOR TODAY



QUESTIONS?



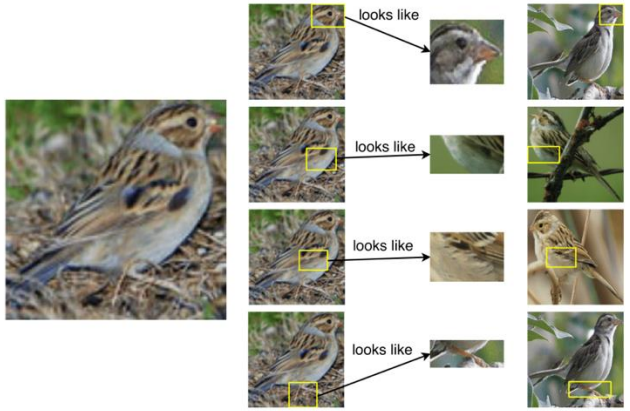
EXTRA MATERIAL



PROTOPNET

CHEN ET AL. (NEURIPS 2019)

Humans very often tend to explain themselves by pointing back at examples from their past experience: “This looks like that…”



ProtoPNet [1]:
“Wouldn’t it be nice if we could learn these prototypes in a differentiable manner?”

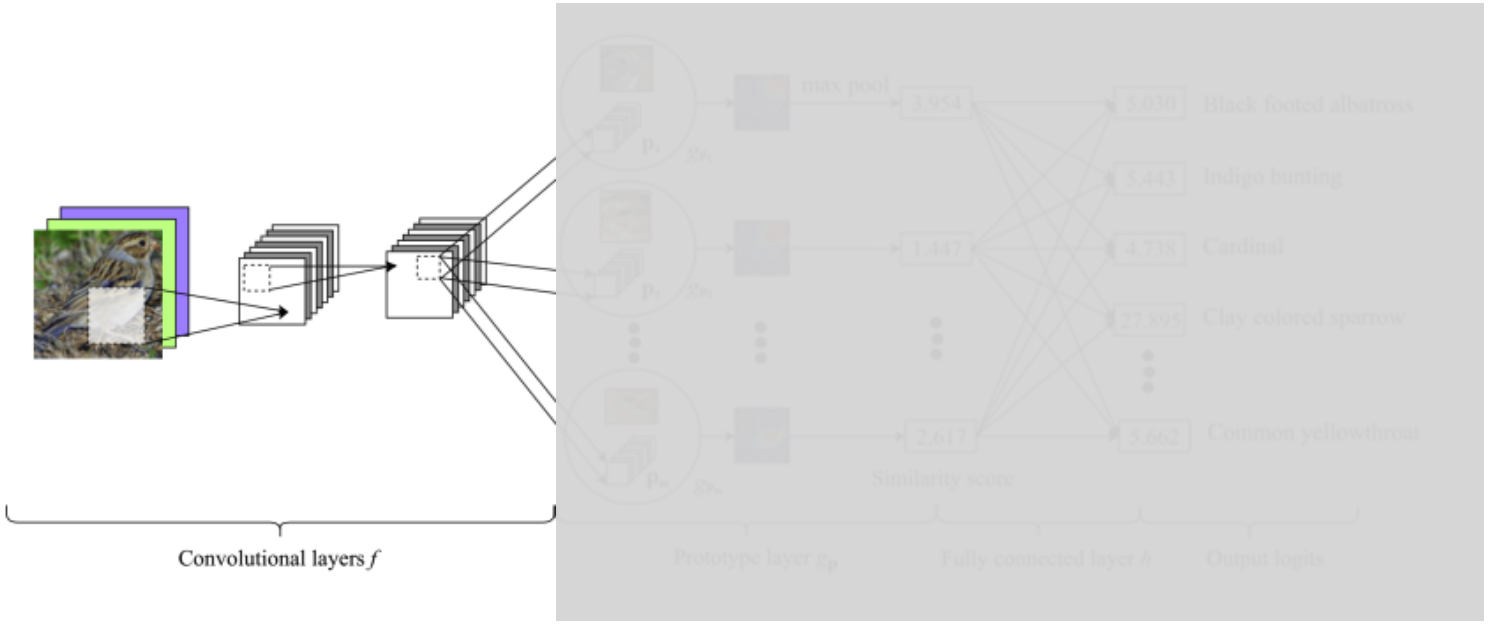
We call this prototypical explanations and we construct them by looking at how training samples can be used to explain new predictions

[1] Chen, Chaofan, et al. "This looks like that: deep learning for interpretable image recognition." *NeurIPS* (2019).

LEARNING PROTOTYPICAL EXPLANATIONS

ProtoPNet divides a neural architecture into four components:

1. We extract 3D representations of our input images through a pretrained model

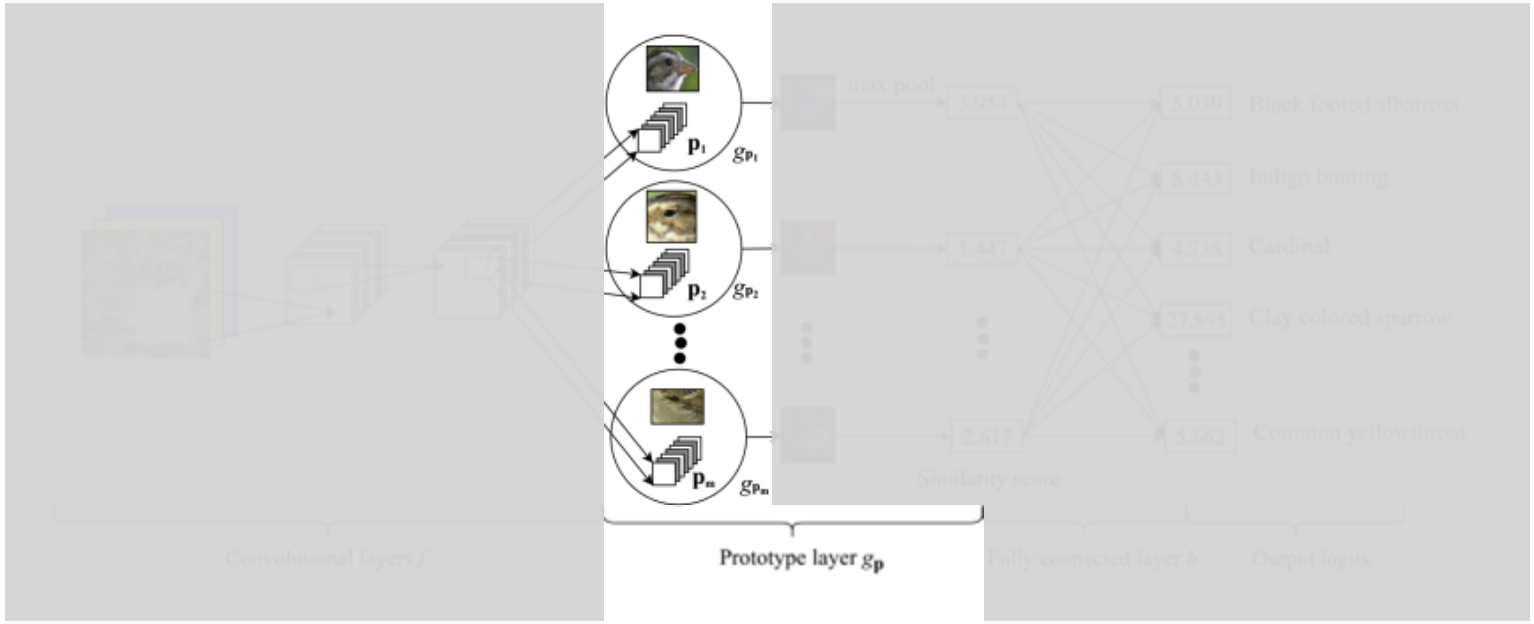


[1] Chen, Chaofan, et al. "This looks like that: deep learning for interpretable image recognition." *NeurIPS* (2019).

LEARNING PROTOTYPICAL EXPLANATIONS

ProtoPNet divides a neural architecture into four components:

2. Then, we will learn a prototype layer which holds m prototypical "patches".



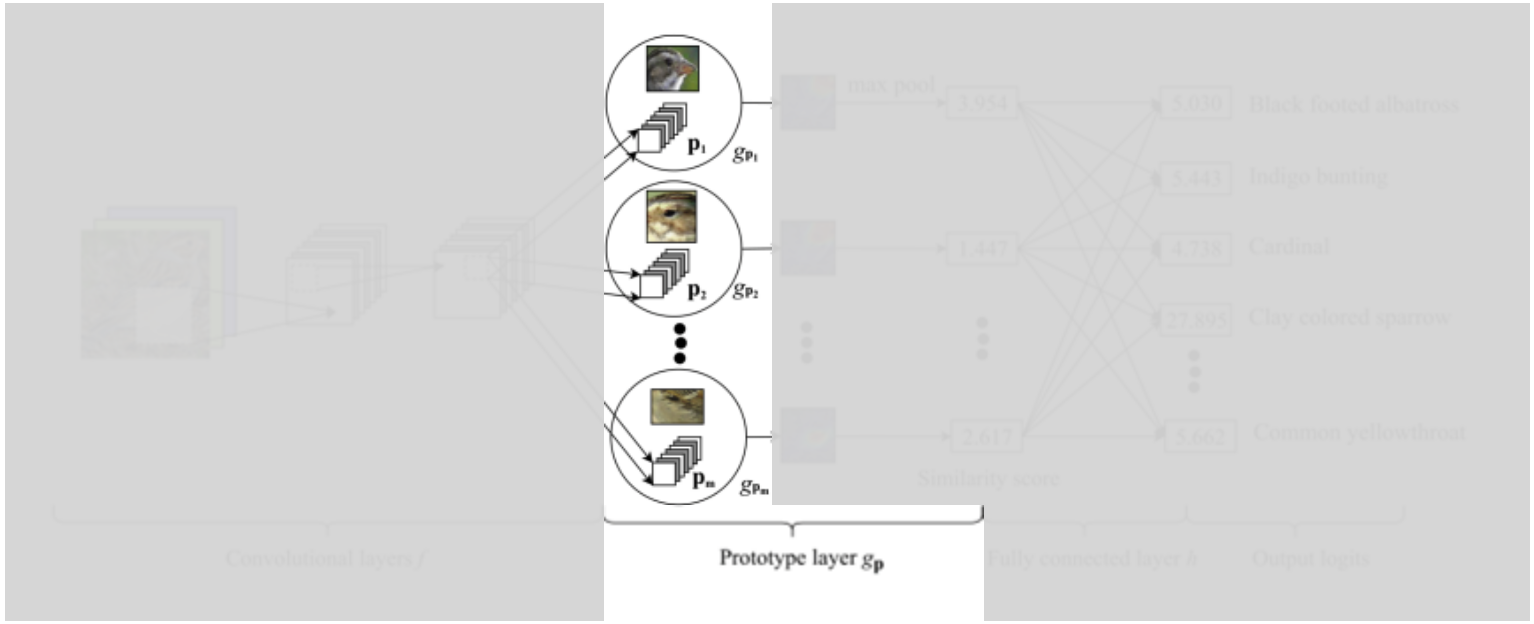
[1] Chen, Chaofan, et al. "This looks like that: deep learning for interpretable image recognition." *NeurIPS* (2019).

LEARNING PROTOTYPICAL EXPLANATIONS

ProtoPNet divides a neural architecture into four components:

2. Then, we will learn a prototype layer which holds m prototypical "patches".

We partition the image's latent code into small patches of the same size as our prototypes!

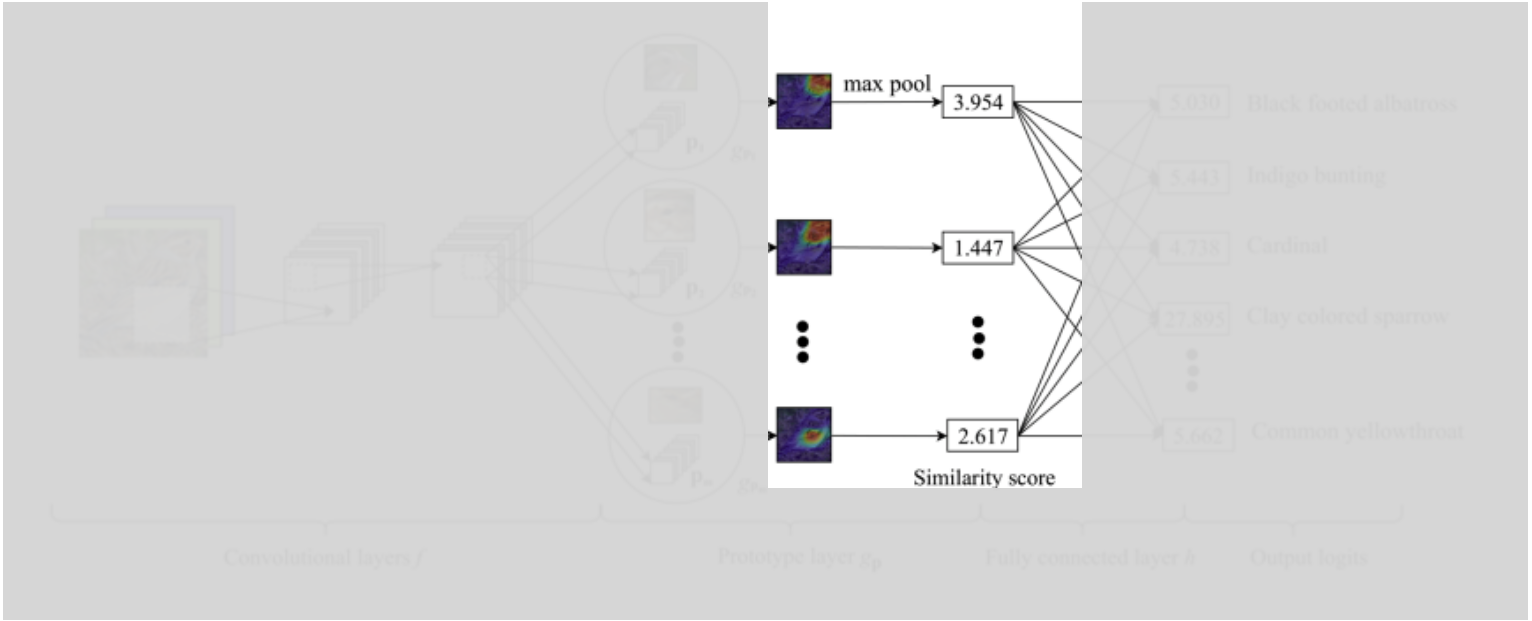


[1] Chen, Chaofan, et al. "This looks like that: deep learning for interpretable image recognition." *NeurIPS* (2019).

LEARNING PROTOTYPICAL EXPLANATIONS

ProtoPNet divides a neural architecture into four components:

3. We compute similarities between patches in the input image and our set of learnable prototypes to get similarities between image patches and prototypes:

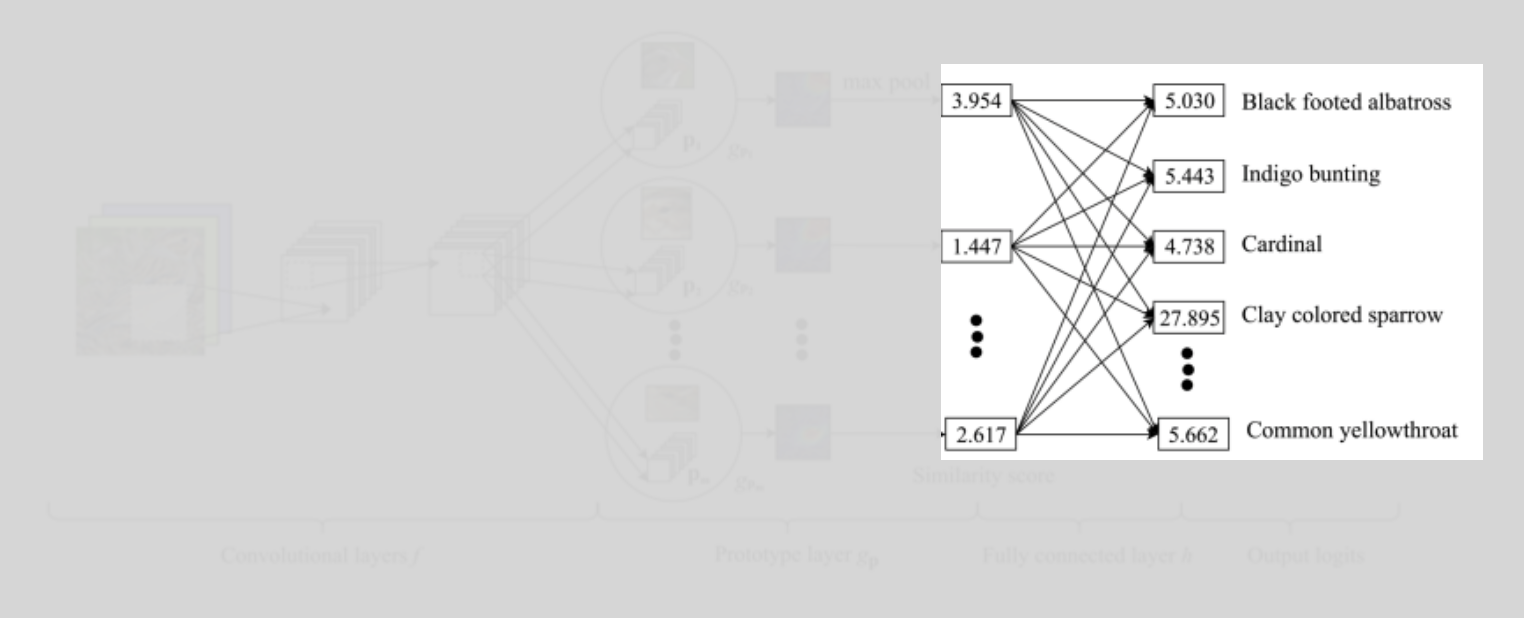


[1] Chen, Chaofan, et al. "This looks like that: deep learning for interpretable image recognition." *NeurIPS* (2019).

LEARNING PROTOTYPICAL EXPLANATIONS

ProtoPNet divides a neural architecture into four components:

4. Finally, we output a prediction from the set of similarity scores using an interpretable classifier



[1] Chen, Chaofan, et al. "This looks like that: deep learning for interpretable image recognition." *NeurIPS* (2019).

HOW TO TRAIN YOUR PROTOPNET?

We train this architecture by iterating over:

a. Training the prototype layer to learn prototypes that are
   i. Equally split across all classes (all classes are assigned k prototypes)

HOW TO TRAIN YOUR PROTOPNET?

We train this architecture by iterating over:

a. Training the prototype layer to learn prototypes that are
   i. Equally split across all classes (all classes are assigned k prototypes)
   ii. Clustered (at least one patch close to a prototype of your own class)

$$\text{Clst} = \frac{1}{n}\sum_{i=1}^{n} \min_{j:\mathbf{p}_j \in \mathbf{P}_{y_i}} \min_{\mathbf{z} \in \text{patches}(f(\mathbf{x}_i))} \|\mathbf{z} - \mathbf{p}_j\|_2^2$$

Minimise patch distance for at least one patch in the same class

HOW TO TRAIN YOUR PROTOPNET?

We train this architecture by iterating over:

a. Training the prototype layer to learn prototypes that are
   i. Equally split across all classes (all classes are assigned k prototypes)
   ii. Clustered (at least one patch close to a prototype of your own class)
   iii. Separated (patches are far from prototypes from other classes)

$$\text{Sep} = -\frac{1}{n}\sum_{i=1}^{n} \min_{j:\mathbf{p}_j \notin \mathbf{P}_{y_i}} \min_{\mathbf{z} \in \text{patches}(f(\mathbf{x}_i))} \|\mathbf{z} - \mathbf{p}_j\|_2^2$$

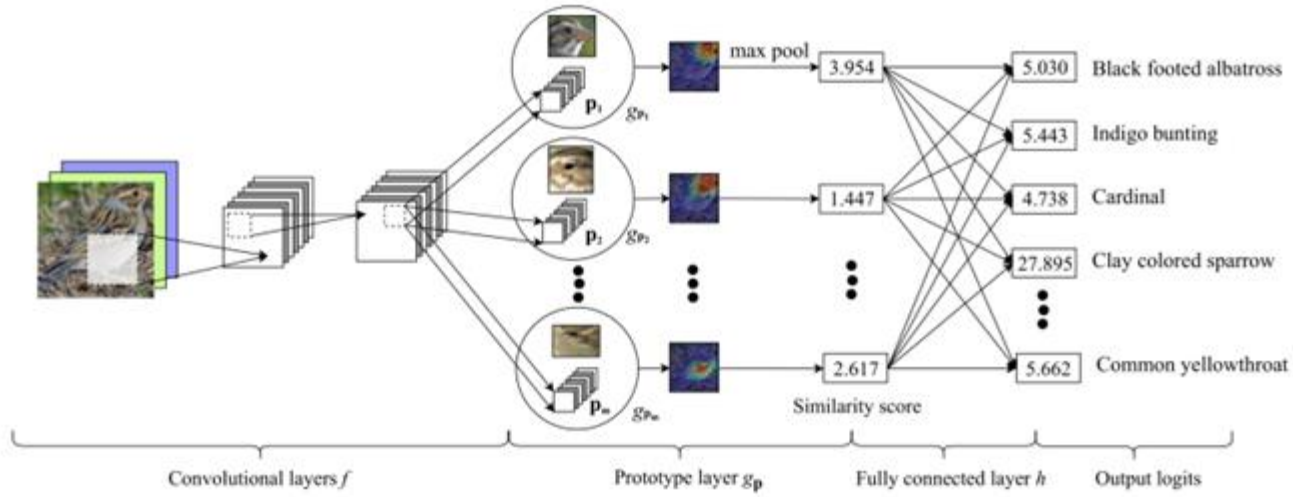
Maximise patch distance for all patches in other classes

HOW TO TRAIN YOUR PROTOPNET?

We train this architecture by iterating over:

a. Training the prototype layer.

b. Projecting learnt prototypes to their closest patch in the training set.



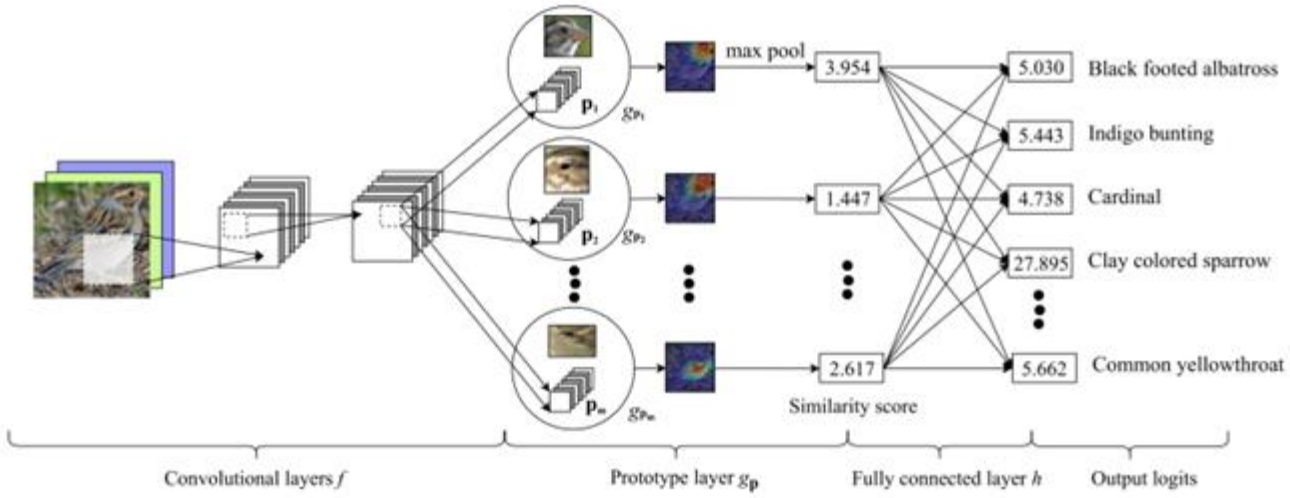
HOW TO TRAIN YOUR PROTOPNET?

We train this architecture by iterating over:

a. Training the prototype layer.

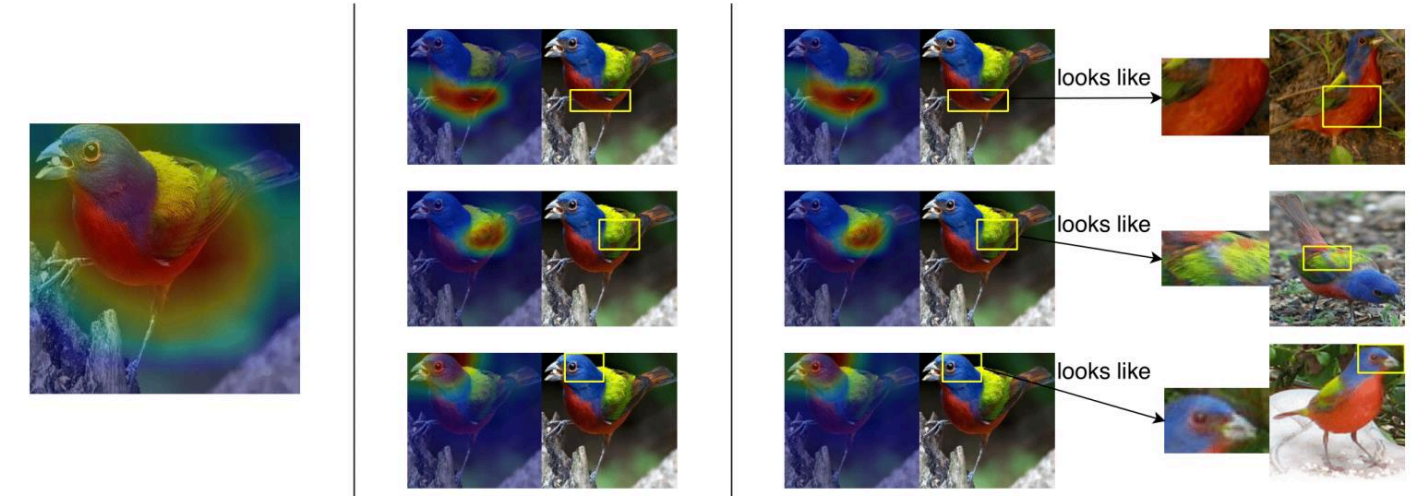
b. Projecting learnt prototypes to their closest patch in the training set.

c. Fine-tuning the output fully connected layer to map prototype scores to labels.



EXPLAINING BY EXAMPLE

ProtoPNet explains its predictions with prototypes that are highlighted in the input image with heatmaps → More interpretable than feature attribution!



(a) Object attention (class activation map)

(b) Part attention (attention-based models)

(c) Part attention + comparison with learned prototypical parts (our model)

ON PROTOTYPICAL EXPLANATIONS

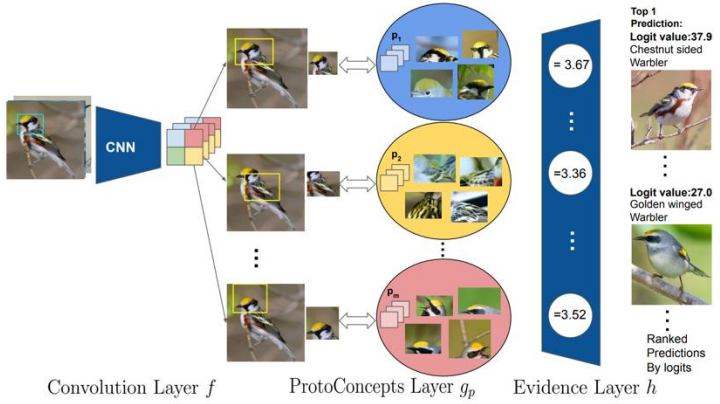
Prototypes are certainty an interesting research direction! However:

1. The are prone to complicated training pipelines as it is non-trivial to learn "traceable" prototypes in a differentiable manner

ON PROTOTYPICAL EXPLANATIONS

Prototypes are certainty an interesting research direction! However:

1. The are prone to complicated training pipelines as it is non-trivial to learn "traceable" prototypes in a differentiable manner
2. They tend to select only single patches/parts of examples as prototypes, complicating disambiguating the reason behind selecting that prototype



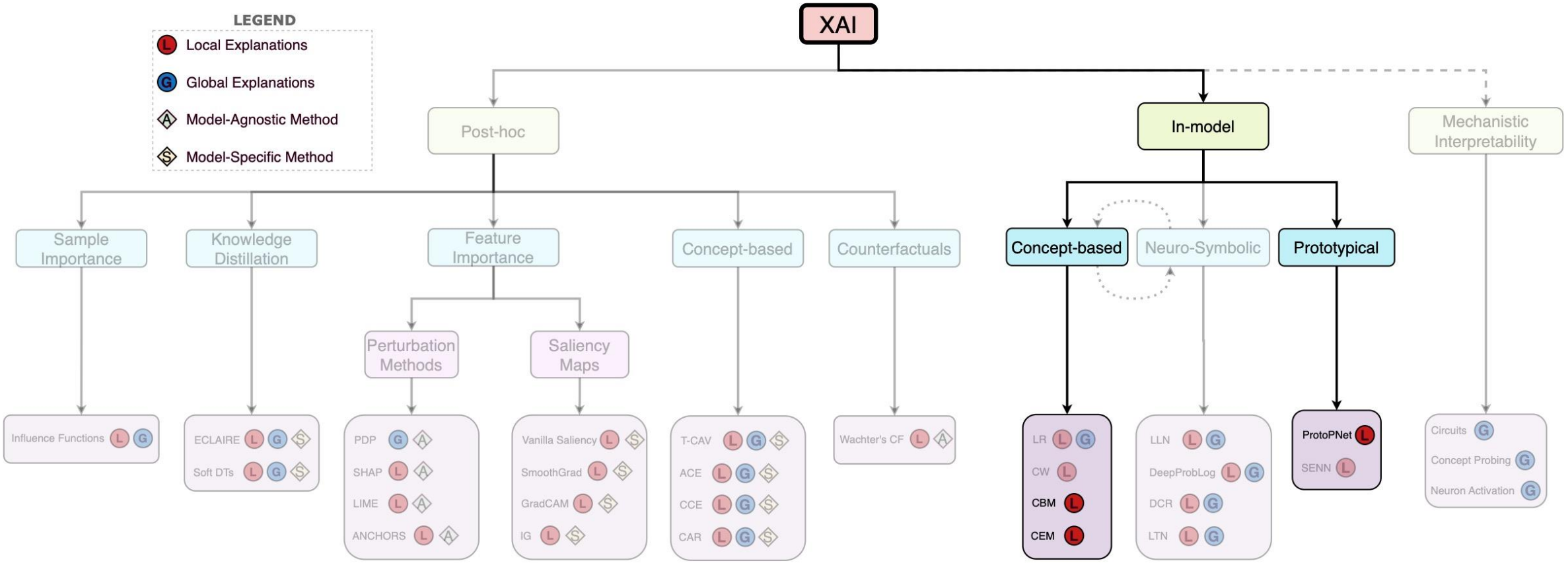
[1] Ma, Chiyu, et al. "This Looks Like Those: Illuminating Prototypical Concepts Using Multiple Visualizations."NeurIPS (2023).

ON PROTOTYPICAL EXPLANATIONS

Prototypes are certainty an interesting research direction! However:

1. The are prone to complicated training pipelines as it is non-trivial to learn "traceable" prototypes in a differentiable manner
2. They tend to select only single patches/parts of examples as prototypes, complicating disambiguating the reason behind selecting that prototype
3. Part prototypes are not very useful in a lot of domains such as language and genomics (e.g., when trying to understand memorisation)

RECAP FOR TODAY



QUESTIONS?