

EXPLAINABLE ARTIFICIAL INTELLIGENCE

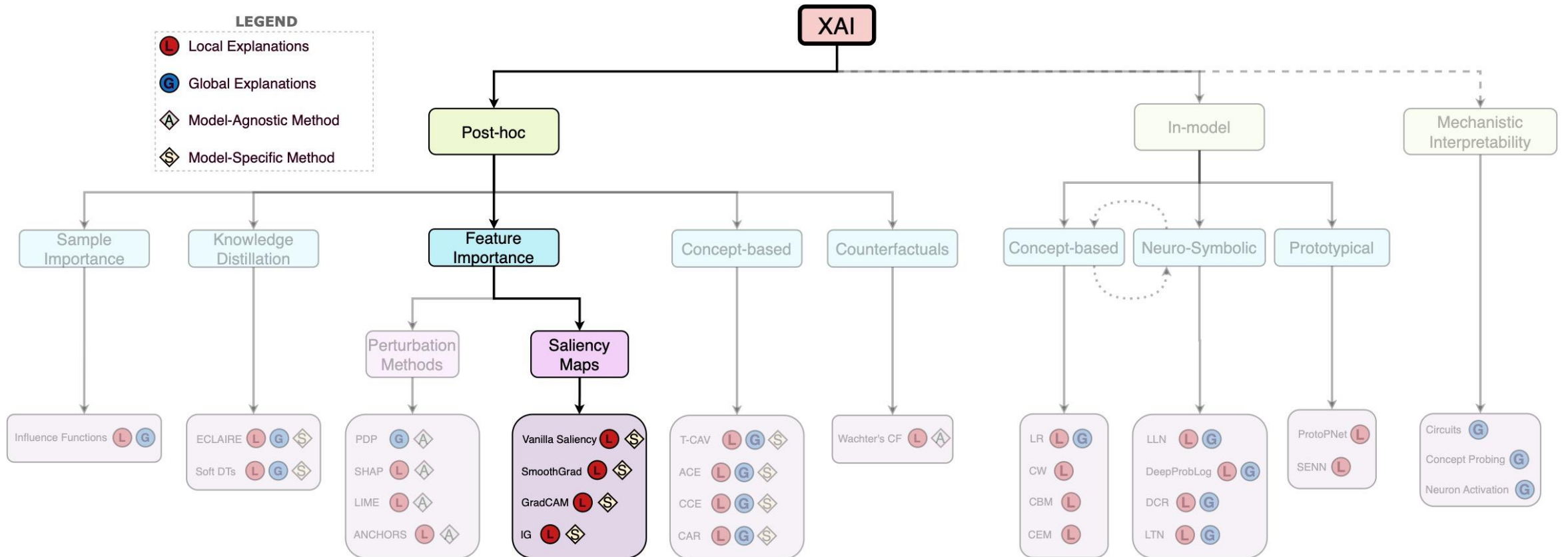
L193 – Lecture 3 – Lent 2025



UNIVERSITY OF
CAMBRIDGE



WHERE WE LEFT THINGS



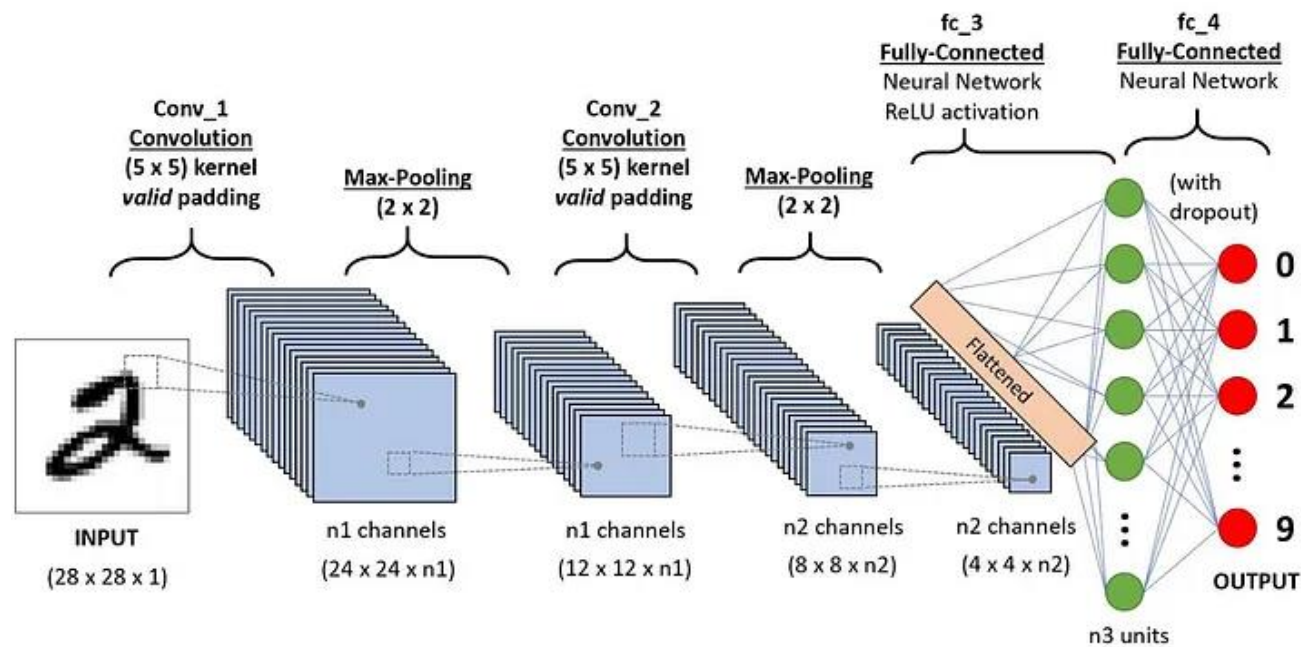
Saliency Maps:

- Vanilla Gradient
- SmoothGrad
- Grad-CAM



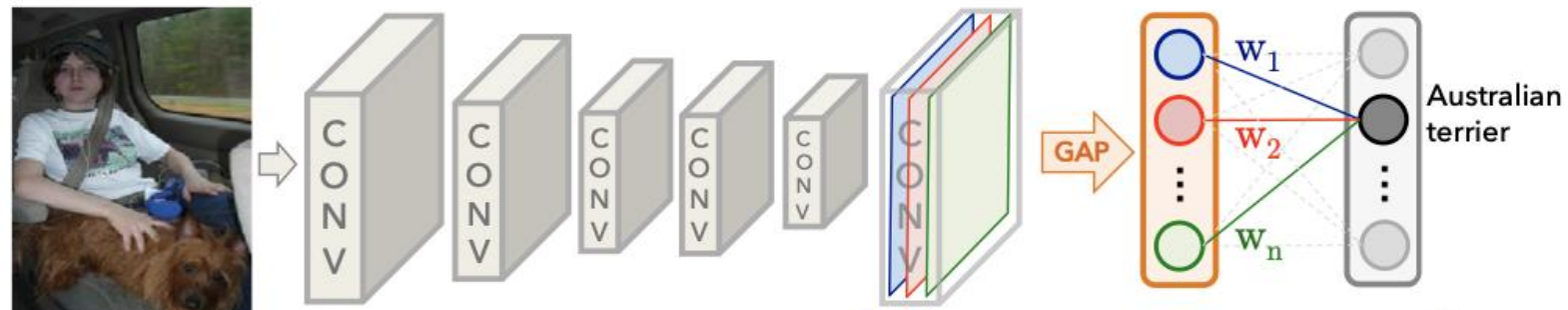
CONVENTIONAL CNNs

The **output feature maps** from the last convolutional layer are typically **flattened** and then passed into one or more **fully connected layers** before reaching the **final classification layer**



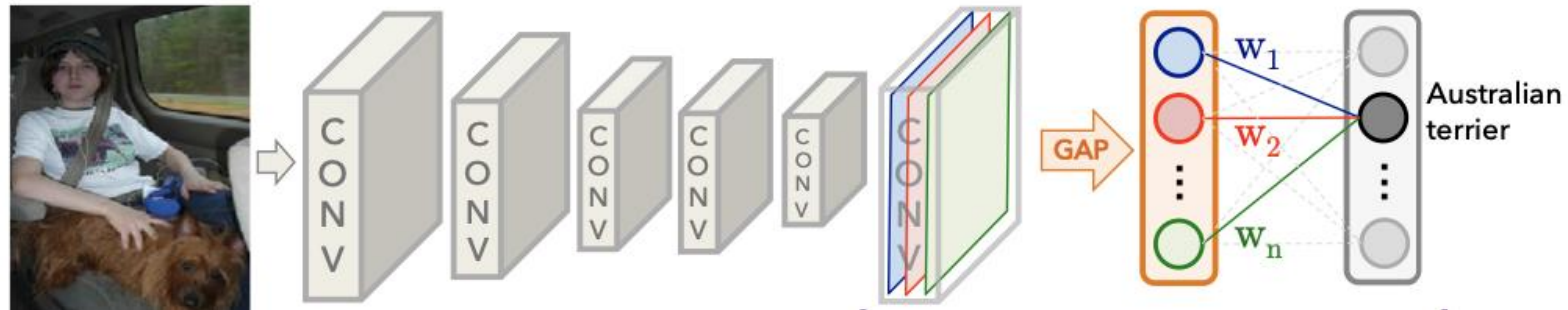
CNN WITH GLOBAL AVERAGE POOLING (GAP)

In architectures using *Global Average Pooling* (GAP) [1], the feature maps are summarized per channel using GAP, and the resulting feature vector is directly passed to the final classification layer



CNN WITH GLOBAL AVERAGE POOLING (GAP)

In architectures using *Global Average Pooling* (GAP) [1], the feature maps are summarized per channel using GAP, and the resulting feature vector is directly passed to the final classification layer



$$GAP^{(k)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W A_{ij}^{(k)}$$

Height and width of feature map

Activation value at pixel (i, j) for the k^{th} feature map

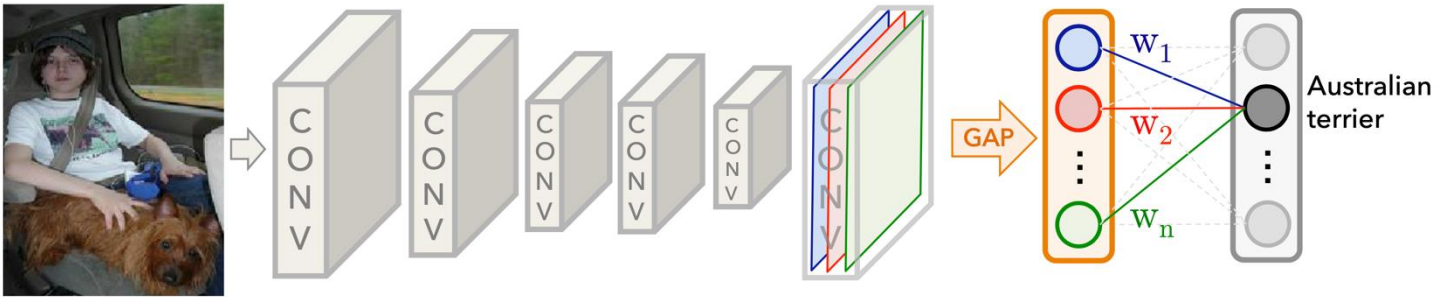
$$y^c = \sum_{k=1}^K w_k^c \cdot GAP^{(k)} + b^c$$

linear weight associated with feature map k for class c

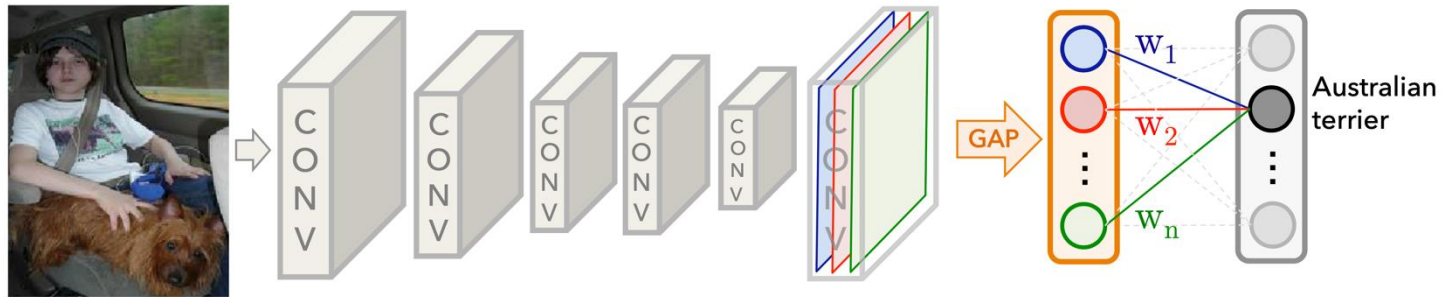
CLASS ACTIVATION MAPPING (CAM) I

- Advantages of using GAP:
 - It is parameter-free and reduces dimensionality → helps **avoid overfitting**
 - It sums out the spatial information → robust to spatial transformation (**spatial invariance**)
 - Most importantly is its **interpretability** properties: the weight of the linear layer directly translate to importance of spatial features for a class → the very idea in CAM [1]

CLASS ACTIVATION MAPPING (CAM) II

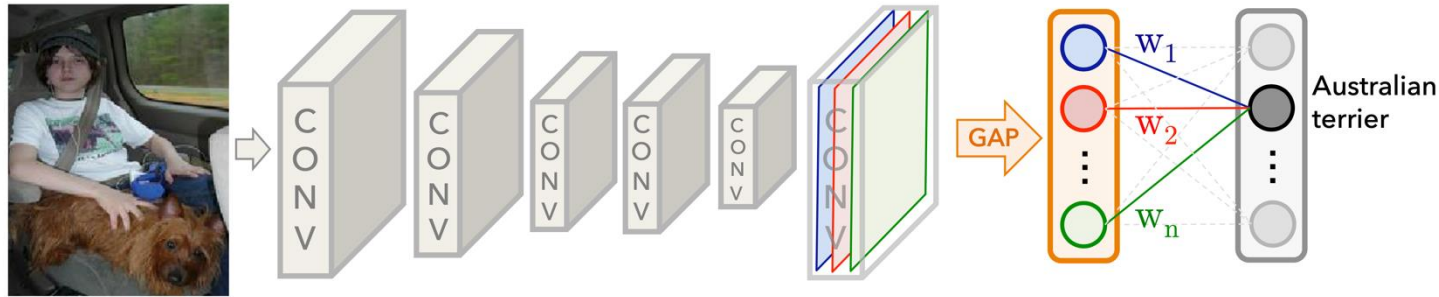


CLASS ACTIVATION MAPPING (CAM) II



Assumption: I have normalised the feature maps and have converted them each to a heatmap

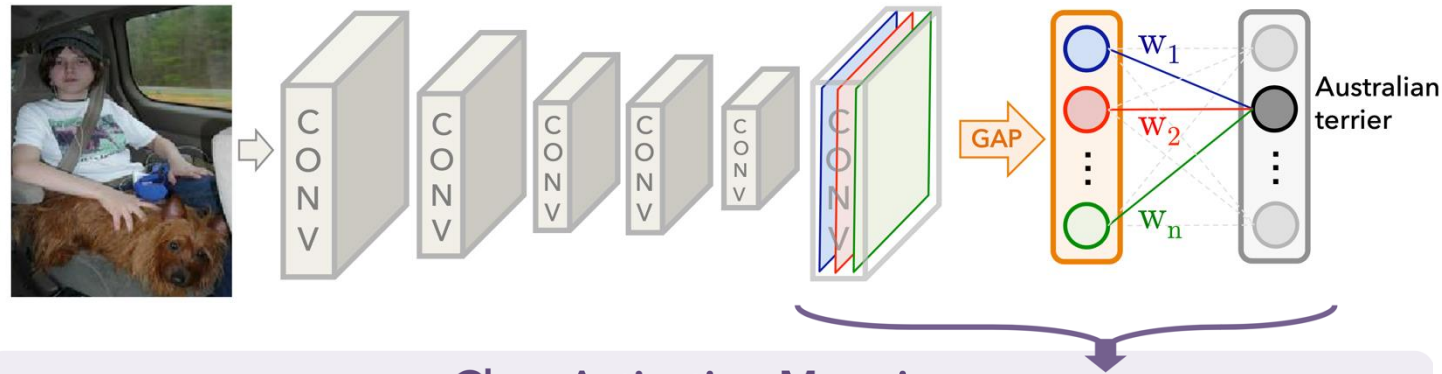
CLASS ACTIVATION MAPPING (CAM) II



Assumption: I have normalised the feature maps and have converted them each to a heatmap

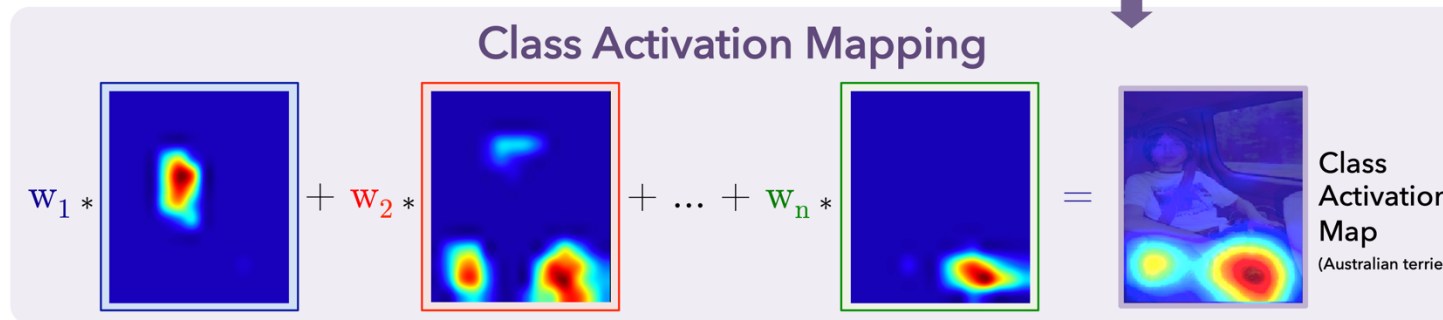
Question: Can I overlay the activation maps on the original image?

CLASS ACTIVATION MAPPING (CAM) II



Assumption: I have normalised the feature maps and have converted them each to a heatmap

Question: Can I overlay the activation maps on the original image?



$$\text{CAM heatmap for class } c : L_{CAM}^c = \sum_{k=1}^K w_k^c A^{(k)}$$

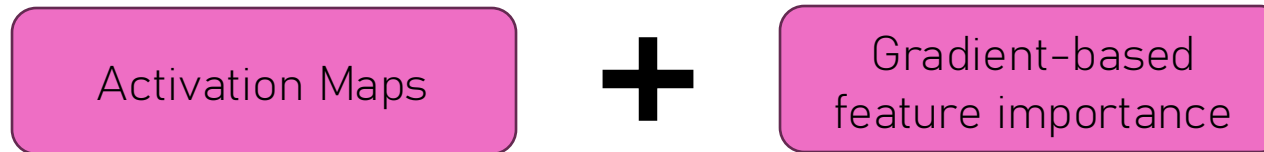
GRAD-CAM

- CAM requires GAP followed by a linear layer → very architecture specific
- Grad-CAM [1] is a generalisation of CAM that works with **any** architecture!

[1] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision (pp. 618-626).

GRAD-CAM

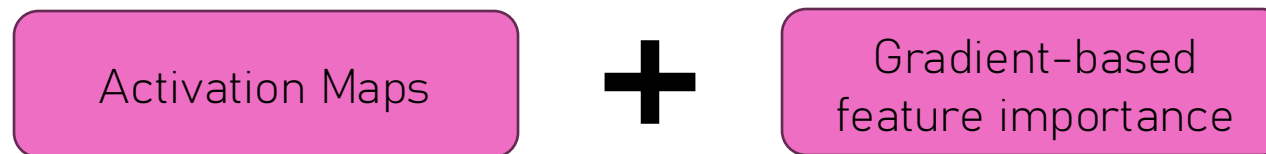
- CAM requires GAP followed by a linear layer → very architecture specific
- Grad-CAM [1] is a generalisation of CAM that works with **any** architecture!



[1] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision (pp. 618-626).

GRAD-CAM

- CAM requires GAP followed by a linear layer → very architecture specific
- Grad-CAM [1] is a generalisation of CAM that works with **any** architecture!



Unlike other gradient methods, the gradient is not back-propagated all the way to the input layer, but to the **last convolutional layer** of the CNN (feature maps)

[1] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision (pp. 618-626).

GRAD-CAM FORMULATION

- Step 1: compute gradient of class score w.r.t. feature maps

$$\alpha_k^c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \frac{\partial y^c}{\partial A_{ij}^{(k)}} \quad \text{Gradient of class score w.r.t. pixel } (i, j) \text{ in feature map } A^k$$

GRAD-CAM FORMULATION

- Step 1: compute gradient of class score w.r.t. feature maps

$$\alpha_k^c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \frac{\partial y^c}{\partial A_{ij}^{(k)}} \quad \text{Gradient of class score w.r.t. pixel } (i, j) \text{ in feature map } A^k$$

CAM

$$\text{GAP}^{(k)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W A_{ij}^{(k)}$$

GRAD-CAM FORMULATION

- Step 1: compute gradient of class score w.r.t. feature maps

$$\alpha_k^c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \frac{\partial y^c}{\partial A_{ij}^{(k)}} \quad \text{Gradient of class score w.r.t. pixel } (i, j) \text{ in feature map } A^k$$

- Step 2: Grad-CAM heatmap for class c

CAM

$$\text{GAP}^{(k)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W A_{ij}^{(k)}$$

GRAD-CAM FORMULATION

- Step 1: compute gradient of class score w.r.t. feature maps

$$\alpha_k^c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \frac{\partial y^c}{\partial A_{ij}^{(k)}} \quad \text{Gradient of class score w.r.t. pixel } (i, j) \text{ in feature map } A^k$$

- Step 2: Grad-CAM heatmap for class c

CAM

$$\text{GAP}^{(k)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W A_{ij}^{(k)}$$

$$L_{CAM}^c = \sum_{k=1}^K w_k^c A^{(k)}$$

GRAD-CAM FORMULATION

- Step 1: compute gradient of class score w.r.t. feature maps

$$\alpha_k^c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \frac{\partial y^c}{\partial A_{ij}^{(k)}} \quad \text{Gradient of class score w.r.t. pixel } (i, j) \text{ in feature map } A^k$$

- Step 2: Grad-CAM heatmap for class c

$$L_{Grad-CAM}^c = \text{ReLU} \left(\sum_{k=1}^K \alpha_k^c A^{(k)} \right)$$

CAM

$$\text{GAP}^{(k)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W A_{ij}^{(k)}$$

$$L_{CAM}^c = \sum_{k=1}^K w_k^c A^{(k)}$$

GRAD-CAM FORMULATION

- Step 1: compute gradient of class score w.r.t. feature maps

$$\alpha_k^c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \frac{\partial y^c}{\partial A_{ij}^{(k)}} \quad \text{Gradient of class score w.r.t. pixel } (i, j) \text{ in feature map } A^k$$

- Step 2: Grad-CAM heatmap for class c

$$L_{Grad-CAM}^c = \text{ReLU} \left(\sum_{k=1}^K \alpha_k^c A^{(k)} \right)$$

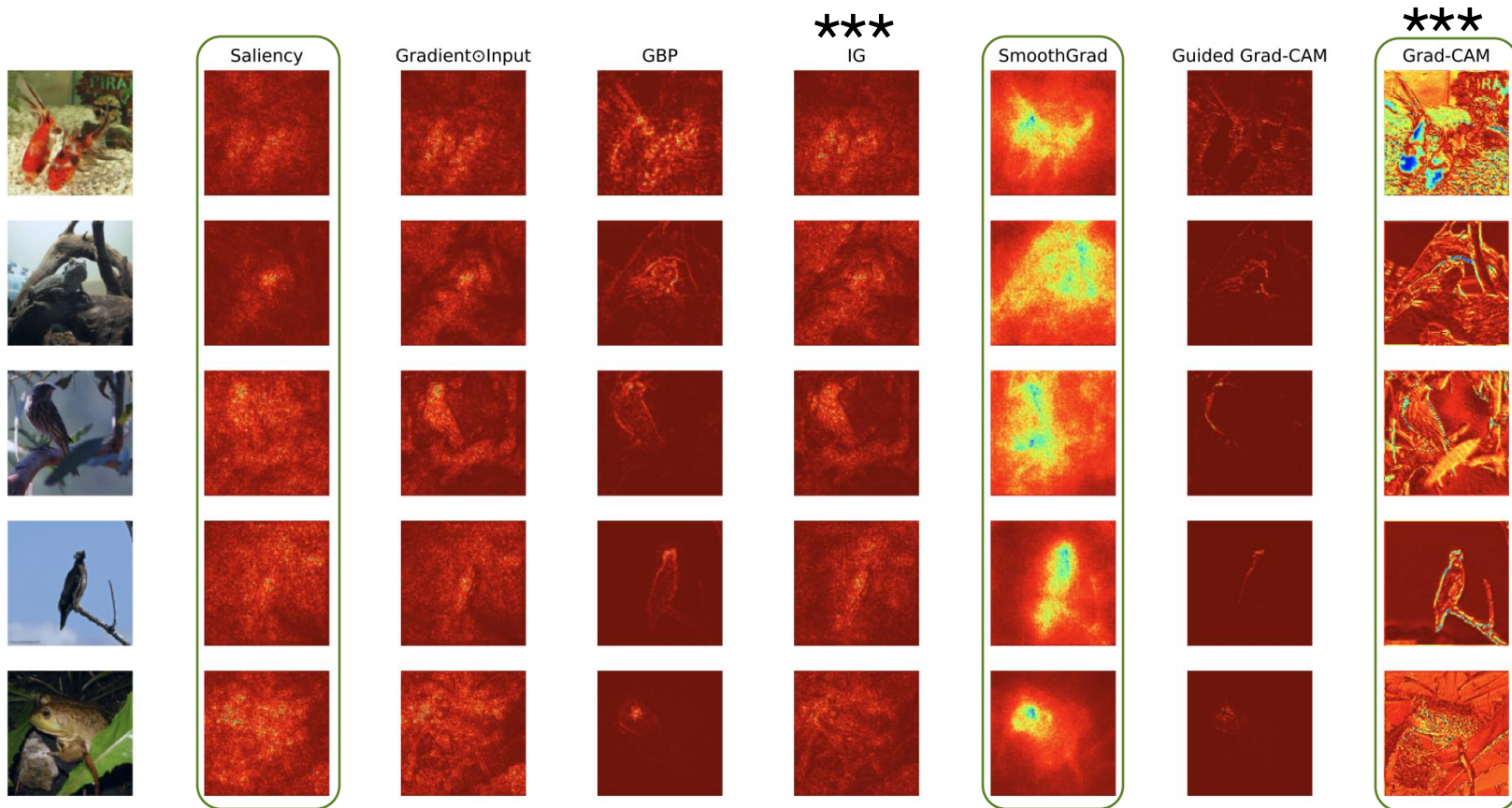
- What's the role of ReLU here?

CAM

$$\text{GAP}^{(k)} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W A_{ij}^{(k)}$$

$$L_{CAM}^c = \sum_{k=1}^K w_k^c A^{(k)}$$

FAMILY OF GRADIENT-BASED METHODS



Useful Python Library



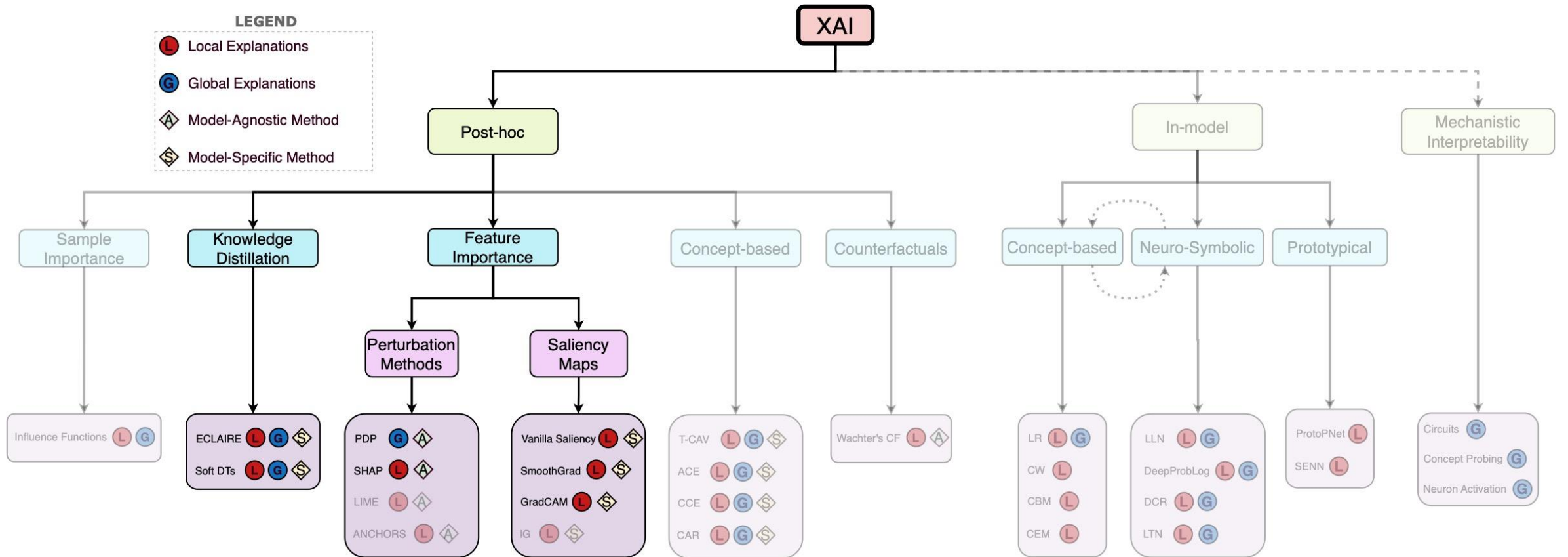
<https://github.com/albermax/investigate>

Comparative review of gradient-based methods [1]

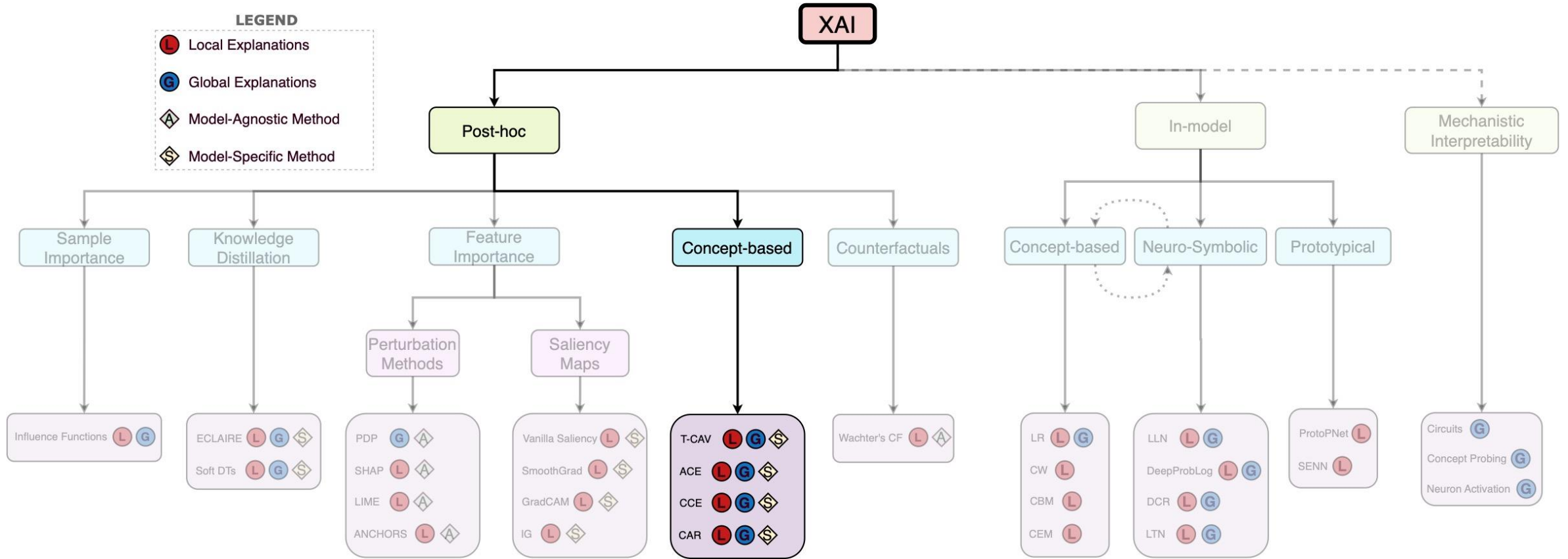
[1] Nielsen, Ian E., et al. "Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks." *IEEE Signal Processing Magazine* 39.4 (2022): 73-84.

Image taken from [1].

THE STORY SO FAR



DIFFERENT TYPES OF EXPLANATIONS



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?

WHAT'S WRONG WITH FEATURE ATTRIBUTION?

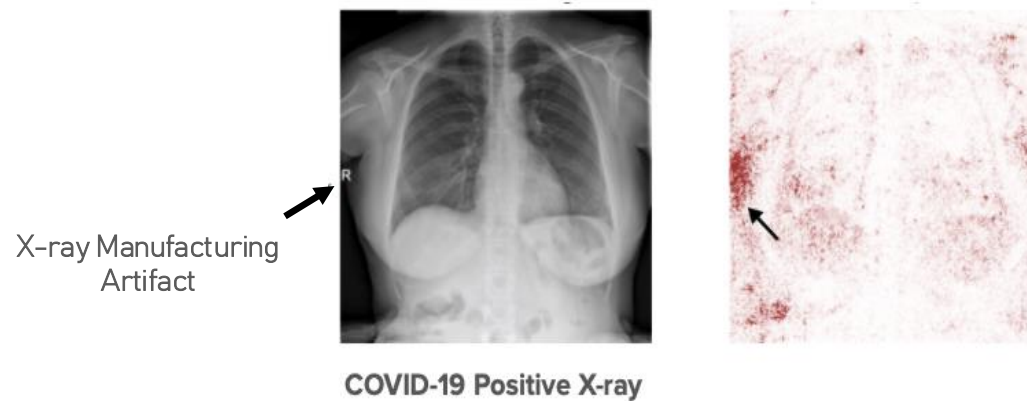
1. Low-level features like individual pixels are **not always semantically meaningful**:



Limes!

WHAT'S WRONG WITH FEATURE ATTRIBUTION?

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



WHAT'S WRONG WITH FEATURE ATTRIBUTION?

2. Feature maps lack of **actionability!**



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

WHAT'S WRONG WITH FEATURE ATTRIBUTION?

3. They are susceptible to **adversarial attacks** [1,2]



More on this next week!

[1] Dombrowski, Ann-Kathrin, et al. "Explanations can be manipulated and geometry is to blame." *Advances in Neural Information Processing Systems* 32 (2019).

[2] Ghorbani, Amirata, Abubakar Abid, and James Zou. "Interpretation of neural networks is fragile." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. No. 01. 2019.

WHAT'S WRONG WITH FEATURE ATTRIBUTION?

How can we go around the limitations of feature attribution?

- Other forms of explainability (e.g., knowledge distillation)
- **Concept-based explainability!**

WHAT ARE CONCEPTS?

Concepts are **high-level** and semantically **meaningful** units of information

Task:
bird species



Explanation of the prediction:

- wing colour
- beak length
- tail shape

WHAT ARE CONCEPTS?

Concepts are **high-level** and semantically **meaningful** units of information

Task:
bird species



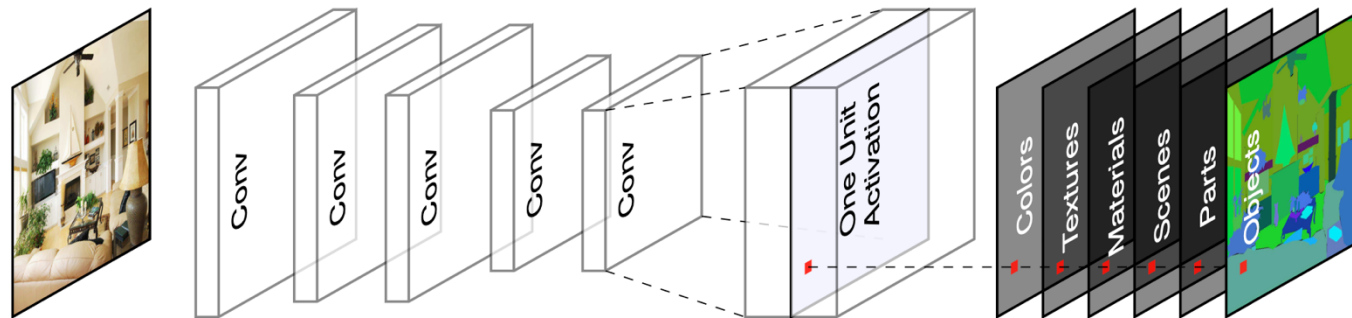
Explanation of the prediction:

- wing colour
- beak length
- tail shape

Concepts are useful when they are **used by domain experts** to communicate or explain things to one another

DO NEURAL NETWORKS NATURALLY LEARN CONCEPTS?

Evidence generally suggests that is the case: lower levels are detecting texture or surface, whilst higher levels learn more semantically meaningful concepts [1,2]



Next week you will hear in more detail about this :)

[1] Bau, David, et al. "Network dissection: Quantifying interpretability of deep visual representations." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017

[2] Fong, Ruth, and Andrea Vedaldi. "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

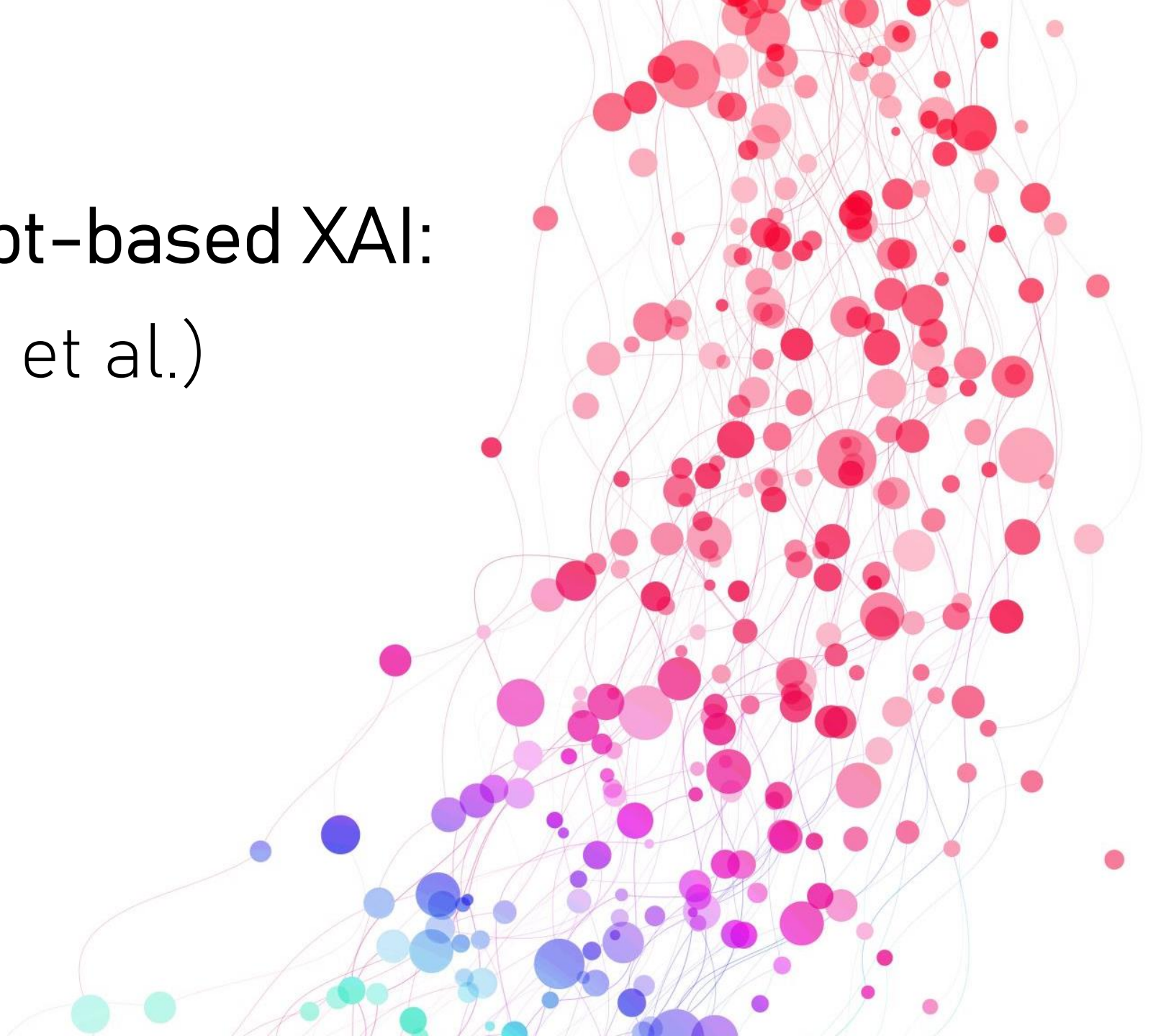
Image taken from [1]

POST-HOC
CONCEPT-BASED
EXPLAINABILITY



Post-hoc Concept-based XAI:

- T-CAV (Kim et al.)



T-CAV (CONCEPT ACTIVATION VECTOR): INTUITION

- **T-CAV** [1] provides **global** explanations for a class of interest



T-CAV (CONCEPT ACTIVATION VECTOR): INTUITION

- **T-CAV** [1] provides **global** explanations for a class of interest
- Learns concepts from examples



T-CAV (CONCEPT ACTIVATION VECTOR): INTUITION

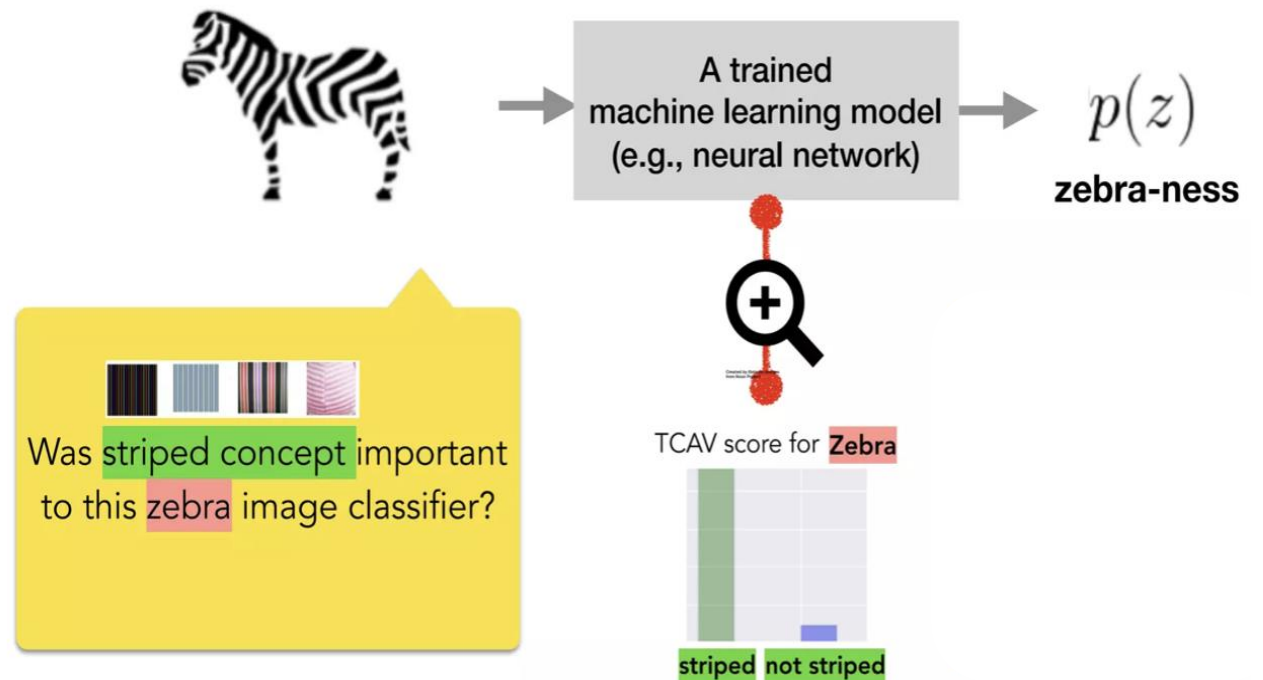
- **T-CAV** [1] provides **global** explanations for a class of interest
- Learns concepts from examples
- Quantifies the degree to which a **user-defined** concept is important to a classification result



T-CAV (CONCEPT ACTIVATION VECTOR): INTUITION

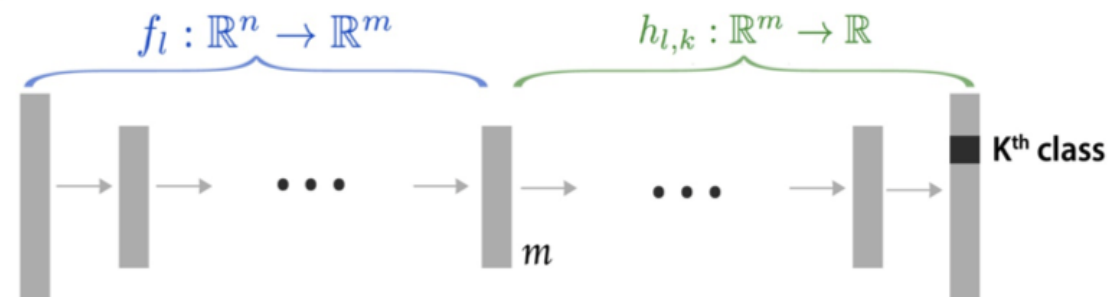
- **T-CAV** [1] provides **global** explanations for a class of interest
- Learns concepts from examples
- Quantifies the degree to which a **user-defined** concept is important to a classification result

How sensitive prediction of zebra is to the presence of stripes



T-CAV: FORMULATION

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

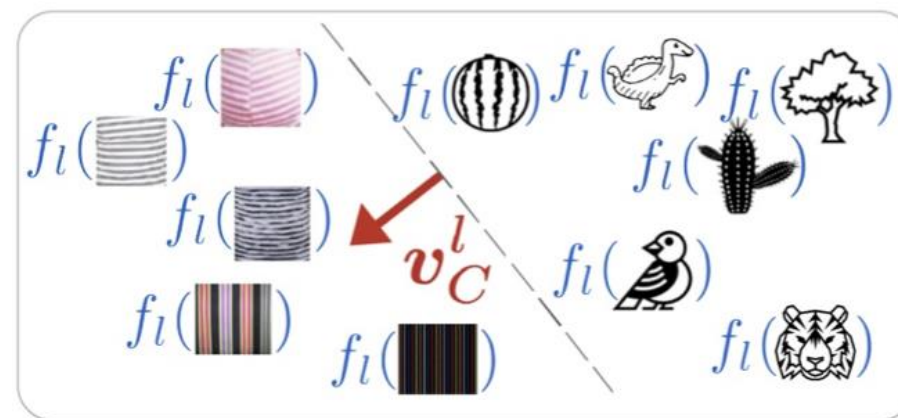


T-CAV: FORMULATION

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

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



T-CAV: T-CAV SCORES

Step 3: Getting **importance scores** from CAVs

- T-CAV gauges the *sensitivity* of class k to concept \mathcal{C}
- Given a sample's latent representation and the CAV, **how do you think we should gauge this sensitivity?**

T-CAV: T-CAV SCORES

Step 3: Getting **importance scores** from CAVs

- T-CAV gauges the *sensitivity* of class k to concept C
- Given a sample's latent representation and the CAV, **how do you think we should gauge this sensitivity?**

When in doubt, 72% of the time, the answer will be “**derivatives**”

T-CAV: T-CAV SCORES


Step 3: Getting **importance scores** from CAVs

- TCAV uses the **directional derivative**, $S_{C,k,l}(x)$ to gauge how much a classification changes with a change in a concept

Intermediate representation of
 at layer l

$$S_{C,k,l}(x) = \nabla h_{l,k}(f_l(x)) \cdot v_C^l = \nabla h_{l,k}(f_l(\text{zebra})) \cdot v_C^l$$

Output function CAV for concept C (e.g., stripes)

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

T-CAV: T-CAV SCORES

Step 3: Getting **importance scores** from CAVs

- TCAV uses the **directional derivative**, $S_{C,k,l}(x)$ to gauge how much a classification changes with a change in a concept

T-CAV score is the fraction of k -class inputs that are positively influenced by concept C :

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

X_k : inputs
with label k



Question: why not also consider the negative influences?

T-CAV: APPLICATION

Medical diagnosis – image data:

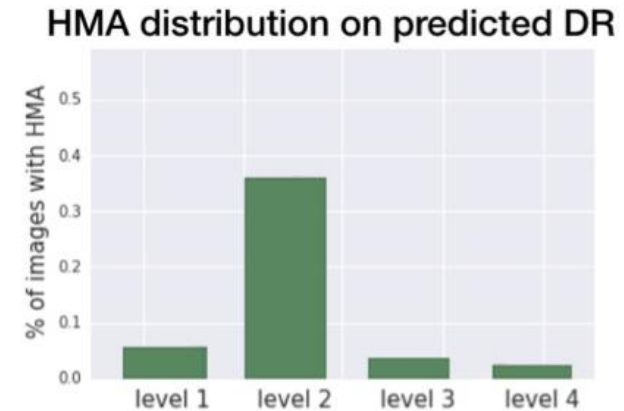
- **Task:** predicting diabetic retinopathy (DR) – level 0 (no DR) to 4 (proliferative)
- DR level depends on evaluating a set of diagnostic concepts, such as microaneurysms (MA) or aneurysms (HMA)
- Different concepts are more prominent at different DR levels



T-CAV: APPLICATION

Medical diagnosis – image data:

- **Task:** predicting diabetic retinopathy (DR) – level 0 (no DR) to 4 (proliferative)
- DR level depends on evaluating a set of diagnostic concepts, such as microaneurysms (MA) or aneurysms (HMA)
- Different concepts are more prominent at different DR levels

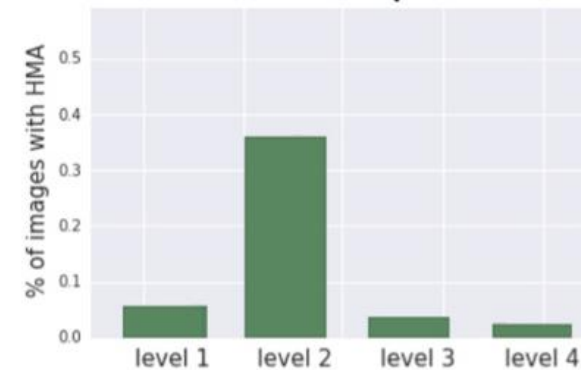


T-CAV: APPLICATION

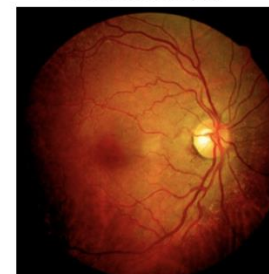
Medical diagnosis – image data:

- **Task:** predicting diabetic retinopathy (DR) – level 0 (no DR) to 4 (proliferative)
- DR level depends on evaluating a set of diagnostic concepts, such as microaneurysms (MA) or aneurysms (HMA)
- Different concepts are more prominent at different DR levels

HMA distribution on predicted DR



DR level 1 Retina



TCAV for DR level 1

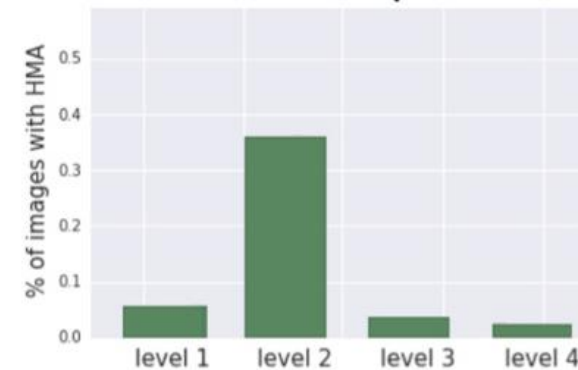


T-CAV: APPLICATION

Medical diagnosis – image data:

- **Task:** predicting diabetic retinopathy (DR) – level 0 (no DR) to 4 (proliferative)
- DR level depends on evaluating a set of diagnostic concepts, such as microaneurysms (MA) or aneurysms (HMA)
- Different concepts are more prominent at different DR levels

HMA distribution on predicted DR



DR level 1 Retina



TCAV for DR level 1



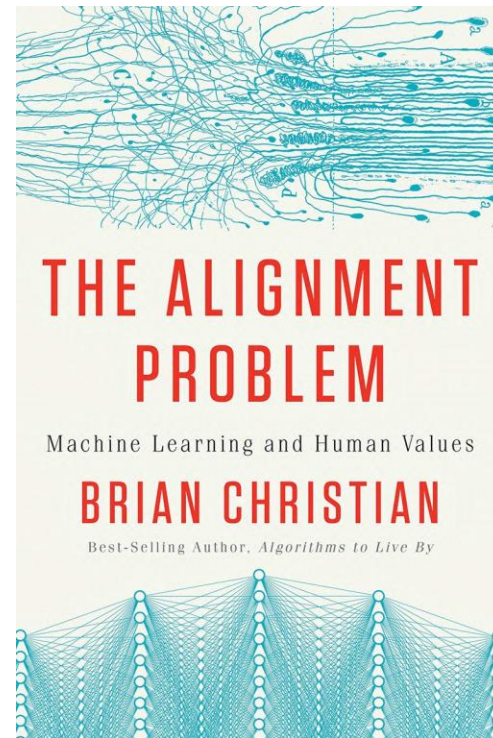
Model error exposed: TCAV score for HMA feature is too high for level 1

THE COST OF BEING GREAT

T-CAV is great!

THE COST OF BEING GREAT

T-CAV is great! So great it is even discussed in a science communication book



THE COST OF BEING GREAT

However, 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”

This is what we call a **concept-supervised approach!**

THE COST OF BEING GREAT

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

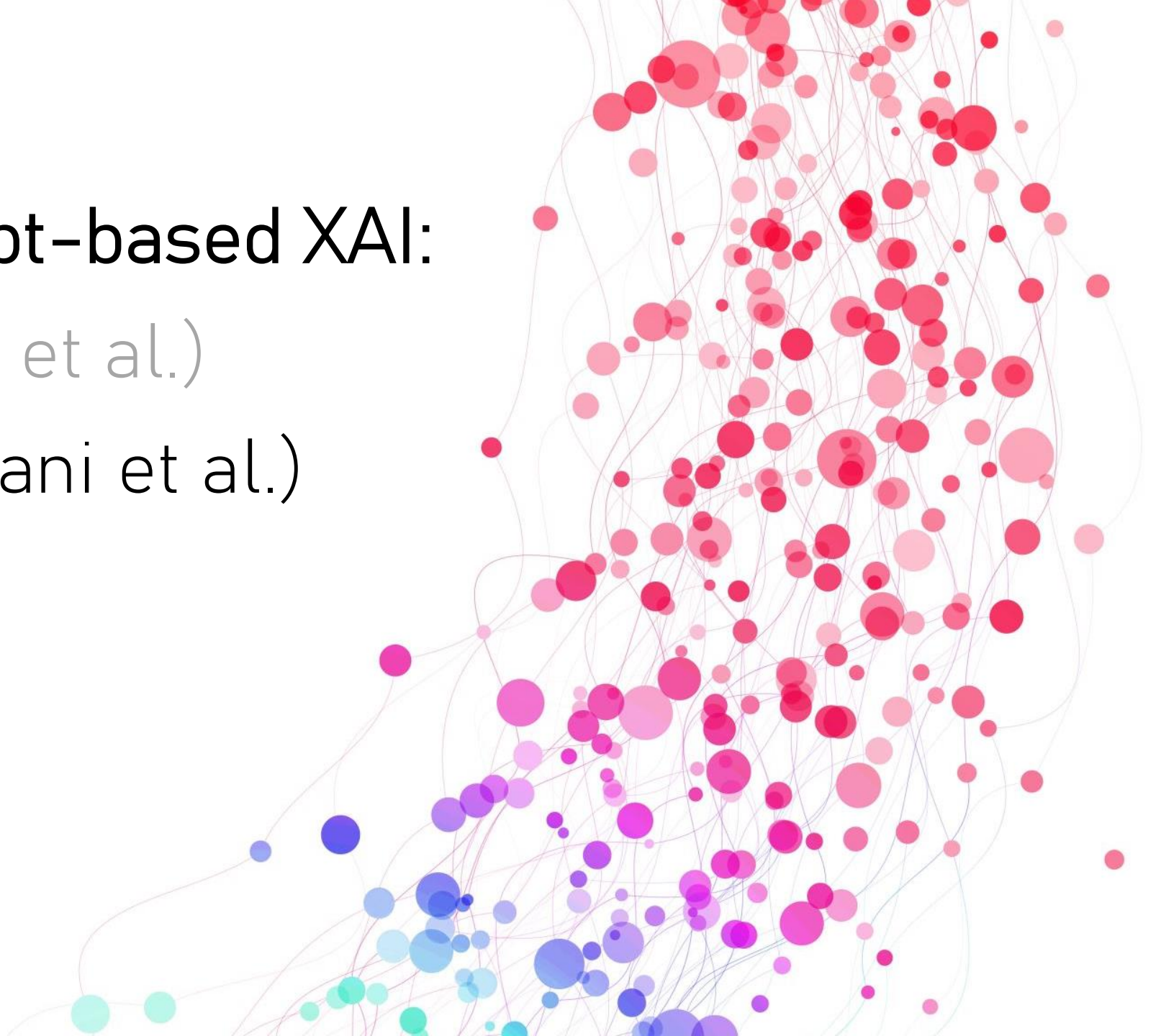


For example, when finding the influence of the concept “stripes” for a DNN, T-CAV requires a set of samples that all have the concept “stripes”

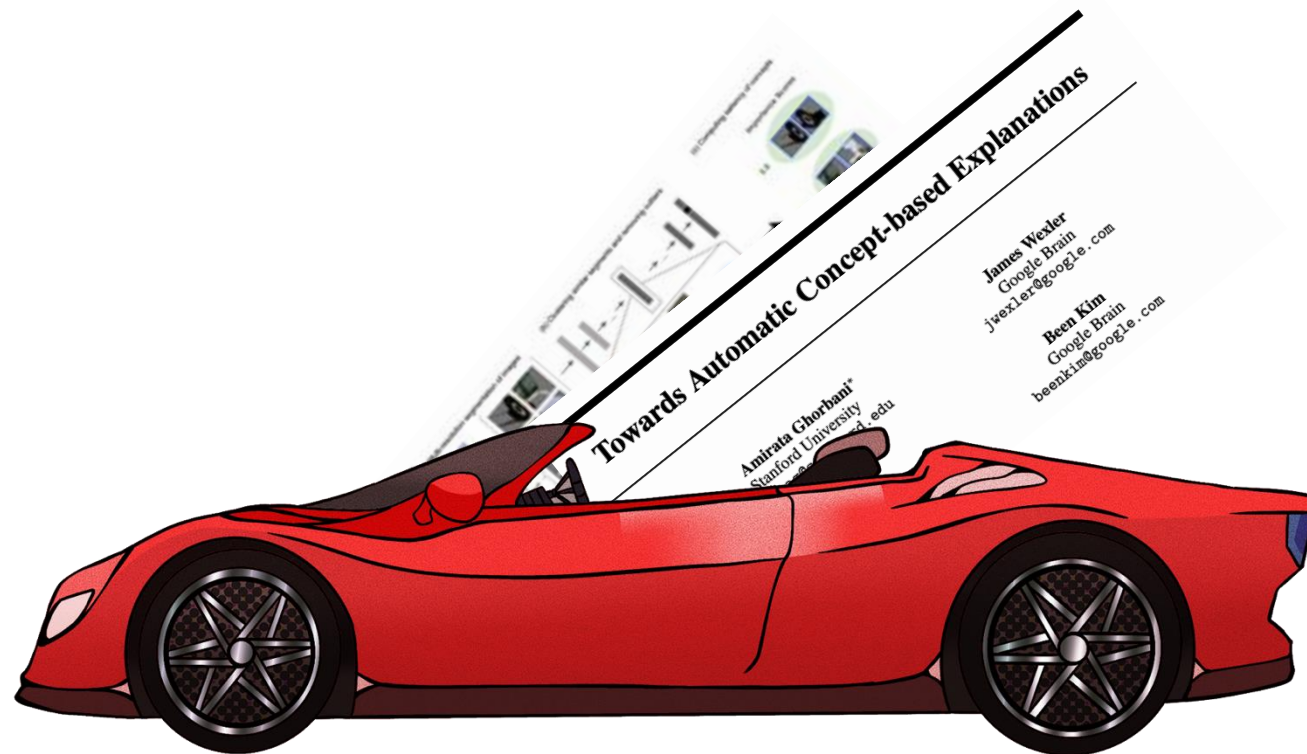
But, obtaining concept labels can be **expensive** and **intractable**

Post-hoc Concept-based XAI:

- T-CAV (Kim et al.)
- ACE (Ghorbani et al.)



GOING UNSUPERVISED



"Wouldn't it be nice if T-CAV concepts could be automatically discovered?"

- Amirata Ghorbani et al. (Probably...)



GOING UNSUPERVISED



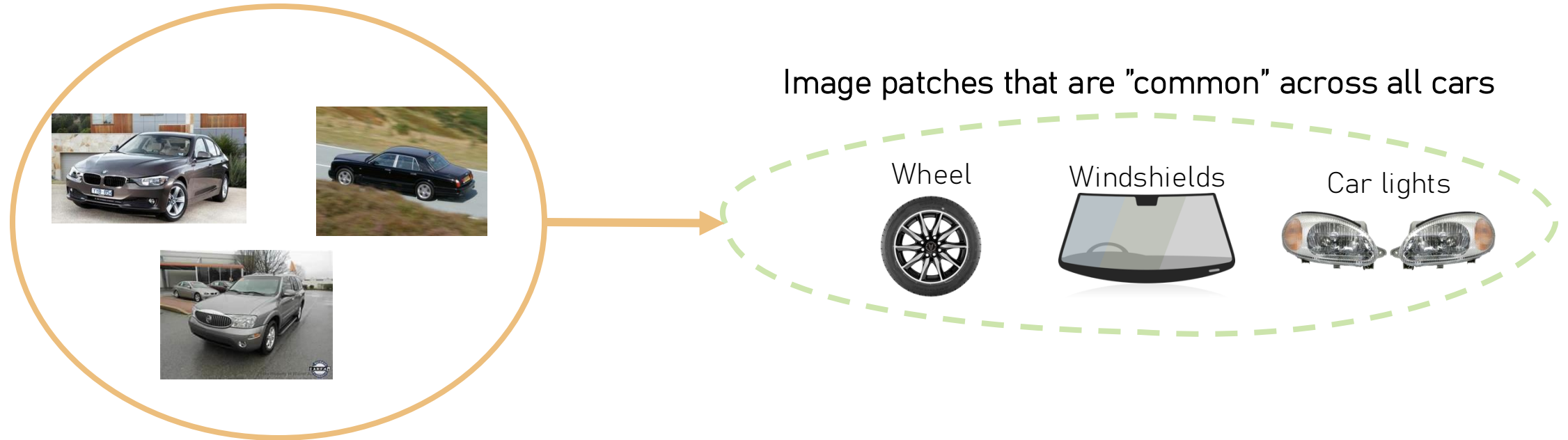
We would like to discover concepts that are:

1. Meaningful
2. Coherent
3. Important

TOWARDS AUTOMATIC CONCEPT EXTRACTION

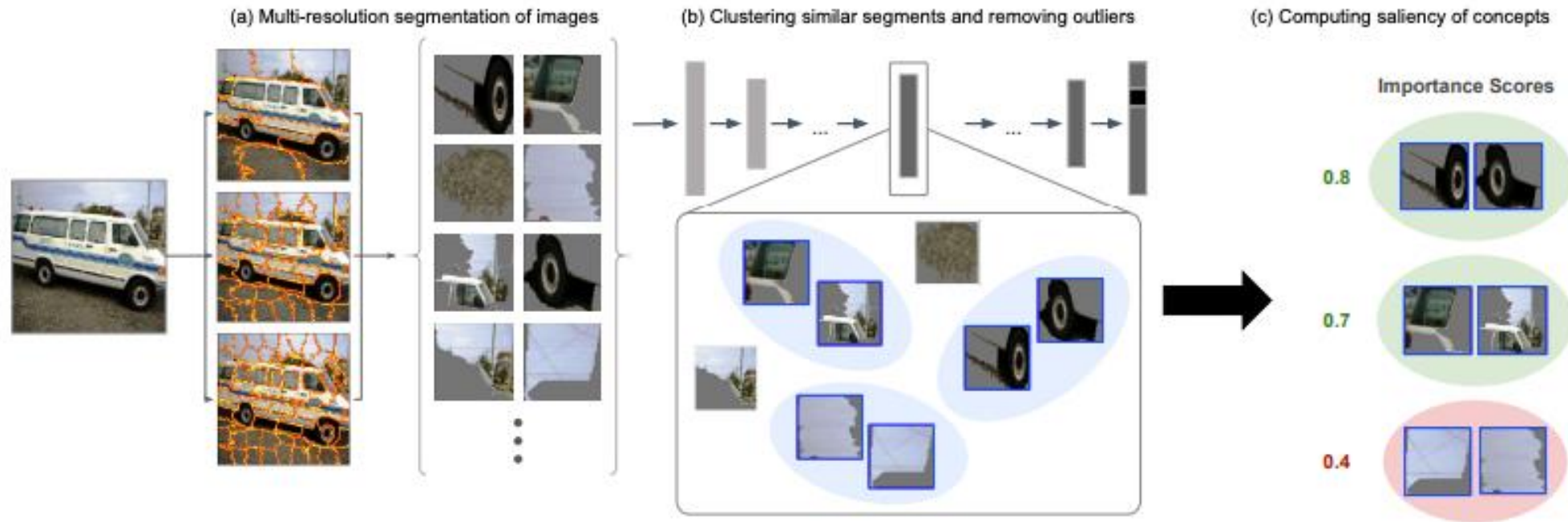
Main Idea: patches of pixels found across images can be thought of as concepts!

Training examples for class "car"



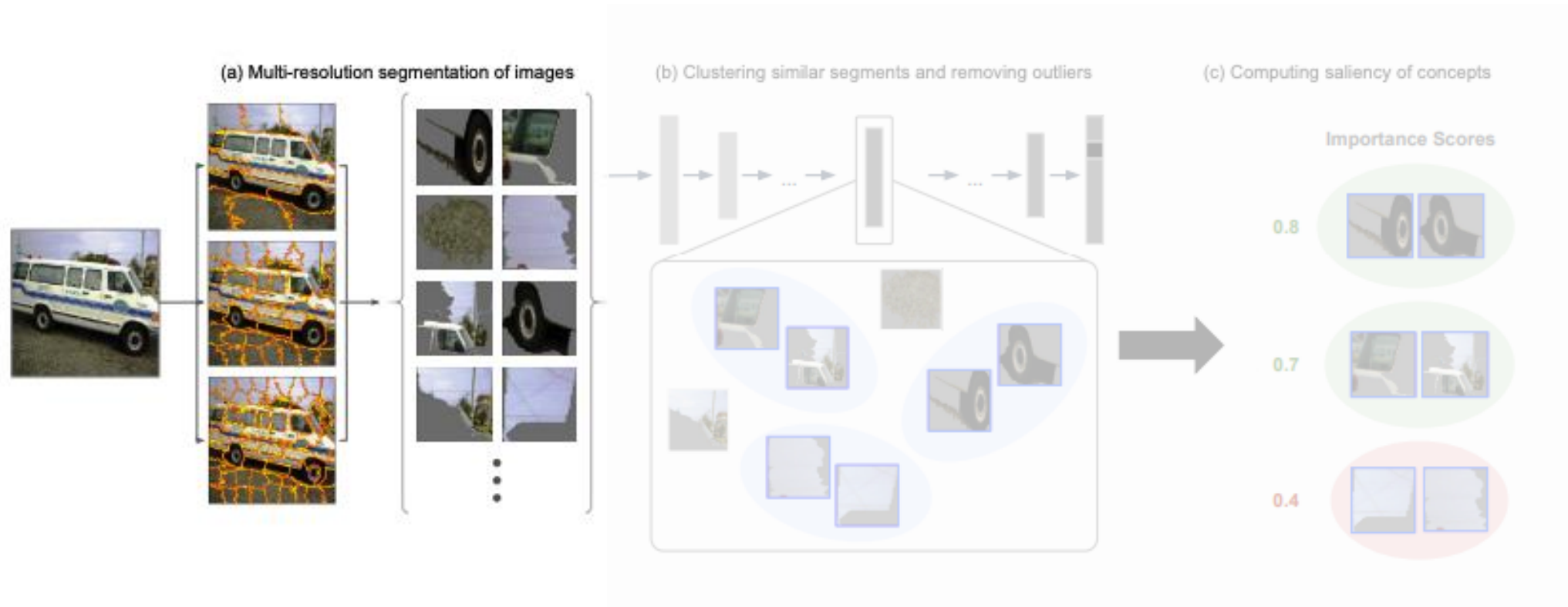
TOWARDS AUTOMATIC CONCEPT EXTRACTION

Main Idea: patches of pixels found across images can be thought of as concepts!



Here's an architecture that can do this!

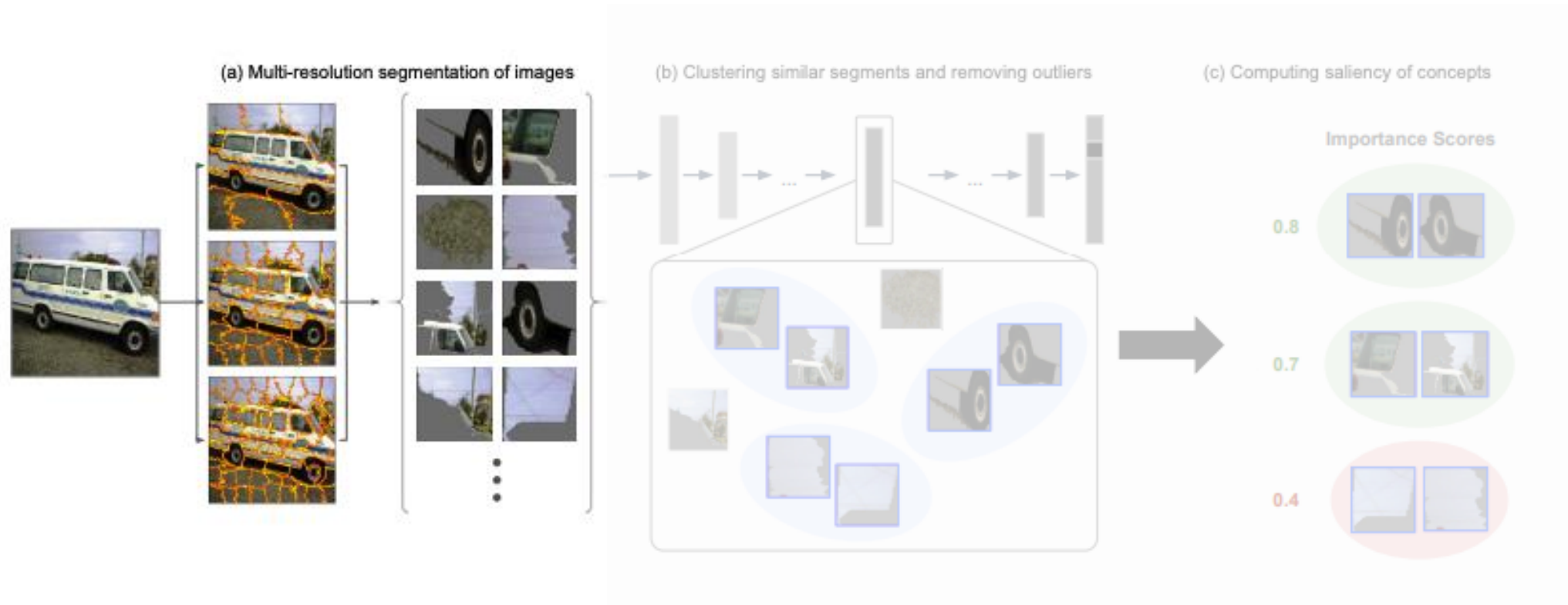
ACE: ARCHITECTURE



Step 1: segment the sample across multiple-resolutions (**why?**)

Desiderata enforced: meaningfulness

ACE: ARCHITECTURE

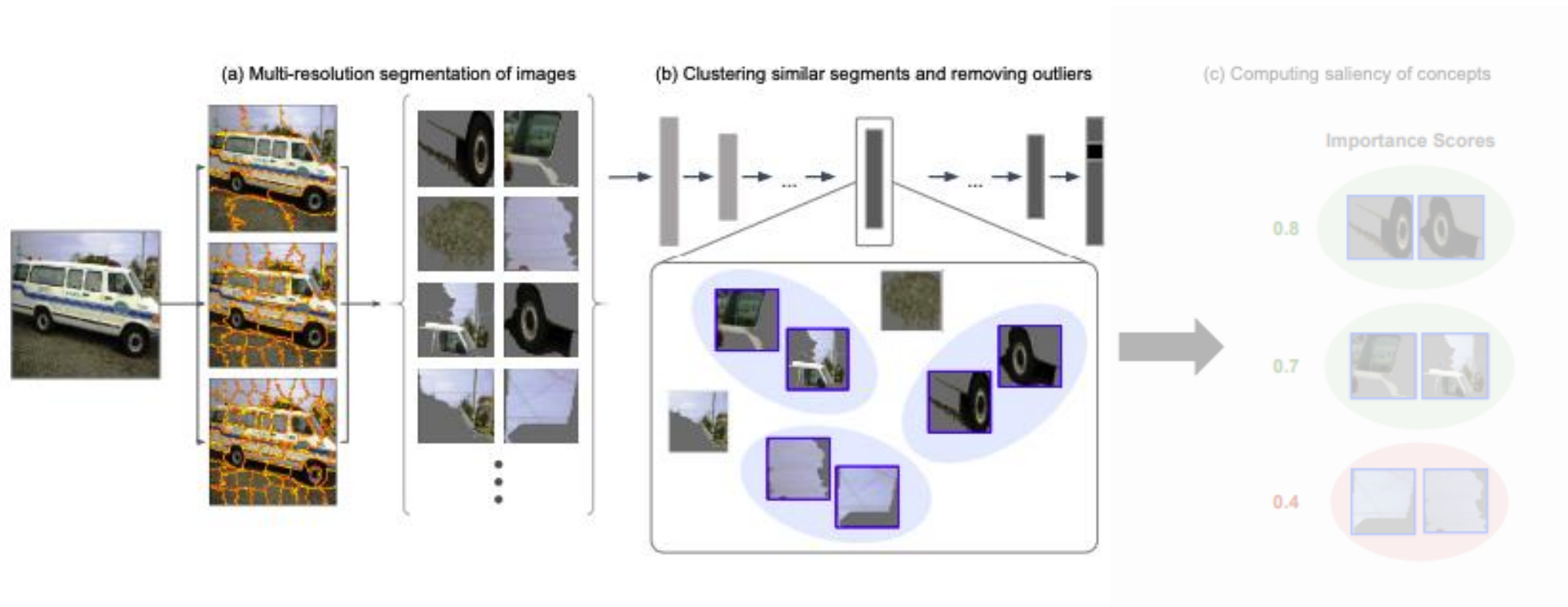


Step 1: segment the sample across multiple-resolutions (**why?**)
ACE uses SLIC [1], a fast* multi-resolution segmentation algorithm.

Desiderata enforced: meaningfulness

[1] Achanta, Radhakrishna, et al. "SLIC superpixels compared to state-of-the-art superpixel methods." *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012): 2274-2282.

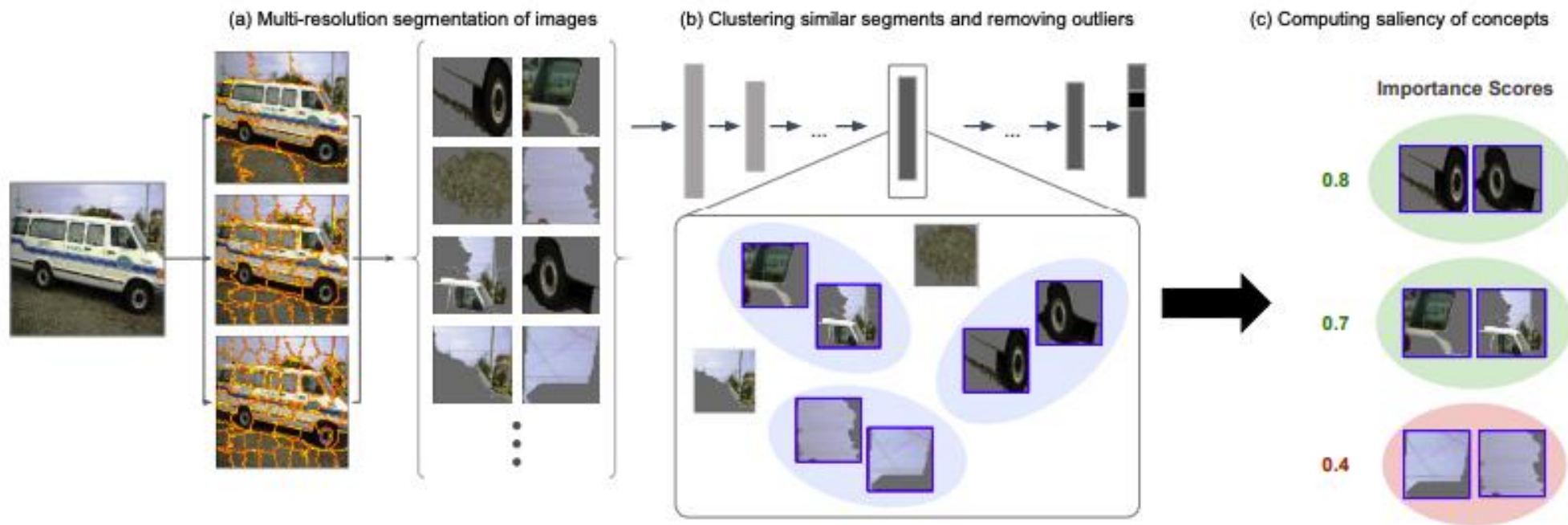
ACE: ARCHITECTURE



Step 2: cluster extracted segments using a hidden layer of a CNN as a feature extractor (**why?**). Then get rid of outliers as these are not useful concepts.

Desiderata enforced: coherence

ACE: ARCHITECTURE



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

Desiderata enforced: importance

ACE IN THE WILD



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

ACE IN THE WILD



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

Why do you think the jersey is a more salient concept for “Basketball” than the actual ball?

ACE IN THE WILD



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] Figure adapted from: Ghorbani, Amirata, et al. "Towards automatic concept-based explanations." *Advances in Neural Information Processing Systems* 32 (2019).

[2] Magister, Lucie Charlotte, et al. "GCExplainer: Human-in-the-Loop Concept-based Explanations for Graph Neural Networks." *arXiv preprint arXiv:2107.11889* (2021).

GROUNDING 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!

GROUNDING ACE

ACE's hyperparameters and processing steps have **several limitations**:

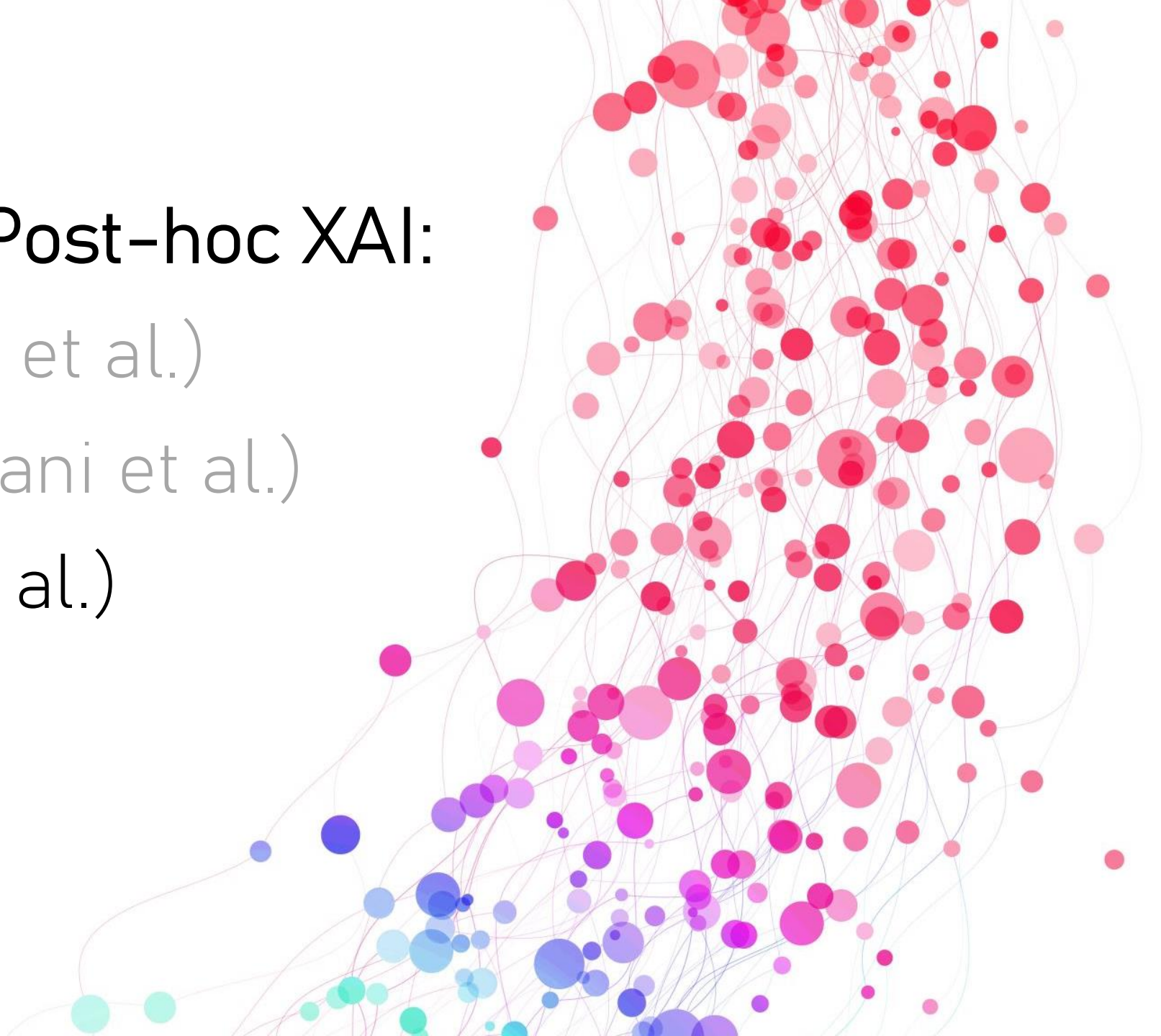
1. We can never be certain that we properly **cover all useful concepts**
2. We won't detect concepts that **interact non-linearly** with the output labels

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

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

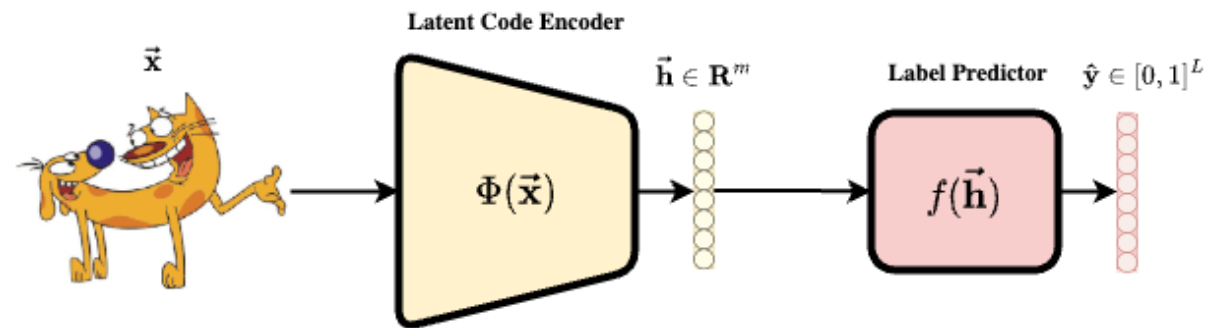
Concept-based Post-hoc XAI:

- T-CAV (Kim et al.)
- ACE (Ghorbani et al.)
- CCE (Yeh et al.)



COMPLETENESS-AWARE CONCEPT EXTRACTION

CCE [1] explains a DNN $\psi(\mathbf{x})$ by discovering a **complete set of concepts**.



We assume that $\psi(\mathbf{x})$ can be decomposed into:

1. A mapping Φ from the inputs \mathbf{x} to an intermediate hidden layer $\Phi(\mathbf{x})$; and
2. A mapping f from that intermediate hidden layer $\Phi(\mathbf{x})$ to the output layer's prediction.



LEARNING CONCEPT MATRICES

Main Idea: learn a **matrix of concept vectors** $\mathbf{C} \in \mathbb{R}^{k \times m}$ and use a “*concept completeness score*” to measure their completeness:

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

LEARNING CONCEPT MATRICES

Main Idea: learn a **matrix of concept vectors** $\mathbf{C} \in \mathbb{R}^{k \times m}$ and use a “*concept completeness score*” to measure their completeness:

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

Then, update \mathbf{C} by optimising this score!

LEARNING CONCEPT MATRICES

Main Idea: learn a **matrix of concept vectors** $\mathbf{C} \in \mathbb{R}^{k \times m}$ and use a “*concept completeness score*” to measure their completeness:

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

Looks very scary but is is saying a very simple thing:

“If I project the hidden state into the concept space defined by \mathbf{C} , can I faithfully reconstruct it afterwards?”

Then, update \mathbf{C} by optimising this score!

LEARNING CONCEPT MATRICES

Main Idea: learn a **matrix of concept vectors** $\mathbf{C} \in \mathbb{R}^{k \times m}$ and use a “*concept completeness score*” to measure their completeness:

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

Looks very scary but is is saying a very simple thing:

“If I project the hidden state into the concept space defined by \mathbf{C} , can I faithfully reconstruct it afterwards?”

Then, update \mathbf{C} by optimising this score!

It measures “faithfulness” by looking at the DNN’s accuracy when the hidden layer is replaced by its reconstruction!

COMPLETENESS VIA RECONSTRUCTION

We define a concept set's "completeness score" w.r.t. DNN $\psi(\mathbf{x})$ as:

$$\text{ConceptCompleteness}(\mathbf{c}_1, \dots, \mathbf{c}_m) := \frac{\sup_{(x,y) \sim \mathcal{D}} \text{Acc}_{(x,y) \sim \mathcal{D}} \left(f \left(g(\mathbf{C} \Phi(\mathbf{x})) \right), y \right) - \text{Acc}_{(x,y) \sim \mathcal{D}}(\text{random label}, y)}{\text{Acc}_{(x,y) \sim \mathcal{D}}(f(\Phi(\mathbf{x})), y) - \text{Acc}_{(x,y) \sim \mathcal{D}}(\text{random label}, y)}$$

Where:

1. $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m]^T$ is the matrix of concept vectors \rightarrow this is the set of concept vectors we are evaluating!
2. \mathcal{D} is our testing dataset

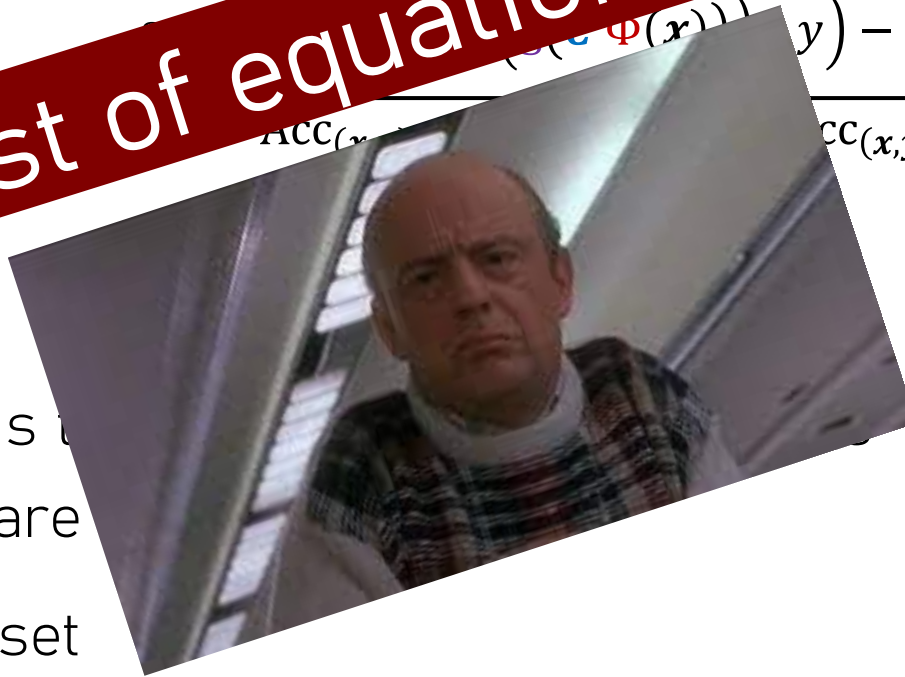
COMPLETENESS VIA RECONSTRUCTION

We define a concept set's "completeness score" wrt \mathcal{D}

$$\text{ConceptCompleteness}(\mathbf{C}, \mathcal{D}) = \frac{\text{Acc}_{(x,y) \sim \mathcal{D}}(\text{argmax}_{c \in \mathbf{C}} \langle c, \phi(x) \rangle), y) - \text{Acc}_{(x,y) \sim \mathcal{D}}(\text{random label}, y)}{\text{Acc}_{(x,y) \sim \mathcal{D}}(\text{random label}, y)}$$

Not the nicest of equations to look at!

1. $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m]^T$ is the set of concept vectors we are interested in → this is the set of
2. \mathcal{D} is our testing dataset



COMPLETENESS VIA RECONSTRUCTION

We define a concept set's "completeness score" w.r.t. DNN $\psi(\mathbf{x})$ as:

Can we learn a model g that can reconstruct the hidden layer from the concept scores?

$$\text{ConceptCompleteness}(\mathbf{c}_1, \dots, \mathbf{c}_m) := \frac{\sup_{\mathbf{x}, y \sim \mathcal{D}} \text{Acc}_{(\mathbf{x}, y) \sim \mathcal{D}} \left(f \left(g(\mathbf{C} \Phi(\mathbf{x})) \right), y \right) - \text{Acc}_{(\mathbf{x}, y) \sim \mathcal{D}}(\text{random label}, y)}{\text{Acc}_{(\mathbf{x}, y) \sim \mathcal{D}}(f(\Phi(\mathbf{x})), y) - \text{Acc}_{(\mathbf{x}, y) \sim \mathcal{D}}(\text{random label}, y)}$$

COMPLETENESS VIA RECONSTRUCTION

We define a concept set's "completeness score" w.r.t. DNN $\psi(\mathbf{x})$ as:

Can we learn a model g that can reconstruct the hidden layer from the concept scores?

$$\text{ConceptCompleteness}(\mathbf{c}_1, \dots, \mathbf{c}_m) := \frac{\sup_g \text{Acc}_{(x,y) \sim \mathcal{D}} \left(f \left(g(\mathbf{C} \Phi(\mathbf{x})) \right), y \right) - \text{Acc}_{(x,y) \sim \mathcal{D}}(\text{random label}, y)}{\text{Acc}_{(x,y) \sim \mathcal{D}}(f(\Phi(\mathbf{x})), y) - \text{Acc}_{(x,y) \sim \mathcal{D}}(\text{random label}, y)}$$

And adjust for the accuracy of a random (why?)

COMPLETENESS VIA RECONSTRUCTION

We define a concept set's "completeness score" w.r.t. DNN $\psi(\mathbf{x})$ as:

$$\text{ConceptCompleteness}(\mathbf{c}_1, \dots, \mathbf{c}_m) := \frac{\sup_{(x,y) \sim \mathcal{D}} \text{Acc}_{(x,y) \sim \mathcal{D}} \left(f \left(g(\mathbf{c} \Phi(\mathbf{x})) \right), y \right) - \text{Acc}_{(x,y) \sim \mathcal{D}}(\text{random label}, y)}{\text{Acc}_{(x,y) \sim \mathcal{D}}(f(\Phi(\mathbf{x})), y) - \text{Acc}_{(x,y) \sim \mathcal{D}}(\text{random label}, y)}$$

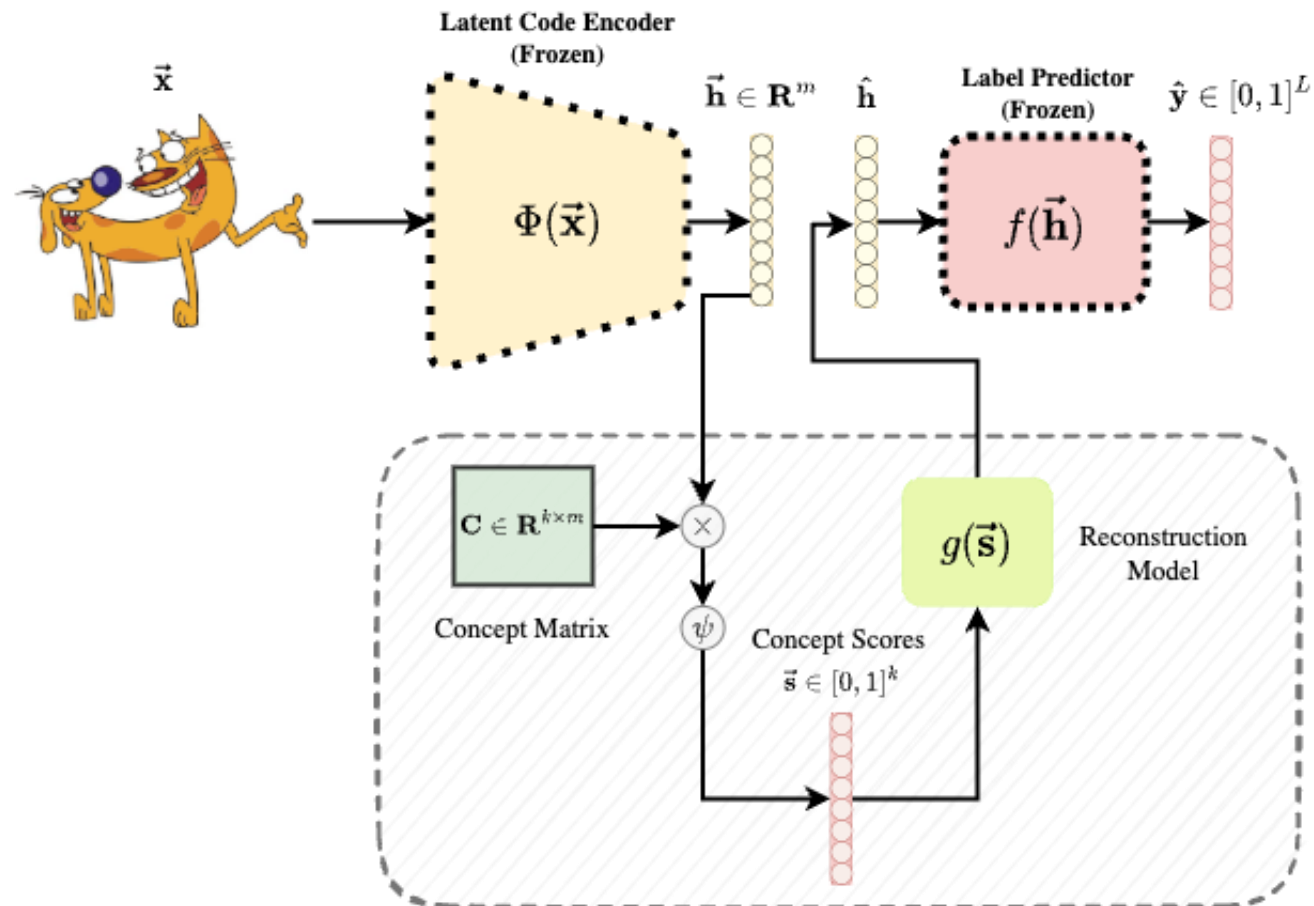
Let's learn a set of concept vectors that maximises this metric!

COMPLETENESS-AWARE CONCEPT EXTRACTION

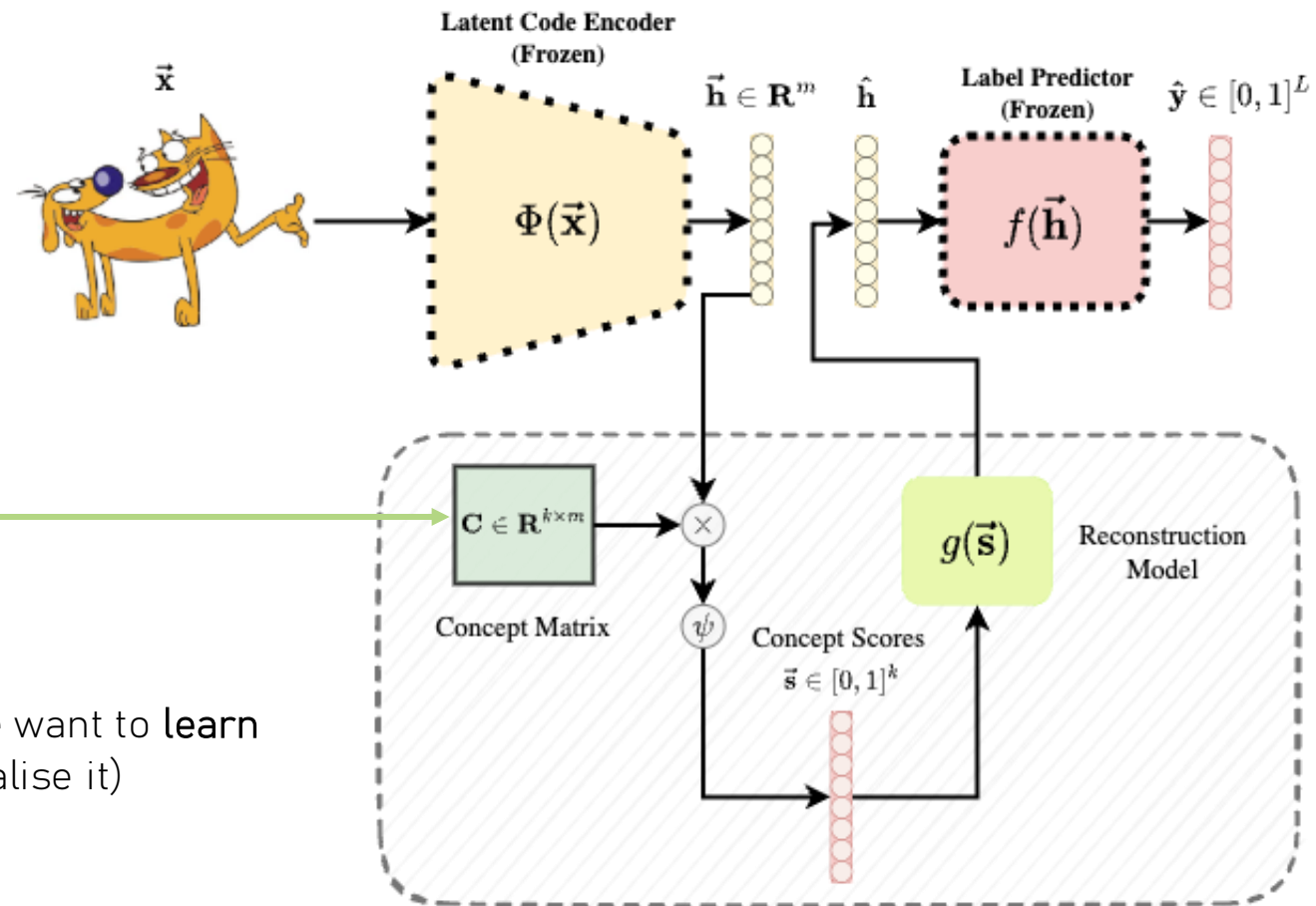
We want to learn k concept vectors $\mathcal{C} \in \mathbb{R}^{k \times m}$ such that:

1. Each vector represents a **distinct concept direction**
2. When a hidden layer of the input DNN is projected into the concept space, their resulting score **preserves all the information needed** to reconstruct the hidden layer.

COMPLETENESS-AWARE CONCEPT EXTRACTION

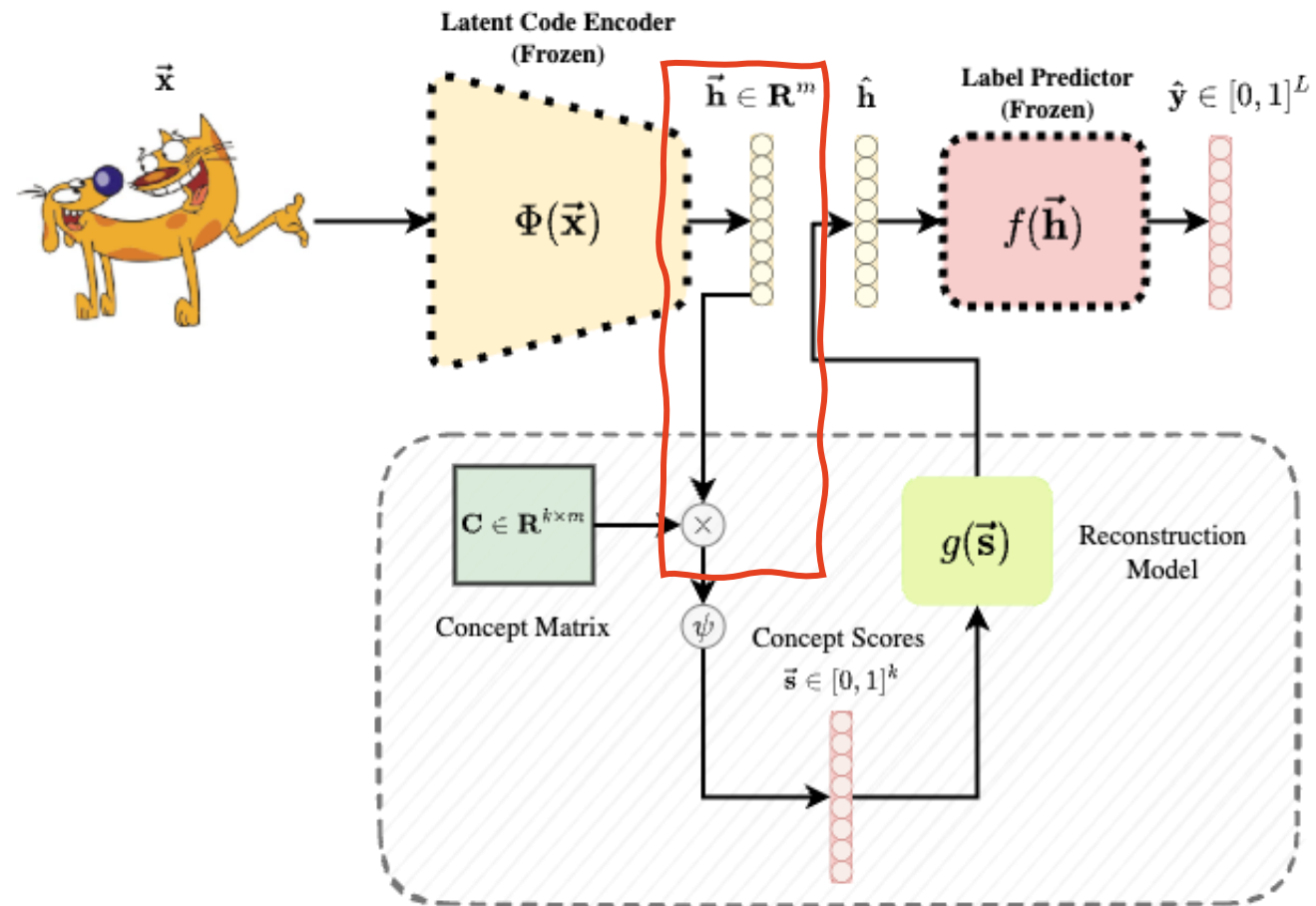


COMPLETENESS-AWARE CONCEPT EXTRACTION



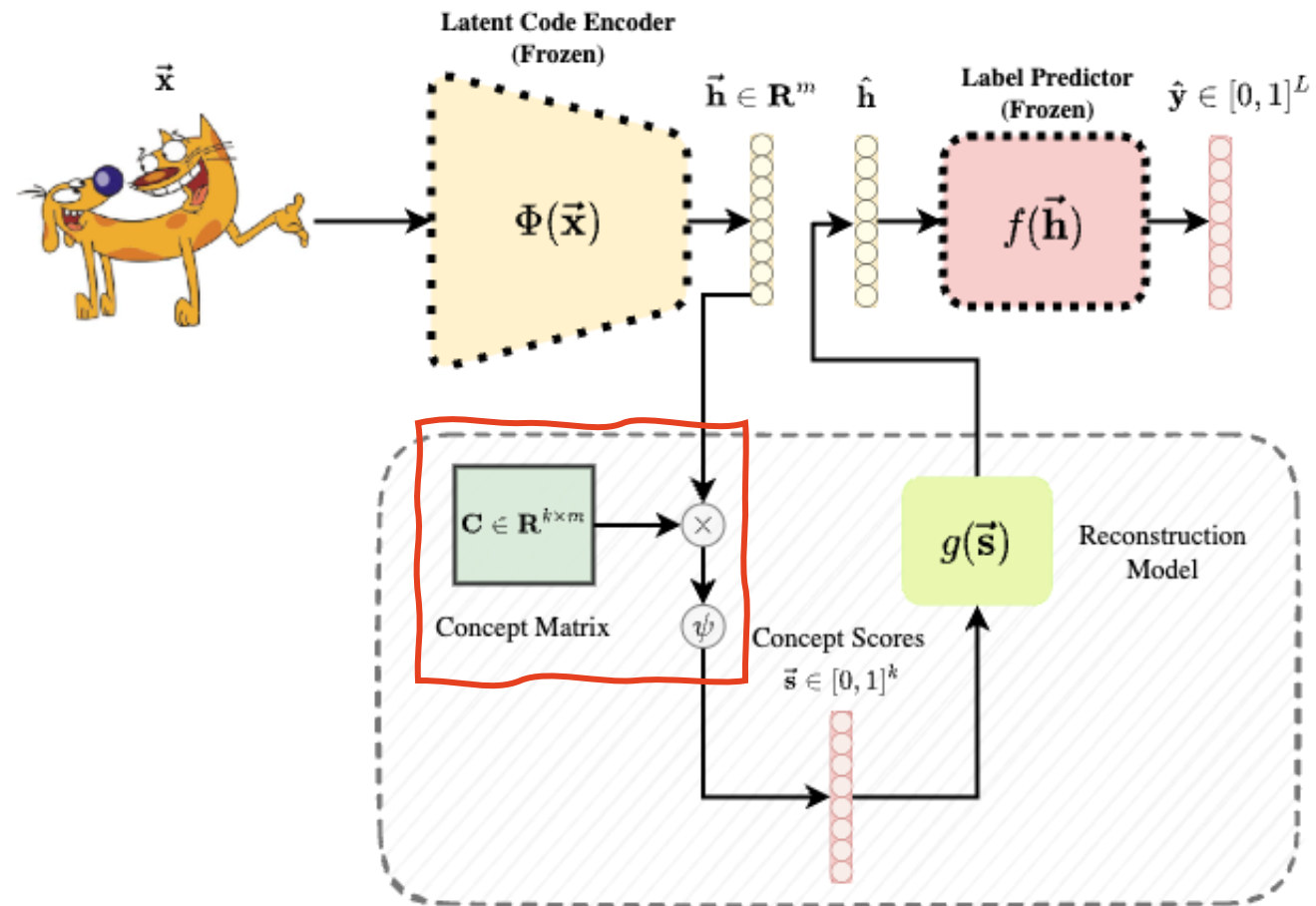
This is the concept matrix we want to learn (originally, we randomly initialise it)

COMPLETENESS-AWARE CONCEPT EXTRACTION



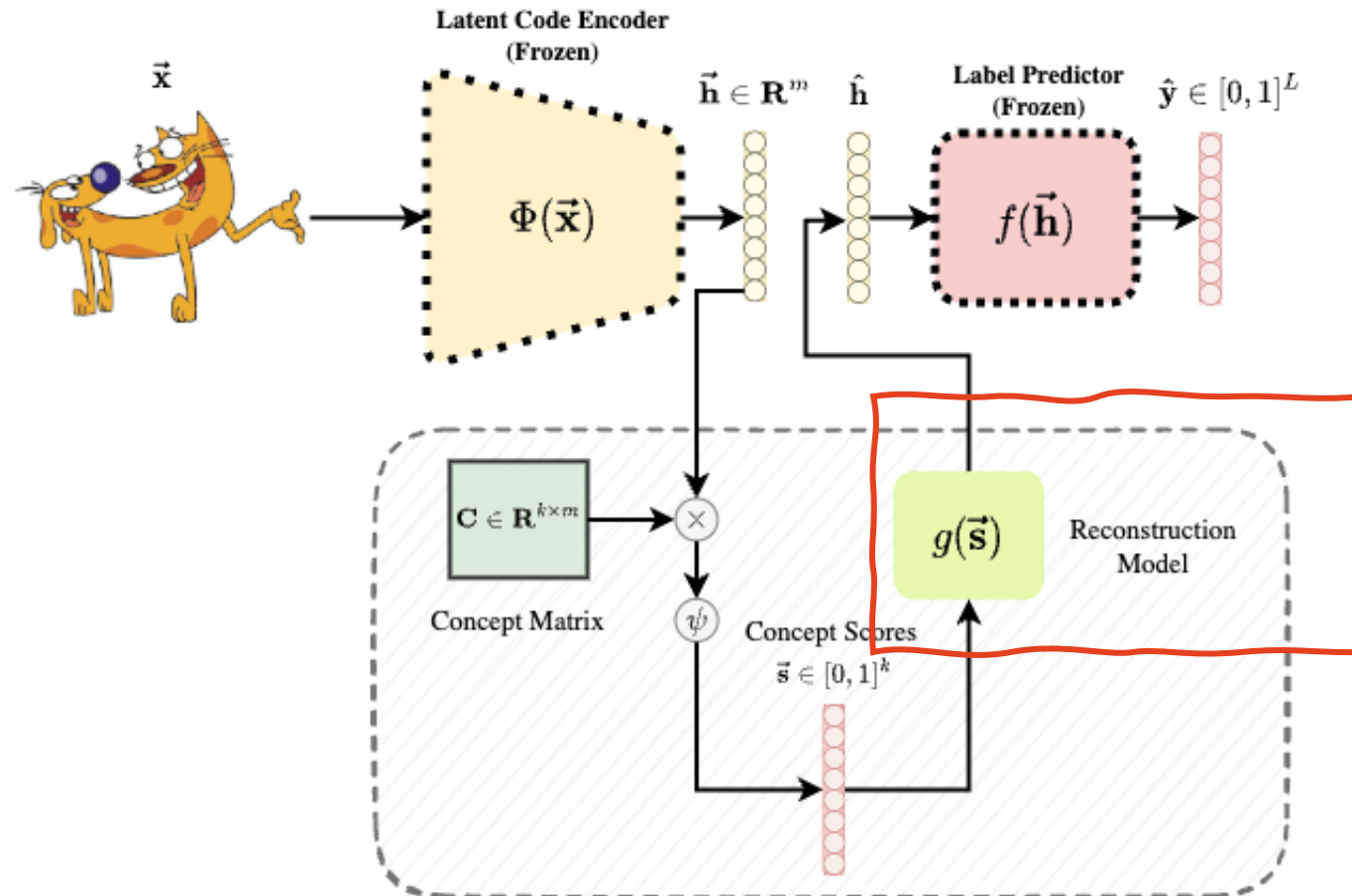
Step 1: project the DNN's hidden layer into the concept space

COMPLETENESS-AWARE CONCEPT EXTRACTION



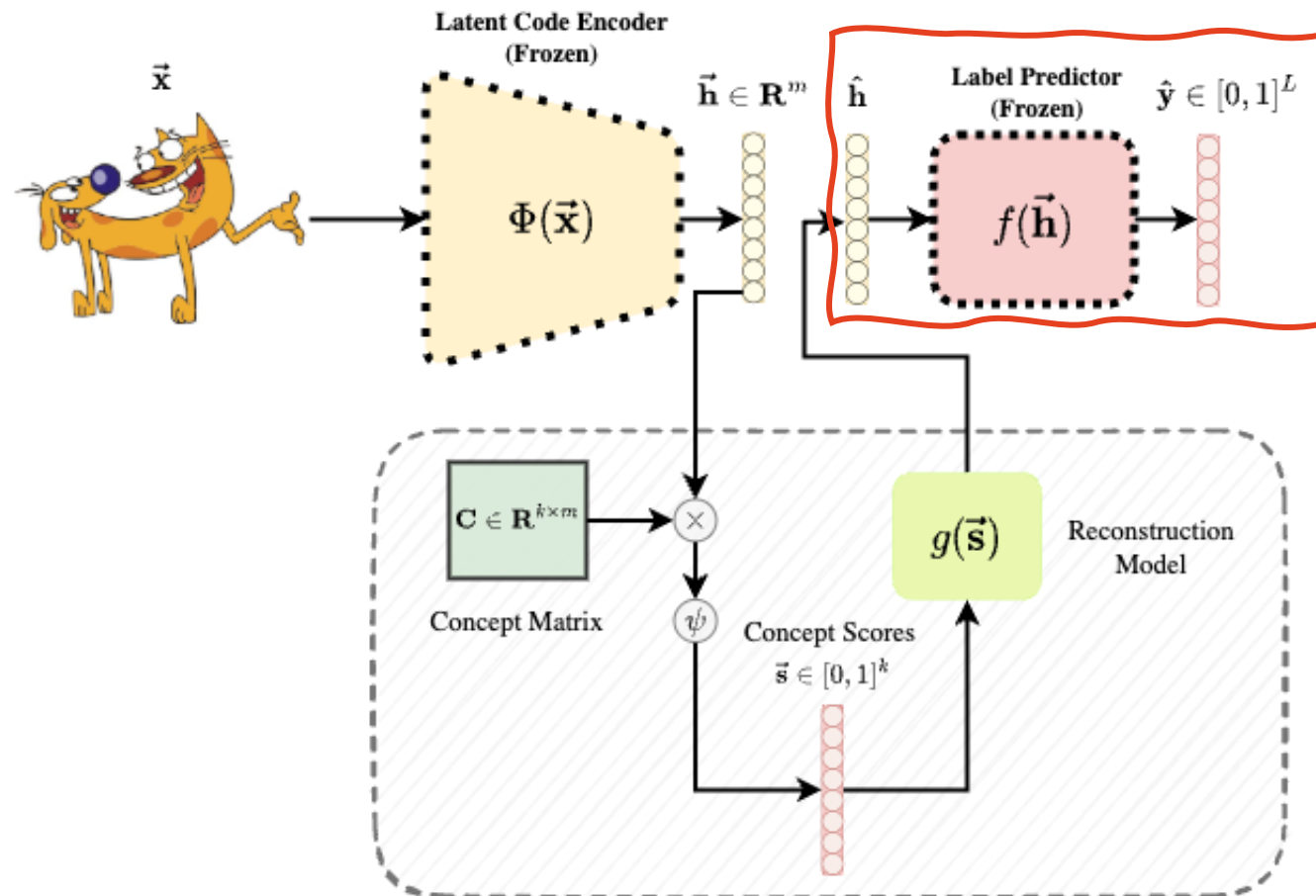
Step 2: compute a set of concept scores by thresholding and normalising the concept projection

COMPLETENESS-AWARE CONCEPT EXTRACTION



Step 3: pass the concepts scores to a learnable model $g(\vec{s}) = \hat{\vec{h}}$ that aims to reconstruct \vec{h} from \vec{s}

COMPLETENESS-AWARE CONCEPT EXTRACTION



Step 4: use \hat{h} as the reconstructed hidden layer and predict an output class using the rest of the DNN

CONCEPT DIVERSITY AND COHERENCE

CCE further encourages discovered concepts to be:

1. **Coherent**: similar samples should remain close in concept-space

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

CONCEPT DIVERSITY AND COHERENCE

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

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

CONCEPT CONTRIBUTIONS

We can assign each concept a **score** that fairly represents how much they contributed towards the completeness score

$$\mathbf{s}_i(\eta) = \sum_{S \subseteq C_s \setminus \mathbf{c}_i} \frac{(m - |S| - 1)! |S|!}{m!} [\eta(S \cup \{\mathbf{c}_i\}) - \eta(S)]$$

Does this look familiar?

CONCEPT CONTRIBUTIONS

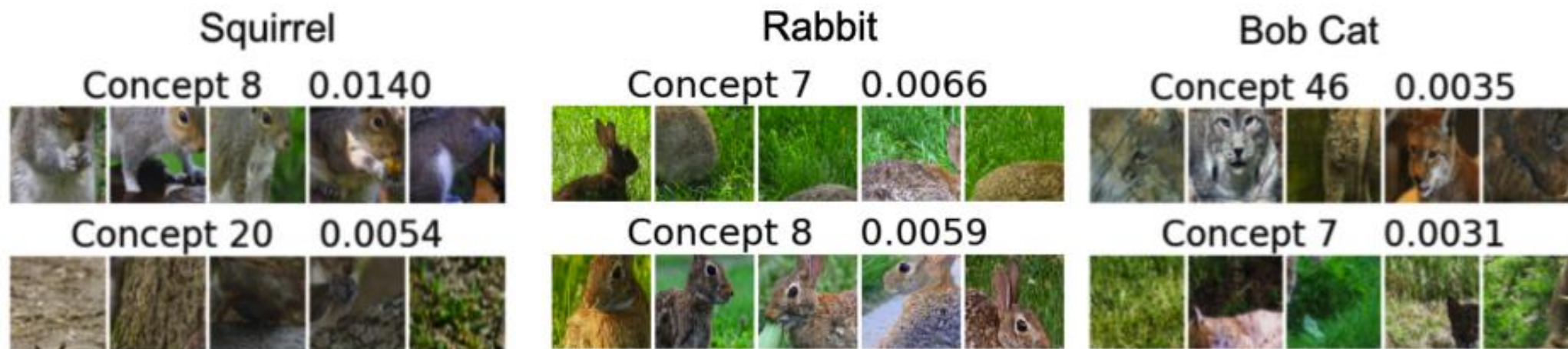
We can assign each concept a **score** that fairly represents how much they contributed towards the completeness score

$$\mathbf{s}_i(\eta) = \sum_{S \subseteq C_s \setminus \mathbf{c}_i} \frac{(m - |S| - 1)! |S|!}{m!} [\eta(S \cup \{\mathbf{c}_i\}) - \eta(S)]$$

These are Shapley Scores (i.e., **ConceptSHAP**)

CCE: APPLICATIONS

In vision, CCE can discover meaningful human-understandable concepts:



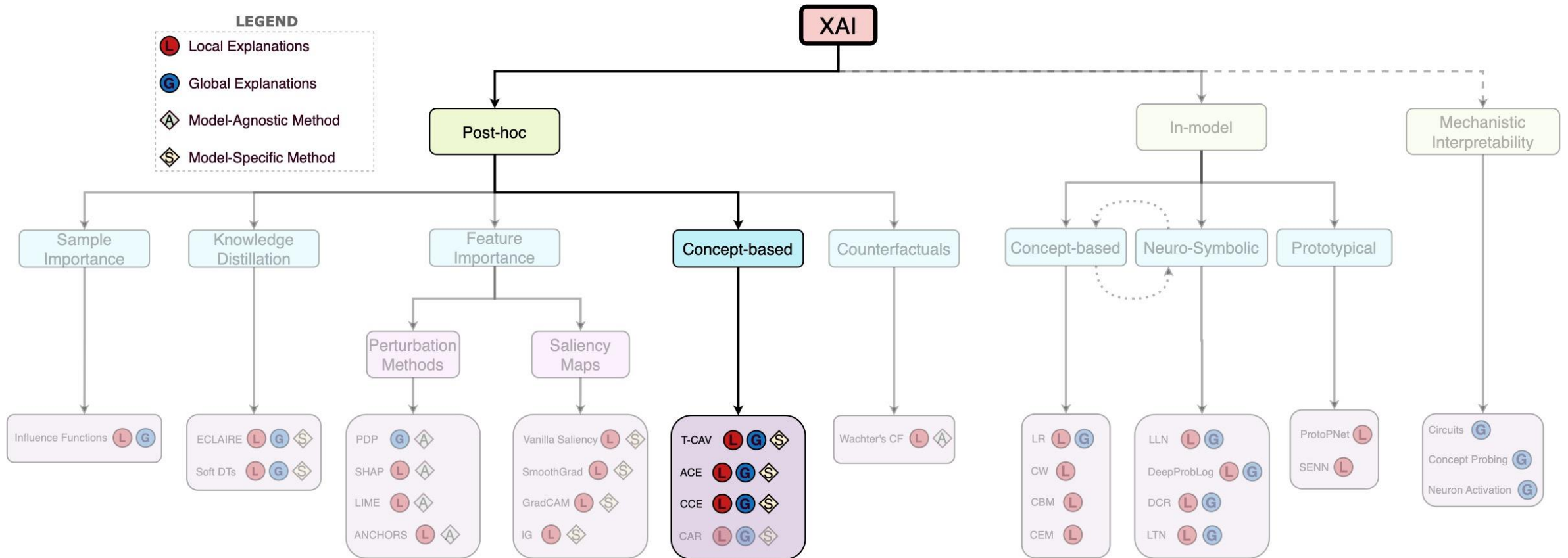
CCE: APPLICATIONS

And it can be applied to other data modalities like text!

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

TODAY IN A NUTSHELL



QUESTIONS?

