A GRADUATE LECTURE

# Concept Bottleneck Models

From black-box predictions to interpretable, editable concepts

Machine Learning Concepts

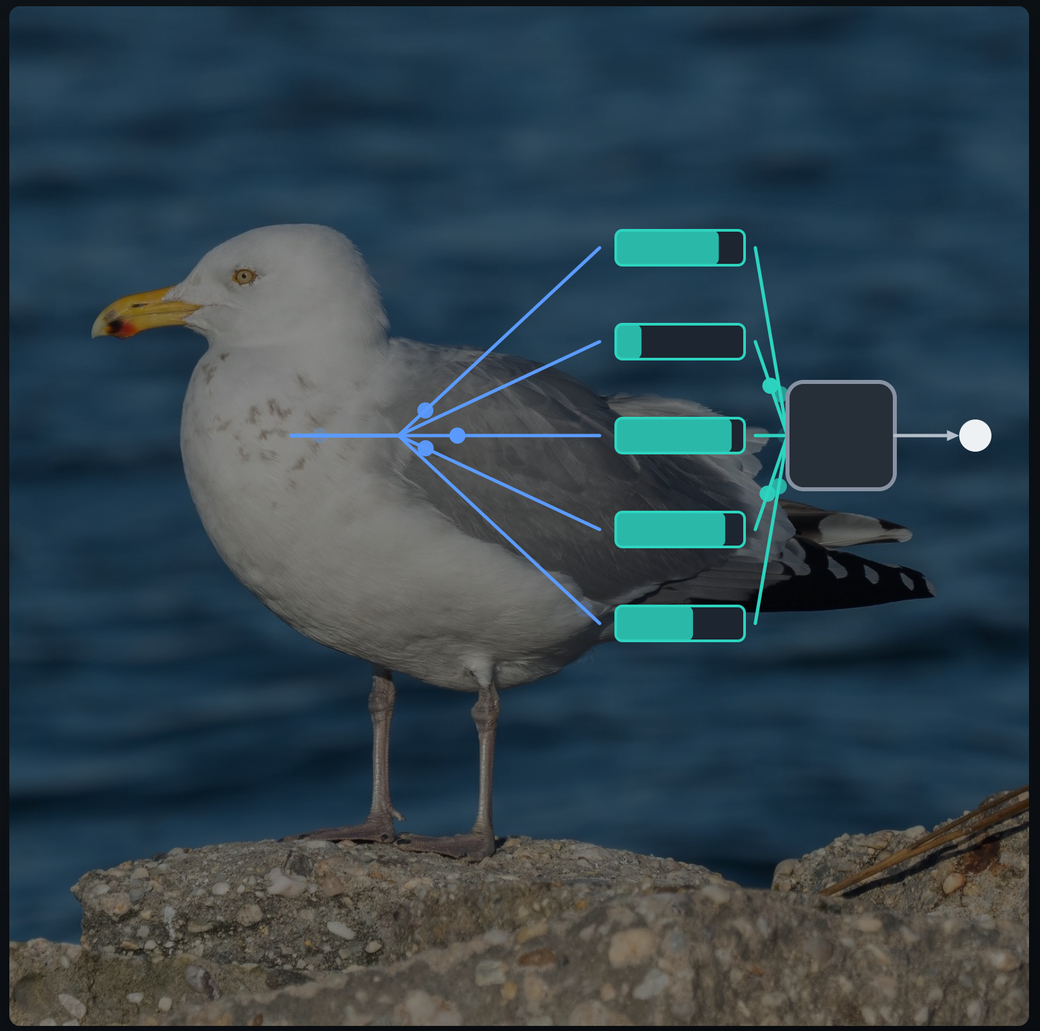


Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

PART 1

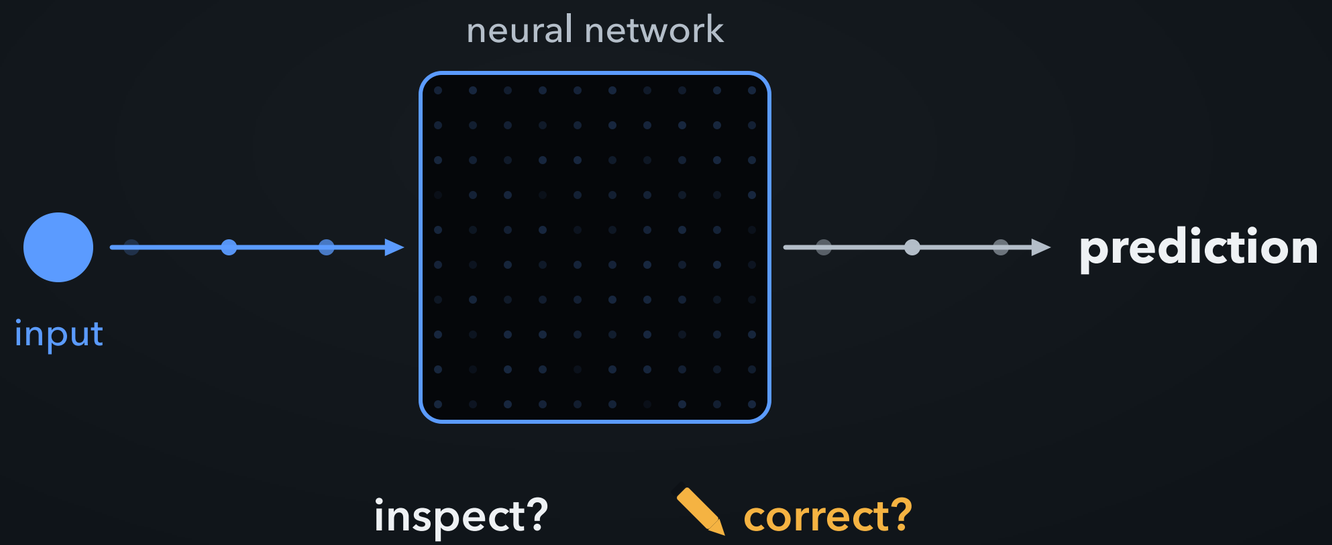
# Why make predictions interpretable?

What evidence did the model use?

PART 1

# The black box

**Can a person inspect, and correct, the evidence a network used?**



# The black box

## A real photograph and a real model, CLIP

Contrastive Language-Image Pre-training (CLIP)



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)




a real, trained model

COMPUTED · REAL MODEL · CLIP ViT-B/32, zero-shot over 6 species names

# The black box

## CLIP compares the photo with one sentence per species



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

"a photo of a **{species}**, a type of bird."

a photo of a **American herring gull**, a type of bird.

a photo of a **glaucous-winged gull**, a type of bird.

a photo of a **ring-billed gull**, a type of bird.

a photo of a **great black-backed gull**, a type of bird.

a photo of a **laughing gull**, a type of bird.

a photo of a **glaucous gull**, a type of bird.

the image is compared with each sentence

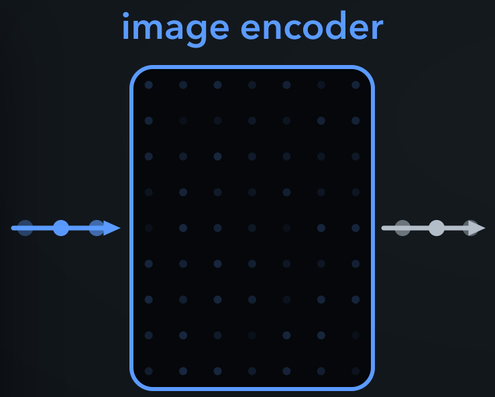
PART 1

# The black box

## The photo goes in, a score for every species comes out



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)







COMPUTED · REAL MODEL · CLIP ViT-B/32, softmax over 6 species names

PART 1

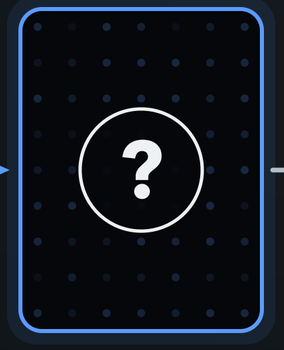
# The black box

## The answer is right. What made the model prefer it?



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

image encoder







COMPUTED · REAL MODEL · CLIP ViT-B/32, softmax over 6 species names

# The black box

## Occlusion: grey out one patch, measure the score drop

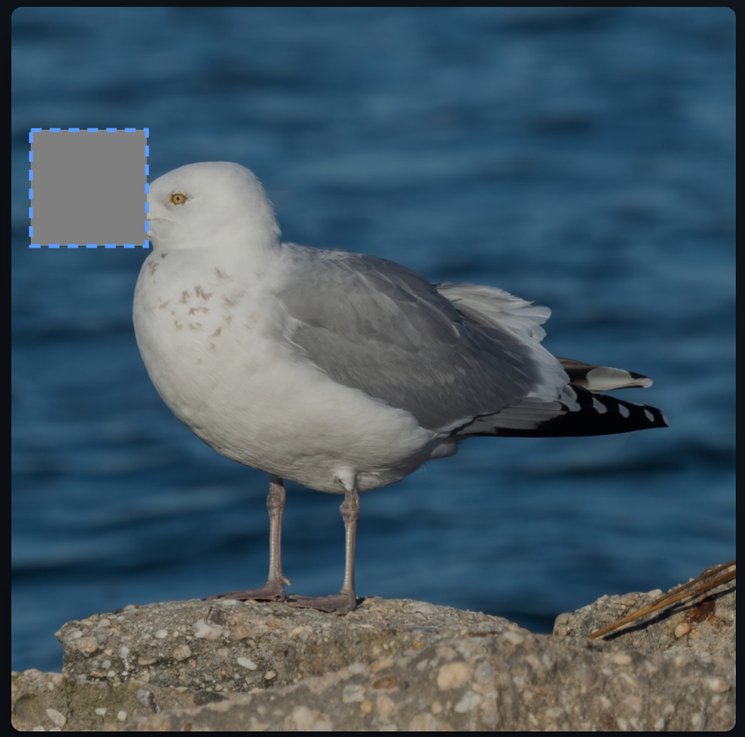


Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

grey out one patch, run the model again

drop in the **herring gull** log-probability

0.92

COMPUTED · REAL MODEL · CLIP, one grey patch at each of 14 x 14 positions

# The black box

## The largest drops sit on the bill and the head




Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

brighter: a larger drop in the **score**

`COMPUTED · REAL MODEL · occlusion sensitivity, 14 x 14 grey patches`

# The black box

## Which property did the model recognise there?

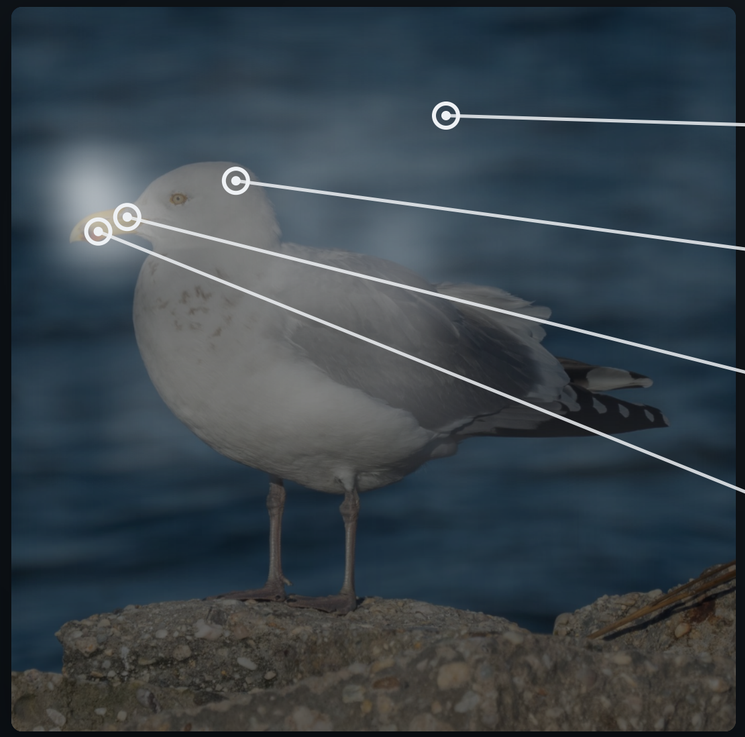



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

# The black box

## The map shows where the score is sensitive, not which property



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

- ✓ where the **score** is sensitive
- ✗ which property the model detected
- ✗ what a person could correct

# The black box

## What we want: evidence stated as named properties

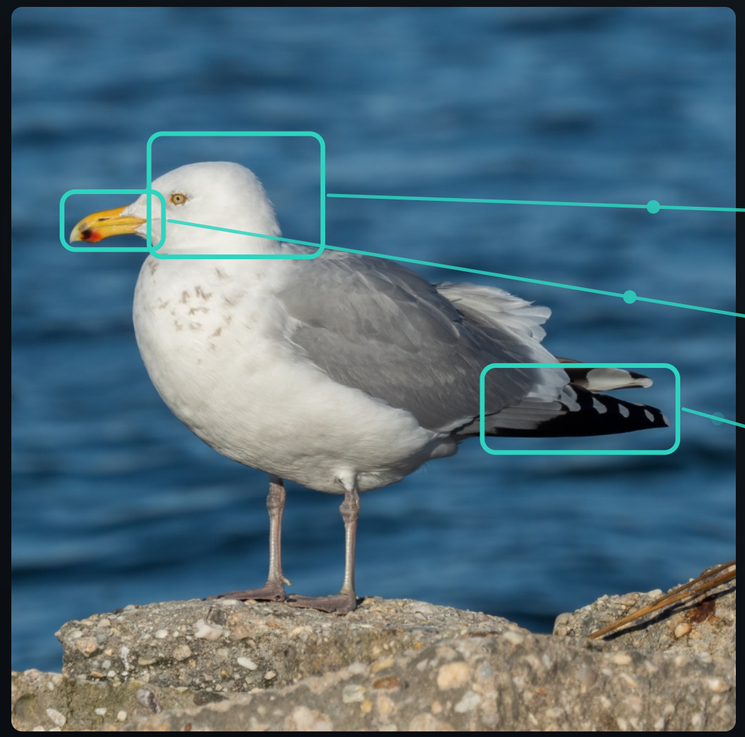


Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

**named properties**

- head colour
- bill shape
- wingtip colour

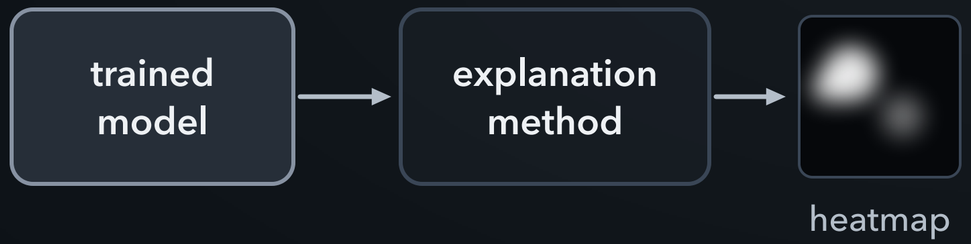
each claim can be checked

# Post-hoc or interpretable by design?

## Two routes: explain a model afterwards, or build one you can read

### post-hoc

explain it afterwards



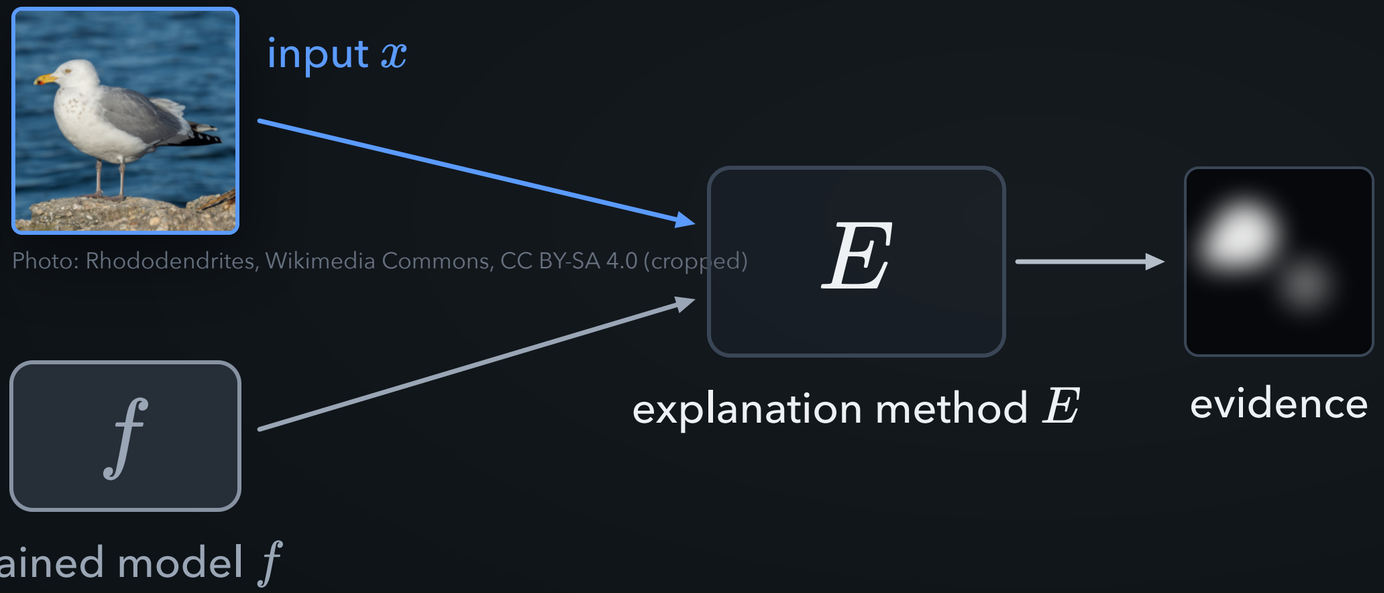

### ante-hoc

interpretable by design



# Post-hoc or interpretable by design?

## Post-hoc: a method explains a trained model on one input




$$\text{evidence} = E(f, x)$$

# Post-hoc or interpretable by design?

## Saliency: nudge one pixel, watch one class score

input photo



Photo: Yathin S Krishnappa, Wikimedia Commons, CC BY-SA 3.0; centre crop

red value: +0.001

**ResNet-50**
`trained, weights fixed`

Residual Network with 50 layers

chosen score: **zebra**

before softmax

**+0.00014**

tiny change in the score

sensitivity = size of **score change** ÷ size of pixel change

$$S_i^{(k)} = \left| \frac{\partial s_k(x)}{\partial x_i} \right| \approx \left| \frac{\Delta s_k}{\Delta x_i} \right| \qquad \frac{0.00014}{0.001} \approx 0.14$$

`COMPUTED · REAL MODEL · real photo, fixed ImageNet model, one pixel channel changed`

# Post-hoc or interpretable by design?

## Same photo, same model, a different class to explain

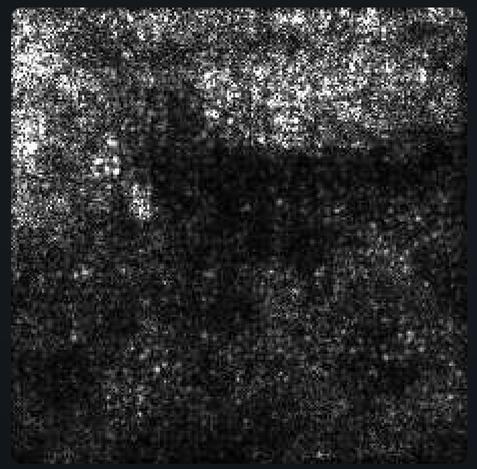
same input photo



same ResNet-50, fixed weights

**zebra score**

probability: 37.7% (top class)



saliency for the zebra score

**tiger score**

probability: 0.08% (not predicted)



saliency for the tiger score

Same brightness scale: brighter = more sensitive

**Low class probability does not mean zero pixel sensitivity**

COMPUTED · REAL MODEL · raw score gradients; max over RGB; shared scale, brightest 1% clipped

# Post-hoc or interpretable by design?

## Occlusion: grey out a patch and measure how far the score drops




Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

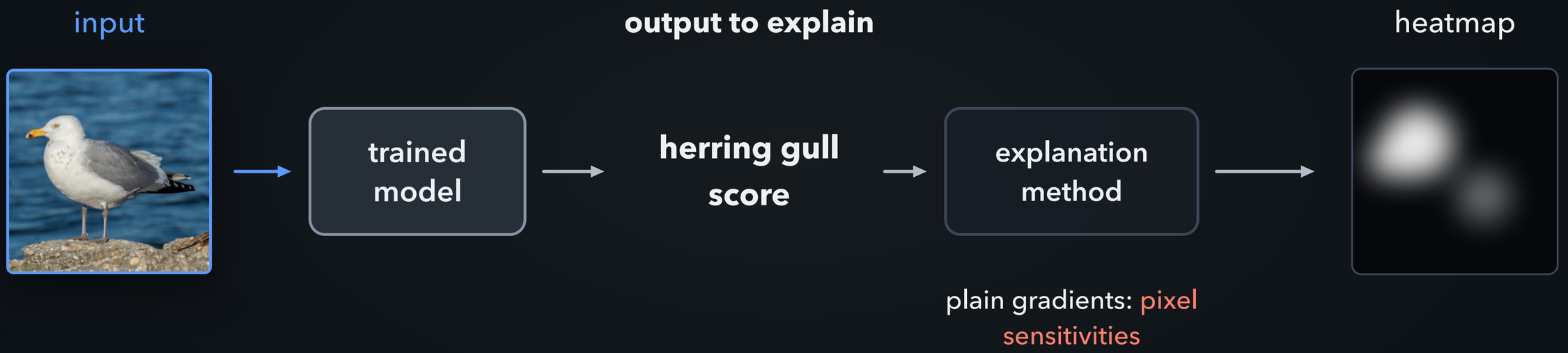
COMPUTED · REAL MODEL · CLIP occlusion map of the gull, Part 1

**score** on the photo minus **score** with patch p greyed

$$\Delta_p = \hat{y}_k(x) - \hat{y}_k(x \text{ with } p \text{ greyed})$$

# Post-hoc or interpretable by design?

## From an input photo and a class score to a heatmap




The method uses this photo, the model and the chosen **class**

**Brighter pixels: the chosen score is more sensitive there**

`SCHEMATIC · workflow and heatmap illustration, not computed for this photo`

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

# Post-hoc or interpretable by design?

## Does the heatmap depend on what the model learned?

Same photo, same **herring gull class**, same method

input photo



trained weights



replace learned weights

**almost unchanged**

random weights



reported example: guided backpropagation

**Before trusting a heatmap, check that it depends on the learned weights**

SCHEMATIC · drawn maps illustrating a reported failure, Adebayo et al., NeurIPS 2018

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

# Post-hoc or interpretable by design?

## The setup: learn pneumonia from two hospital datasets

**TRAIN: use examples from both hospitals**

| | | |
|---|---|---|
| **Mount Sinai** | X-rays + pneumonia labels | 34.2% positive labels |
| **NIH** | X-rays + pneumonia labels | 1.2% positive labels |

NIH: National Institutes of Health Clinical Center

→ **learn a pneumonia predictor**

Labels come from radiology reports

positive = pneumonia reported

**TEST: predict for a new image**

one new X-ray → the trained predictor → **pneumonia score**

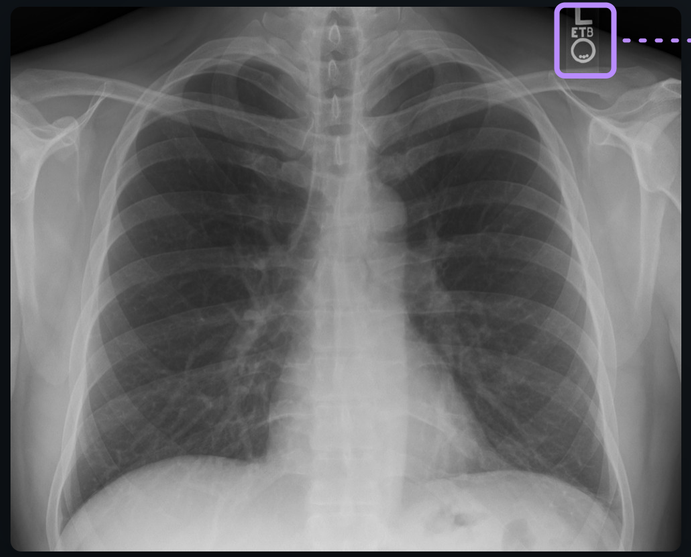
label hidden

**Risk: the hospital is already a clue to the label**

REPORTED · dataset label rates; Zech et al., PLOS Medicine 2018, Table 1

# Post-hoc or interpretable by design?

## The network could tell which hospital an X-ray came from

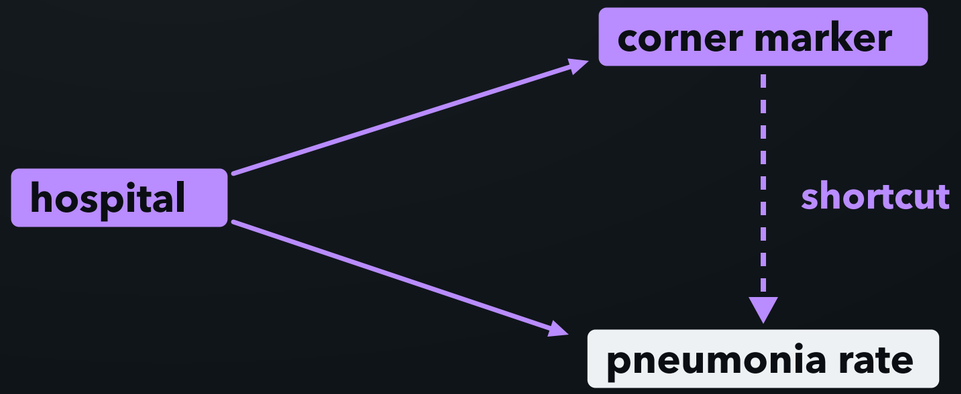


Chest X-ray: Stillwaterising, Wikimedia Commons, CC0

REPORTED · Zech et al., PLOS Medicine 2018

**which hospital?**

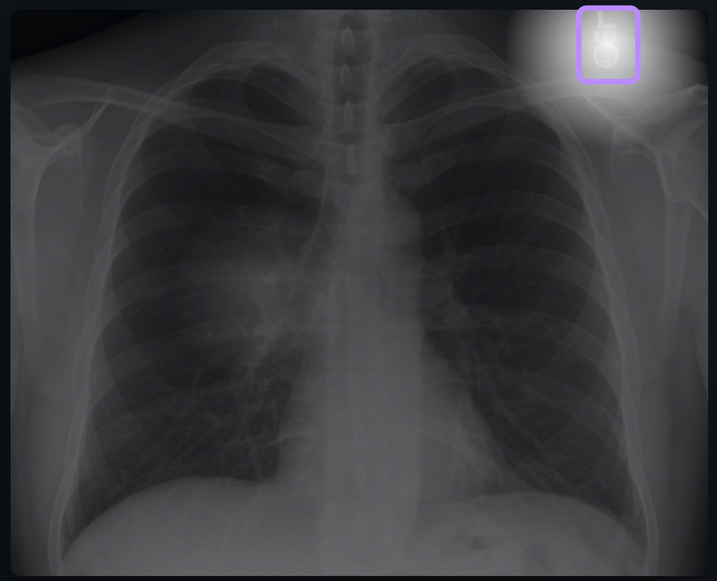
hospital named correctly for 99.95% of NIH X-rays




it can predict pneumonia without looking at the lungs

# Post-hoc or interpretable by design?

## An explanation can expose the shortcut, but the fix is new data



Chest X-ray: Stillwaterising, Wikimedia Commons, CC0

REPORTED · Zech et al., PLOS Medicine 2018

✓ exposes the shortcut

✗ cannot remove it

pneumonia AUC

| | |
|---|---|
| same hospitals | 0.931 |
| a new hospital | 0.815 |

fix: **new data** and retraining

# Post-hoc or interpretable by design?

## Limit 3: a heatmap says where, not what

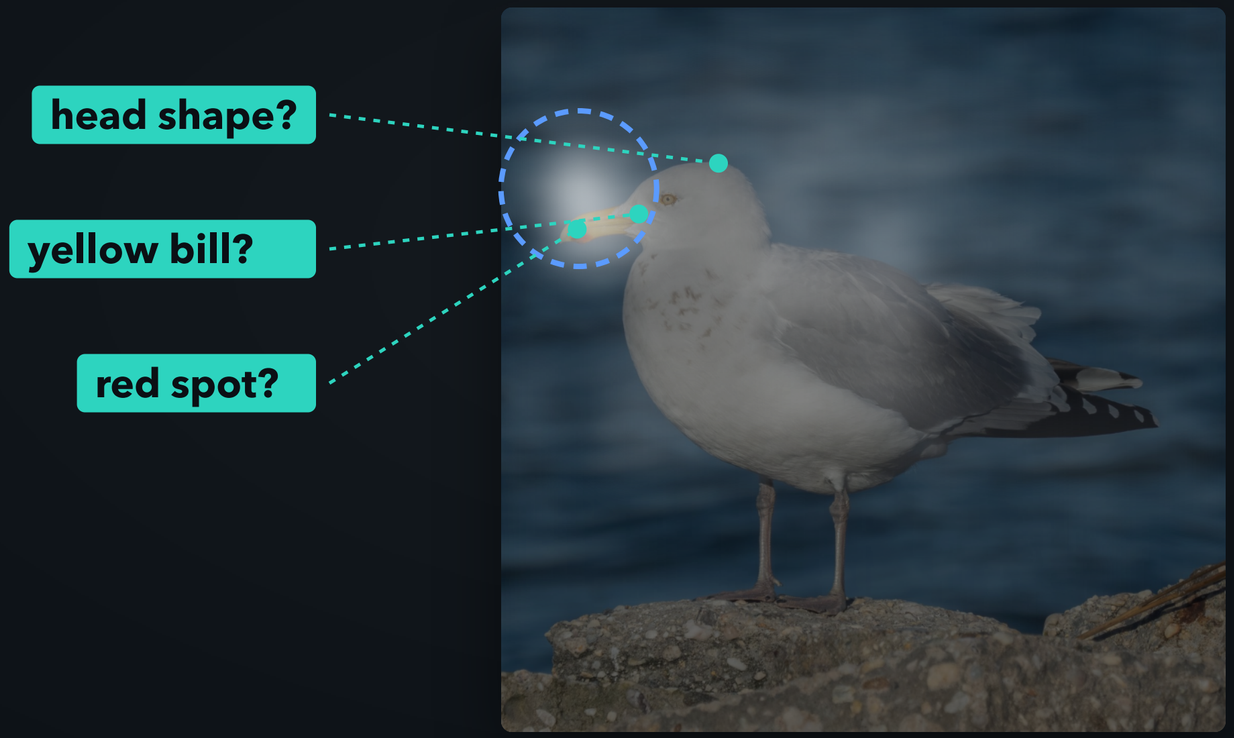



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

COMPUTED · REAL MODEL · CLIP occlusion map of the gull, Part 1

where: the score was sensitive here

what the model saw there: not said

# Post-hoc or interpretable by design?

## Experts speak in findings, and the model has no knob for them

**findings**

- narrowed joint space
- bone spur
- black wingtips



expert

black-box model






no named part to correct

next: a concept bottleneck adds that knob (Part 2)

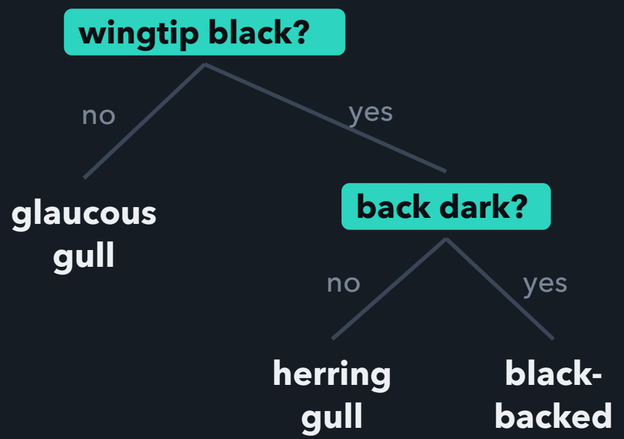
# Post-hoc or interpretable by design?

## Ante-hoc: the prediction is computed from quantities a person can read

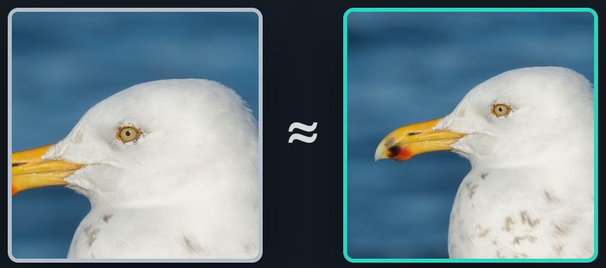
### sparse linear model

$$\hat{y} = 2.1\,\text{wingtip black} - 1.3\,\text{bill hooked} + \dots$$

### short decision tree




### prototype network



this part ≈ looks like that

Chen et al. 2019

ILLUSTRATIVE VALUES · hand-written examples

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

# Post-hoc or interpretable by design?

## The case for models that are interpretable by design

> For high-stakes decisions, prefer models people can read over black boxes explained afterwards.
>
> Rudin, Nature Machine Intelligence 2019 (paraphrase)

# Post-hoc or interpretable by design?

**A concept bottleneck predicts concepts first, then the answer**

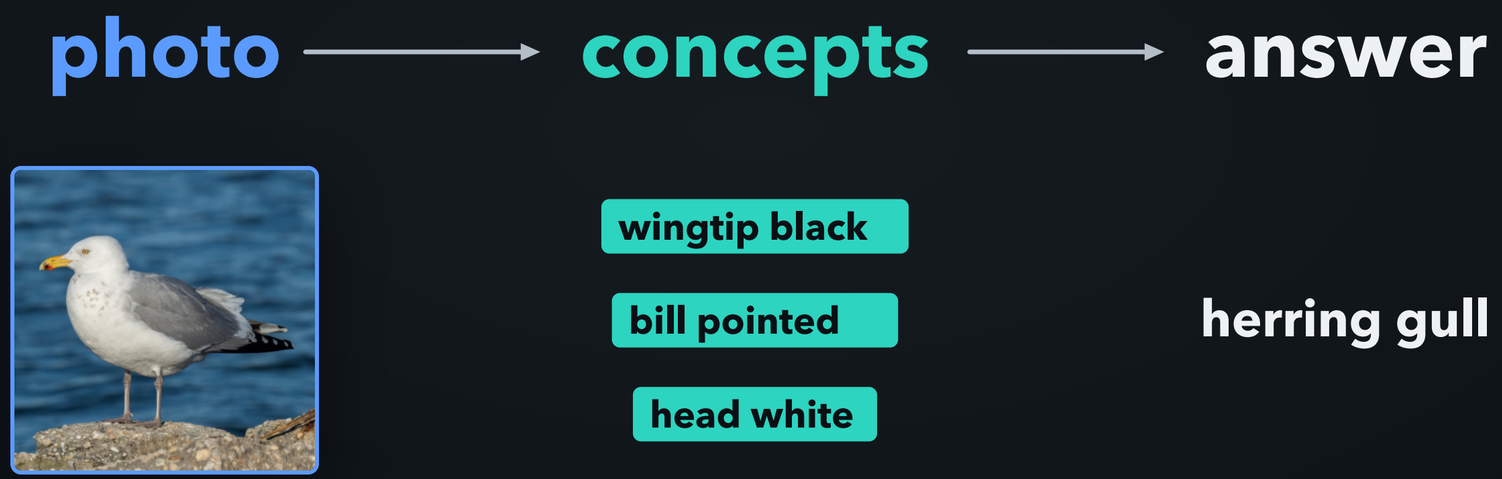
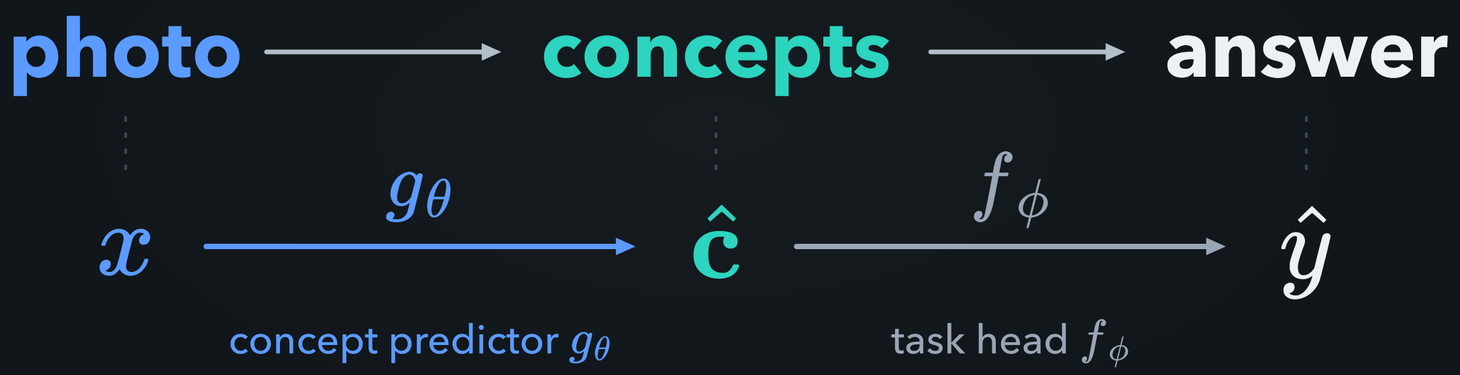



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

the **answer** is computed from the concepts alone

# Post-hoc or interpretable by design?

## The same path in symbols




$\hat{\mathbf{c}}$: one predicted value per concept

PART 2

# The concept bottleneck

Make the prediction pass through named concepts

# Introducing the concept bottleneck

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Here is that idea on a real photograph



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)




a layer of named variables

# Introducing the concept bottleneck

$x$ $g_\theta$ $\hat{c}$ $f_\phi$ $\hat{y}$

## A concept predictor fills the layer, a task head reads it

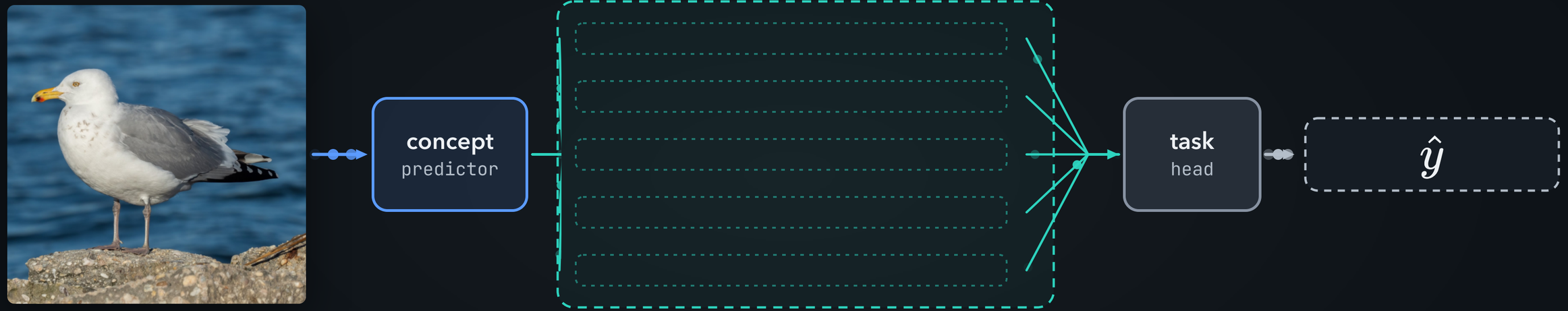



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

$$x \xrightarrow{g_\theta} \hat{\mathbf{c}} \xrightarrow{f_\phi} \hat{y}$$

# Introducing the concept bottleneck



## The concept vector has five named entries



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

$g_\theta$

$$\hat{\mathbf{c}} = \begin{bmatrix} 0.80 \\ 0.20 \\ 0.90 \\ 0.85 \\ 0.60 \end{bmatrix}$$

- 0.80 — wingtip grey
- 0.20 — wingtip black
- 0.90 — bill pointed
- 0.85 — head white
- 0.60 — tail rounded

each entry: a value between 0 and 1

ILLUSTRATIVE VALUES · hand-written concept values, for teaching

# Introducing the concept bottleneck

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The rule: the task head reads only these five numbers



concept predictor

| | |
|---|---|
| wingtip grey | 0.80 |
| wingtip black | 0.20 |
| bill pointed | 0.90 |
| head white | 0.85 |
| tail rounded | 0.60 |

task head

$\hat{y}$

the bottleneck: the only input to the task head

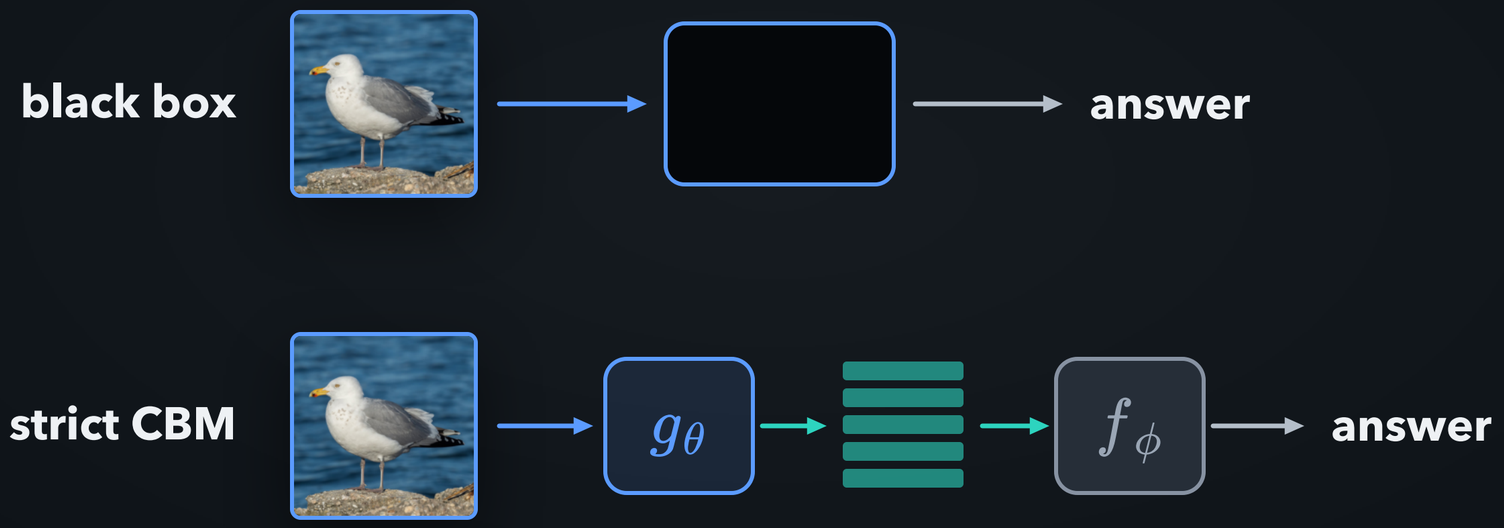
no other path from the image

ILLUSTRATIVE VALUES · hand-written concept values and head, for teaching

# Introducing the concept bottleneck



## A strict CBM predicts the concepts first, then the answer from them alone




the **answer** from the concepts alone

# Introducing the concept bottleneck




## The values and the head on these slides are written by hand

| | weight toward American herring gull |
|---|---|
| wingtip grey | -2.0 |
| wingtip black | +2.0 |
| bill pointed | +0.4 |
| head white | -0.2 |
| tail rounded | +0.2 |

the head adds up weight × value, then turns the total into two **shares**

ILLUSTRATIVE VALUES · hand-written concept values and head, for teaching

# Introducing the concept bottleneck

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The model believes the wingtips are grey, so it says glaucous-winged gull



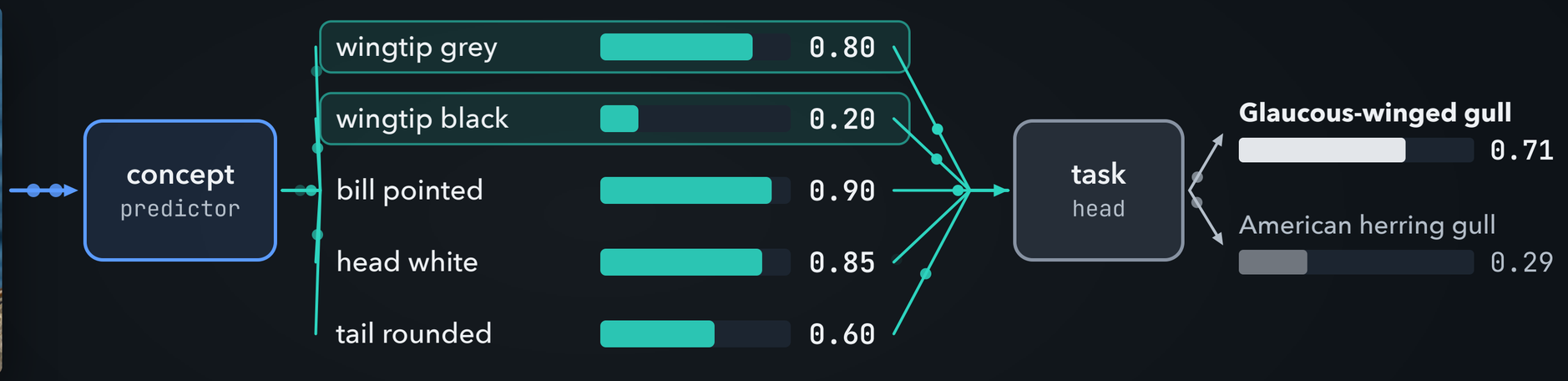



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

grey high, black low: the head favours **glaucous-winged gull**

ILLUSTRATIVE VALUES · hand-written concept values and head, for teaching

# Introducing the concept bottleneck

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The wingtips are black: an observer corrects the wing colour





the wingtips are black

concept predictor

| | |
|---|---|
| wingtip grey | 0.00 |
| wingtip black | 1.00 |
| bill pointed | 0.90 |
| head white | 0.85 |
| tail rounded | 0.60 |

task head

**Glaucous-winged gull** 0.71

American herring gull 0.29

wing colour: corrected as one group

ILLUSTRATIVE VALUES · hand-written concept values and head, for teaching

# Introducing the concept bottleneck

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The same task head recomputes, and herring gull now leads



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

unchanged

concept predictor

- wingtip grey 0.00
- wingtip black 1.00
- bill pointed 0.90
- head white 0.85
- tail rounded 0.60

task head

unchanged

- Glaucous-winged gull 0.09
- **American herring gull** 0.91

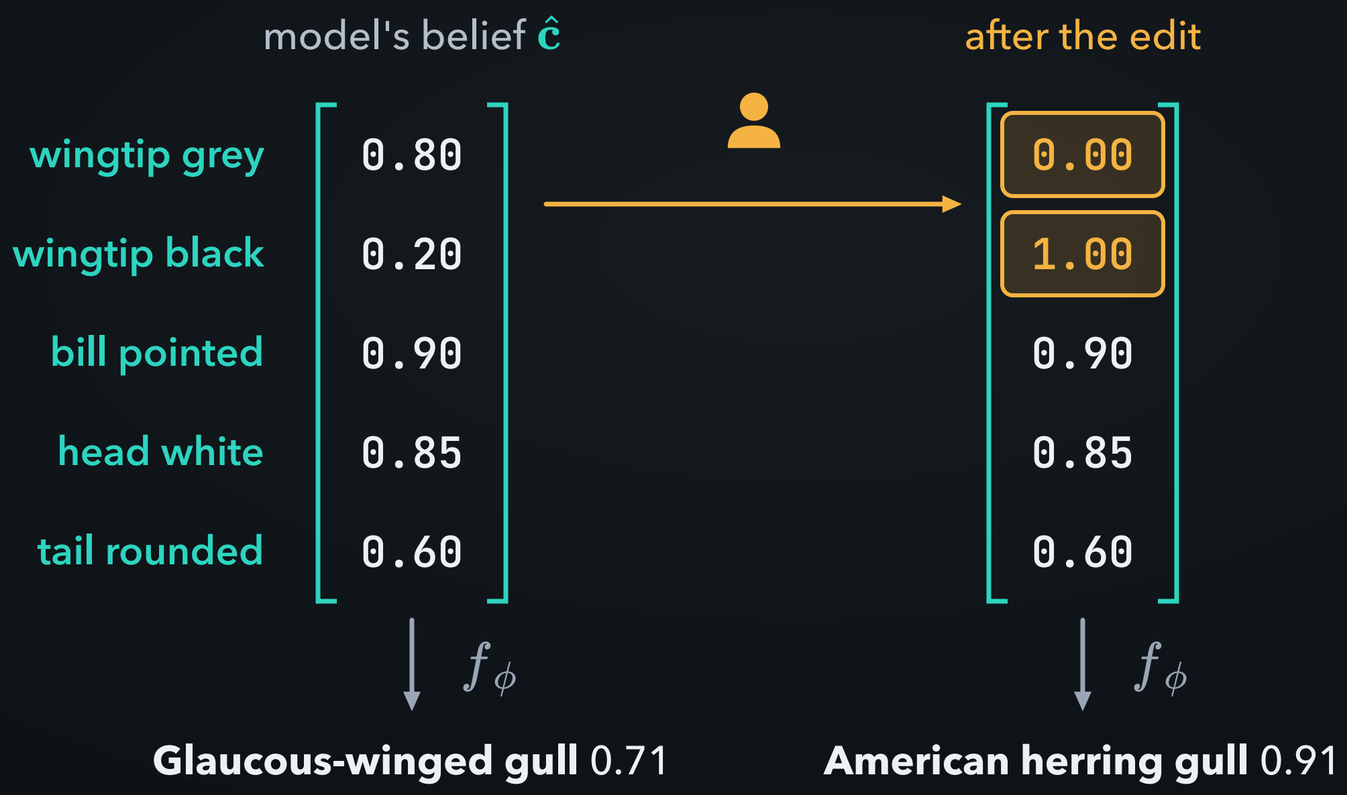
same image, same head, edited concepts: a new **answer**

ILLUSTRATIVE VALUES · hand-written concept values and head, for teaching

# Introducing the concept bottleneck




## An edited belief, a new prediction




ILLUSTRATIVE VALUES · hand-written concept values and head, for teaching

# Introducing the concept bottleneck

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Multi-task learning puts a concept head beside the class head

### multi-task learning



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

shared encoder

concept head

- wingtip grey 0.80
- wingtip black 0.20
- bill pointed 0.90
- head white 0.85
- tail rounded 0.60

class head

- **Glaucous-winged gull**
- American herring gull

# Introducing the concept bottleneck

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## There, an edit to the concepts changes nothing

**multi-task learning**



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

shared encoder

concept head

| | |
|---|---|
| wingtip grey | 0.00 |
| wingtip black | 1.00 |
| bill pointed | 0.90 |
| head white | 0.85 |
| tail rounded | 0.60 |

class head

**Glaucous-winged gull**

American herring gull

no change

the class head never reads the concept head

# Introducing the concept bottleneck

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## In a CBM, the concepts lie on the path to the prediction



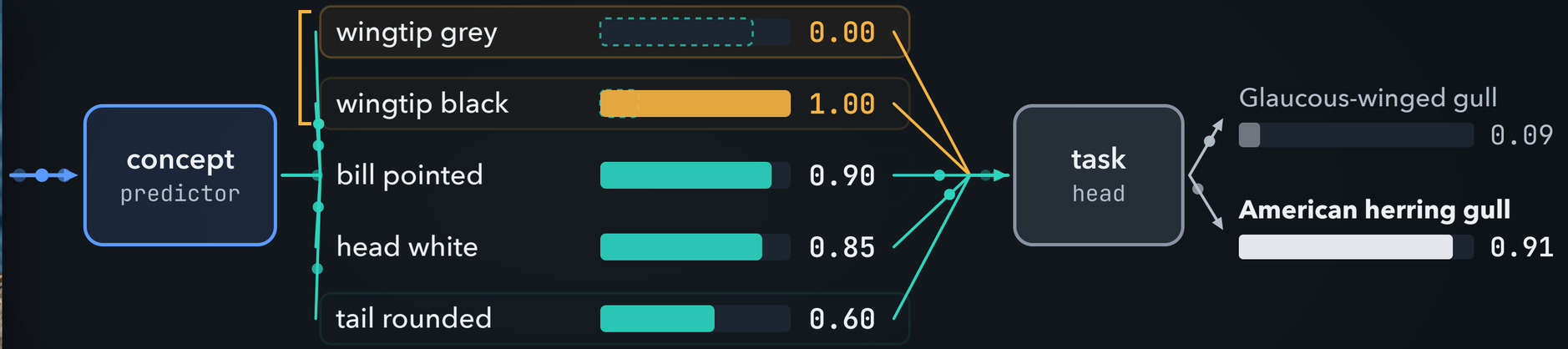



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

every **prediction** passes through the concepts

ILLUSTRATIVE VALUES · hand-written concept values and head, for teaching

# The CUB benchmark

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

**The standard benchmark for concept bottlenecks is a bird dataset**



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

our photo, not from CUB

## Caltech-UCSD Birds-200-2011

# CUB

Wah et al., Caltech 2011

# The CUB benchmark




## Many photographs of many similar species

**11,788**

photographs

**200**

species

8 of the species are gulls

California · Glaucous-winged · Heermann · Herring · Ivory · Ring-billed · Slaty-backed · Western

DATASET · CUB-200-2011, counted

# The CUB benchmark




## Every photograph is described by the same attributes

312 yes-or-no attributes, in 28 groups

| | | | |
|---|---|---|---|
| back colour | back pattern | belly colour | belly pattern |
| bill colour | bill length | **bill shape** | breast colour |
| breast pattern | crown colour | **eye colour** | forehead colour |
| head pattern | leg colour | nape colour | primary colour |
| shape | size | tail pattern | tail shape |
| throat colour | under tail colour | underparts colour | upper tail colour |
| upperparts colour | **wing colour** | **wing pattern** | wing shape |

# The CUB benchmark



## Each attribute is a yes-or-no question



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

| | | |
|---|---|---|
| Does it have a hooked bill? | yes → | 1 |
| Is its wing grey? | yes → | 1 |
| Is its wing yellow? | no → | 0 |

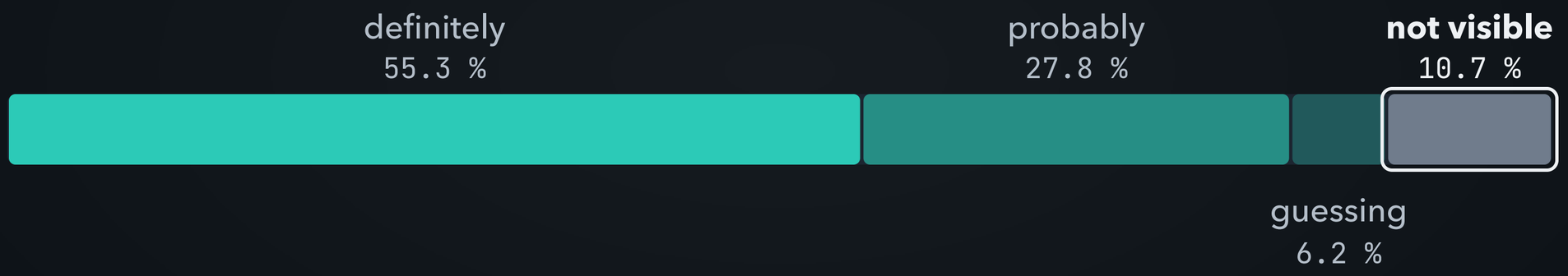
the answer to question $j$: yes is 1, no is 0

$$c_j \in \{0, 1\}$$

# The CUB benchmark



## Annotators also recorded how sure they were



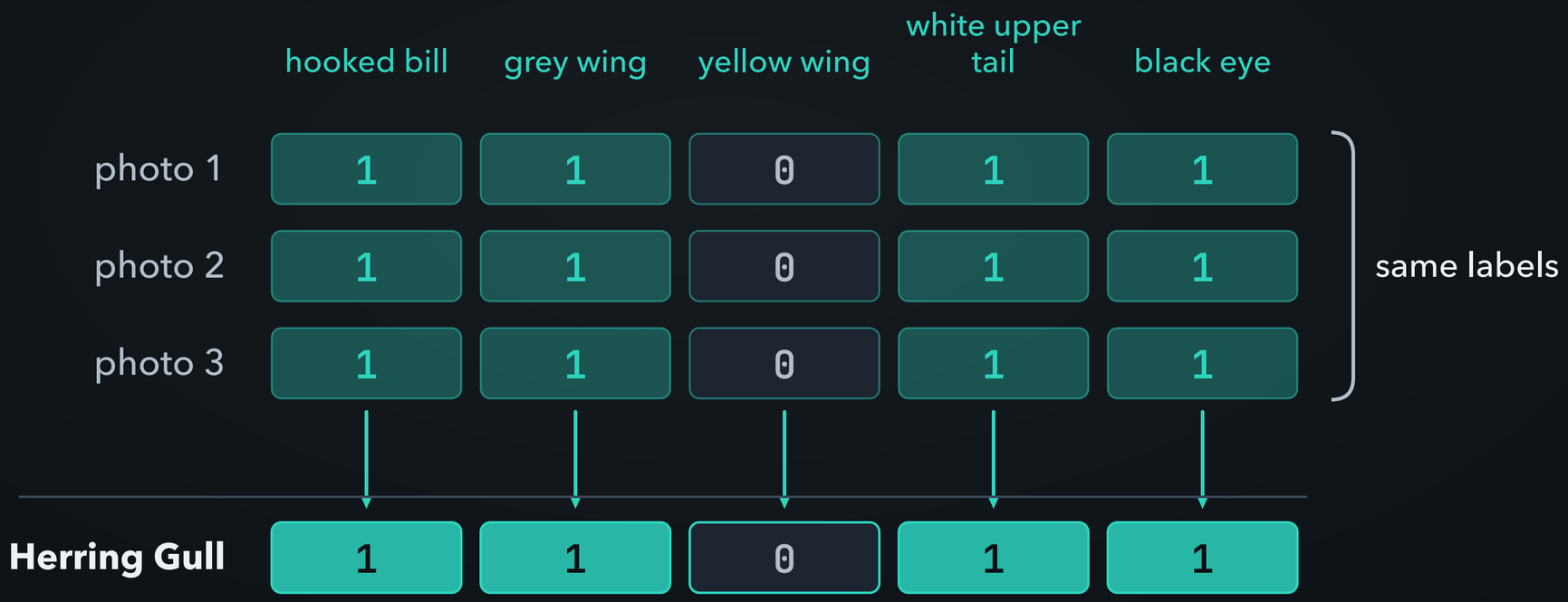

about one label in ten: not visible

DATASET · CUB annotations, counted

# The CUB benchmark

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Single photos are noisy: take a vote within each species



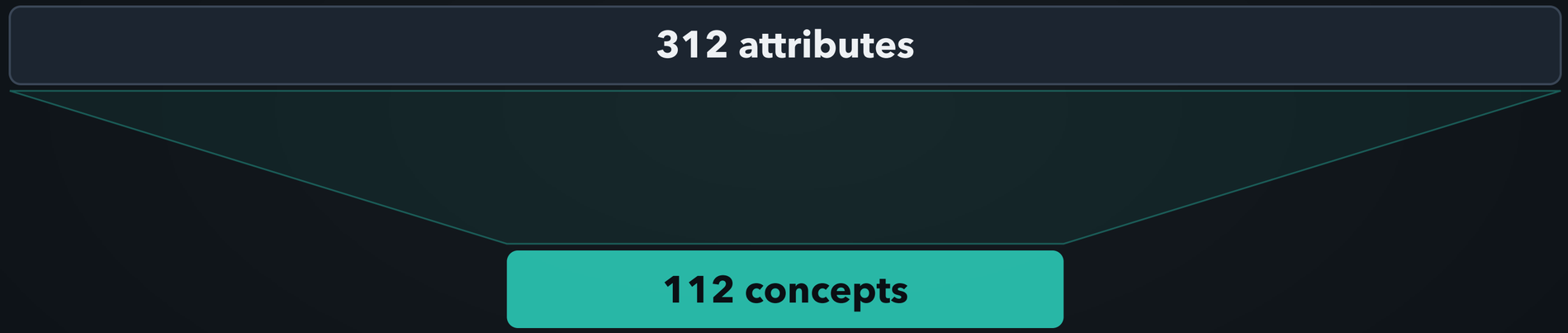

Koh et al., 2020

ILLUSTRATIVE VALUES · photo rows; the voted row is Herring Gull's in CUB

# The CUB benchmark



## Keep only the attributes that many species share




present in at least 10 species

DATASET · CUB, filtered as in Koh et al. 2020

# The CUB benchmark

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## One training record has three parts




Photo: Rhododendrites, CC BY-SA 4.0

DATASET · Herring Gull's labels in CUB, first 20 of 112; our photo

# The CUB benchmark




## The same record in symbols

concept labels, each 0 or 1

$$\mathbf{c} \in \{0,1\}^{112}$$

**species label**

$$y \in \{1, \ldots, 200\}$$

predicted concepts — concept predictor — one value per concept

$$\hat{\mathbf{c}} = g_\theta(x) \in [0,1]^{112}$$

PART 3

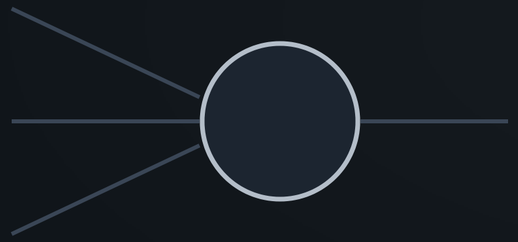
# What is a concept?

Choose what the model should describe

# What is a concept?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## A concept is a claim that a person can check



a name tag on a neuron

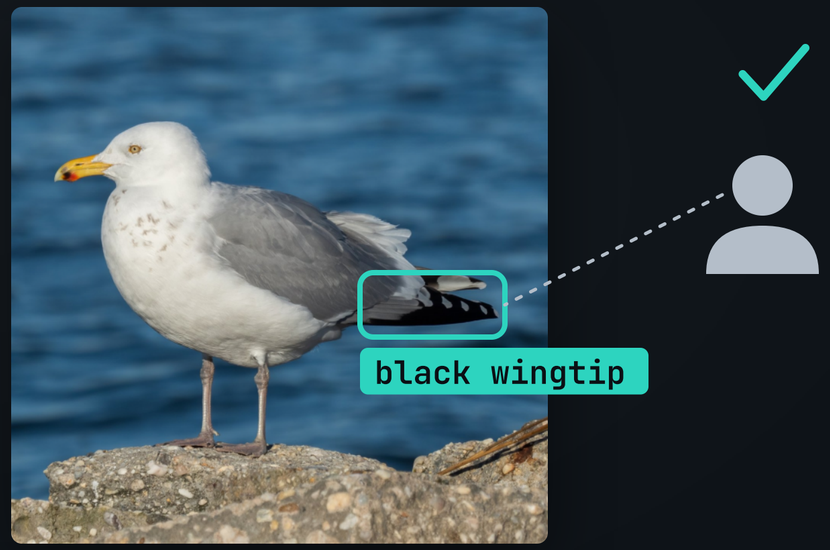



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

a claim a person checks in the input

# What is a concept?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Meaning: a definition that an expert can apply

1 **meaning** 2 grounding 3 representation



expert

`grey wingtip`

the tips of the folded flight feathers are **grey**, not black

# What is a concept?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Grounding: the evidence sits in one region of the photo

1 meaning  2 **grounding**  3 representation

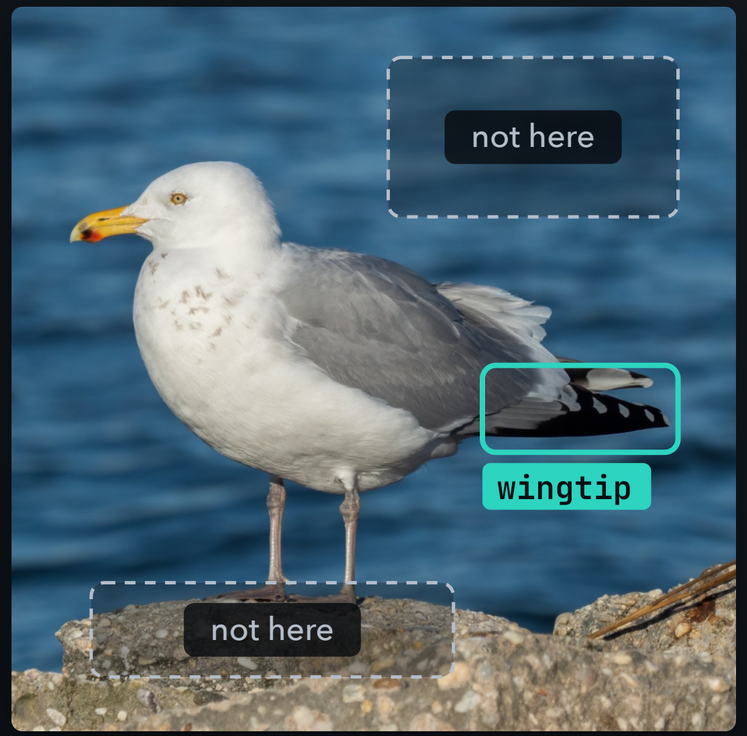



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

the wingtip colour estimate should depend on these feathers

not on the water or the rock

# What is a concept?




## Representation: the simplest concept is binary

1 meaning  2 grounding  3 **representation**






Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

the value of concept j: 0 or 1

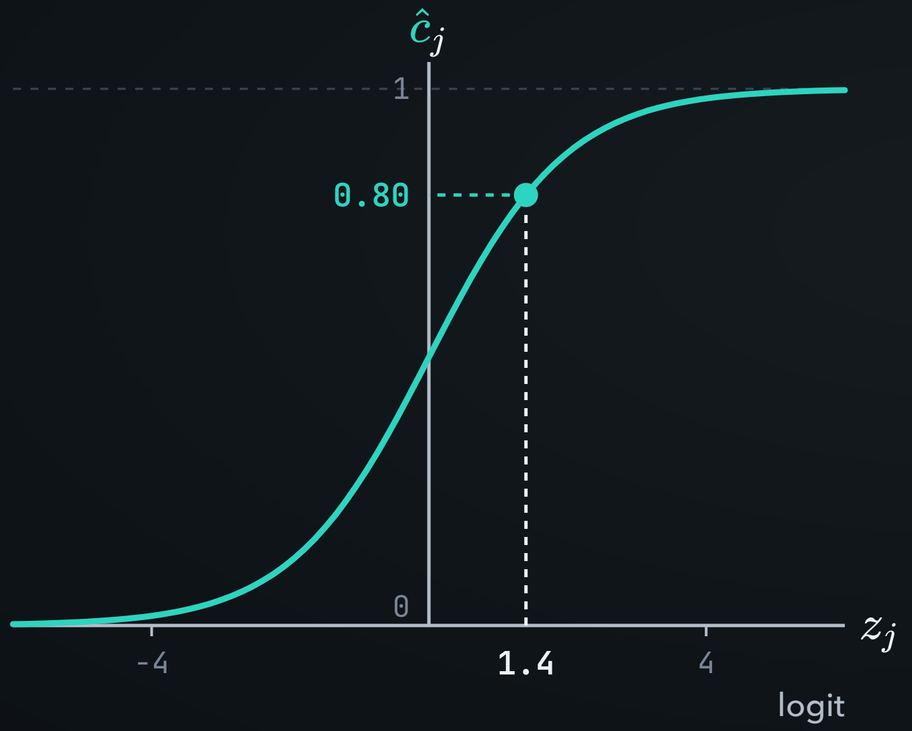
$$c_j \in \{0, 1\}$$

# What is a concept?



## A soft concept turns a logit into a value between 0 and 1

representation ● ● ○ ○ ○ ○




the soft score is the sigmoid of a logit

$$\hat{c}_j = \sigma(z_j) = \frac{1}{1 + e^{-z_j}}$$

example: $z_j = 1.4$ gives $\hat{c}_j = 0.80$

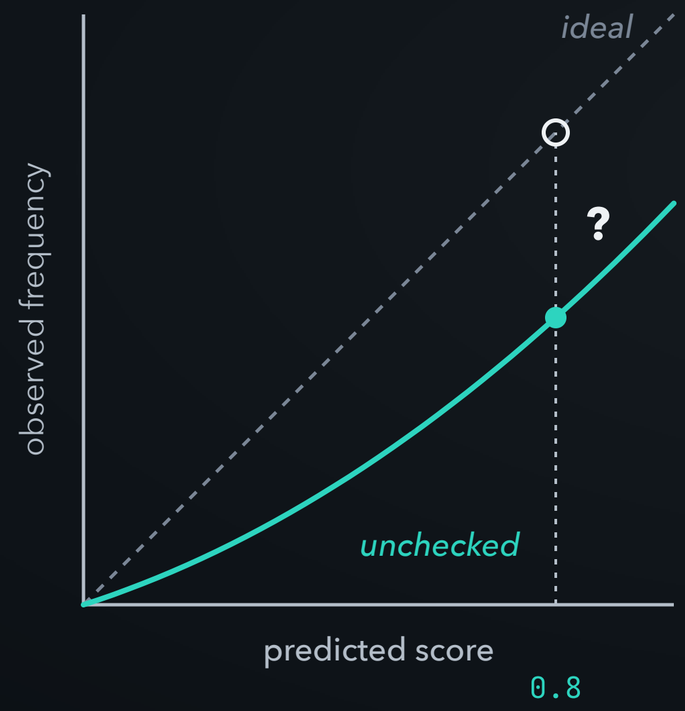
ILLUSTRATIVE VALUES · a worked logit

# What is a concept?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## A soft score is not automatically a probability

representation ●●○○○○




0.8 is a score, not automatically a probability

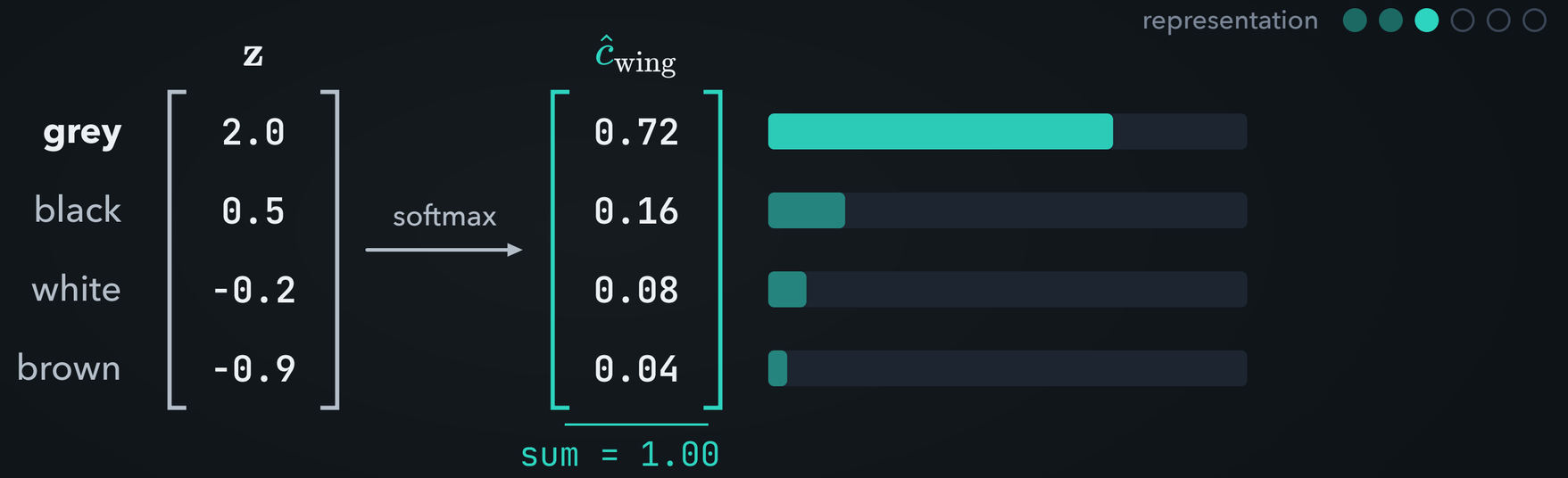
check on data: do 80 % of the cases scored 0.8 have grey wingtips?

SCHEMATIC · a sketch, no data

# What is a concept?



## Categorical: one value from a list, with scores that sum to 1




the wing colour scores are a softmax of the logits, and they sum to 1

$$\hat{c}_{\text{wing}} = \text{softmax}(\mathbf{z})$$

ILLUSTRATIVE VALUES · logits made up for the example

# What is a concept?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Continuous: a measurement, such as bill length

representation ●●●●○○

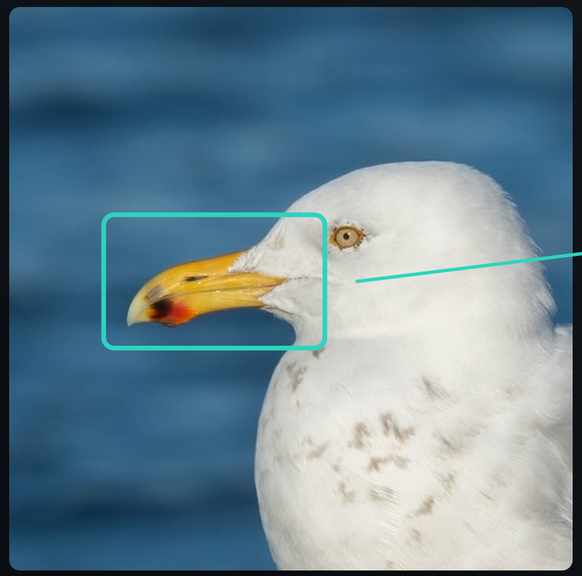
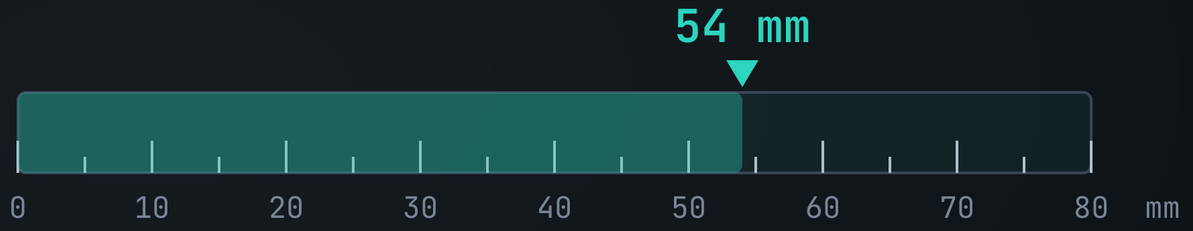


Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)




the bill length is a real number, in mm

$$\hat{c}_{\text{bill}} \in \mathbb{R}$$

predicted by regression

ILLUSTRATIVE VALUES · value

# What is a concept?



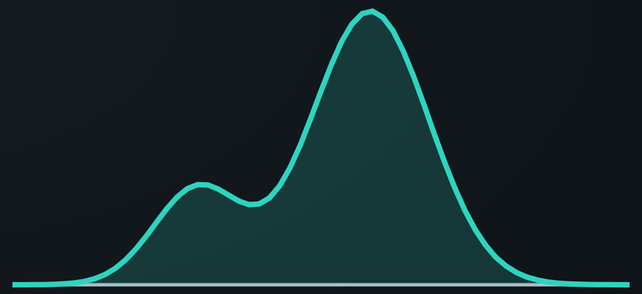

## Richer concepts: a vector or a whole distribution

representation ●●●●●○

**a vector (embedding)**

$$\begin{bmatrix} 0.31 \\ -1.20 \\ 0.85 \\ 0.02 \\ -0.47 \end{bmatrix}$$

**a distribution**



possible values of the concept

$$\hat{c}_j \in \mathbb{R}^m$$

or

$$p(c_j \mid x)$$

we meet these in Part 10

ILLUSTRATIVE VALUES · values

# What is a concept?



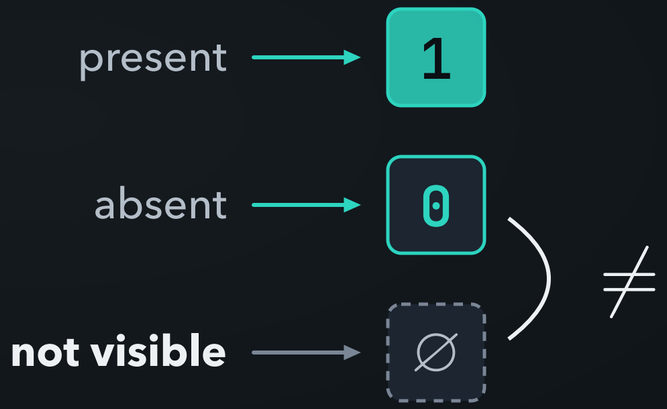

## Unknown: not visible is different from absent

CONSTRUCTED EXAMPLE · wingtip covered for the example




Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

representation




CUB label not visible: unknown, not absent

$$c_j = \varnothing \neq 0$$

# What is a concept?



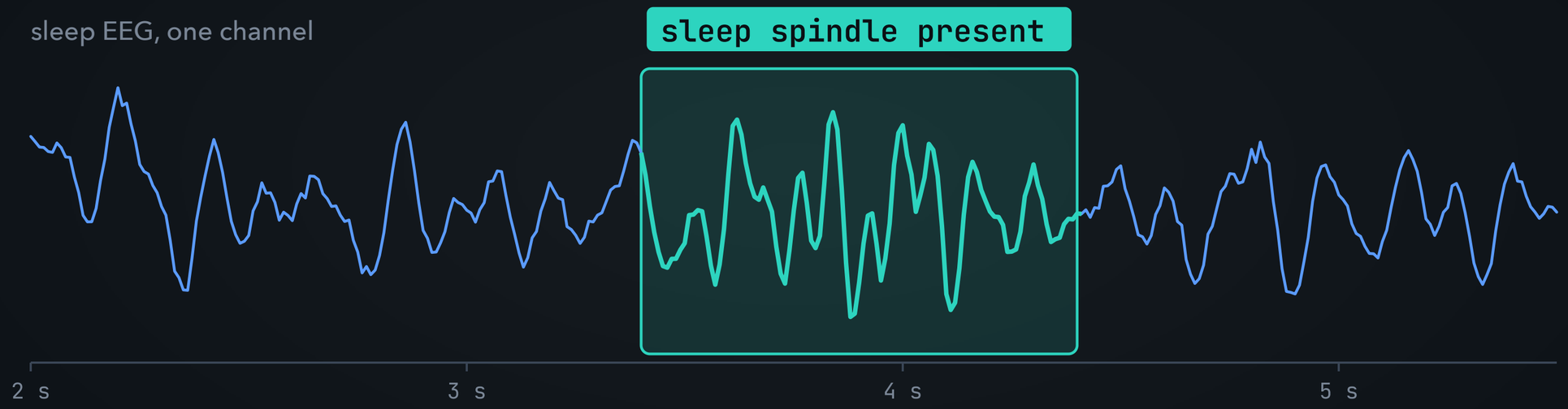

## Six ways to store a concept as a number

| type | in symbols | example |
|---|---|---|
| **binary** | $c_j \in \{0,\ 1\}$ | grey wingtip: 1 |
| **soft** | $\hat{c}_j = \sigma(z_j)$ | logit 1.4 gives 0.80 |
| **categorical** | $\hat{c}_{\text{wing}} = \text{softmax}(\mathbf{z})$ | grey 0.72, scores sum to 1 |
| **continuous** | $\hat{c}_{\text{bill}} \in \mathbb{R}$ | bill length 54 mm |
| **vector or distribution** | $\hat{\boldsymbol{c}}_j \in \mathbb{R}^m,\ \ p(c_j \mid \boldsymbol{x})$ | more in Part 10 |
| **unknown** | $c_j = \varnothing$ | not visible, not absent |

# What is a concept?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The same pattern in a sleep EEG recording



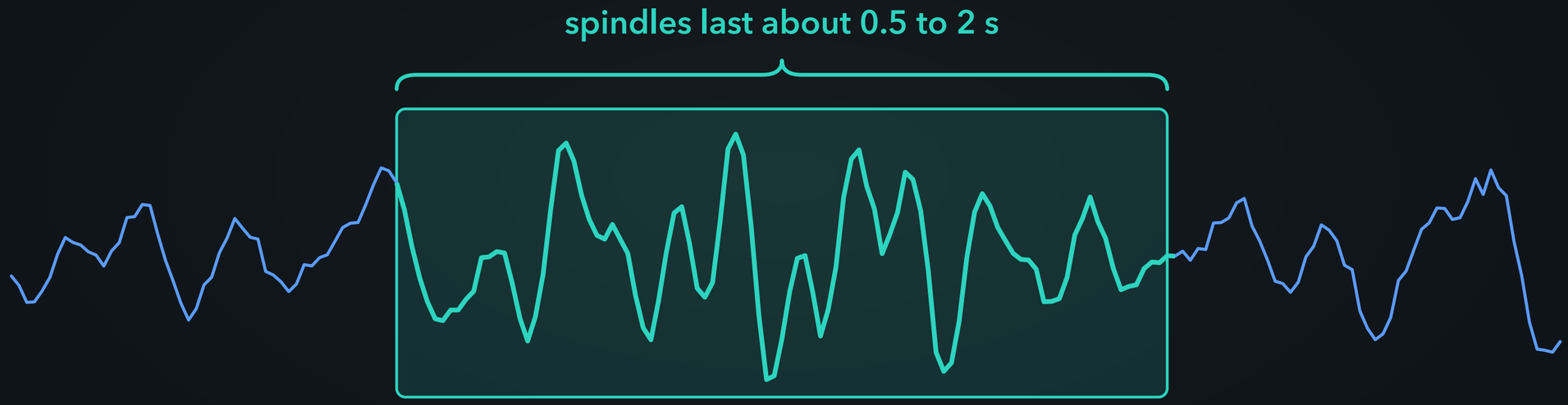

EEG trace → spindle present → sleep stage

SCHEMATIC · seeded sinusoids, not a recording

# What is a concept?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## A spindle is a brief burst of brain waves




a burst of 11 to 16 cycles per second (Hz)

sleep scorers: a sign of **stage 2 sleep**

SCHEMATIC · this spindle: 13 Hz, 1 s; not a recording

# What is a concept?



## The same pattern in a customer message

CUSTOMER MESSAGE

The app works, but I was charged twice. Please refund the extra payment.

↓

`duplicate charge reported`

the customer says one purchase was billed twice

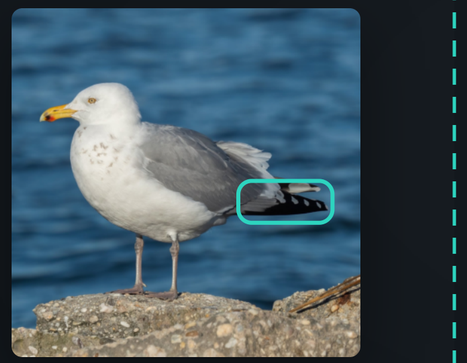
message → duplicate charge → support queue

ILLUSTRATIVE VALUES · an example message

# What is a concept?

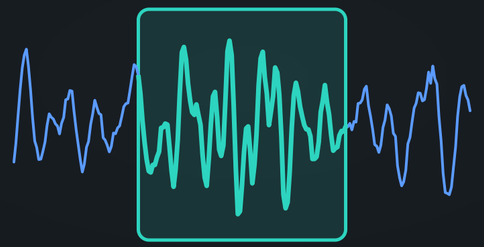
$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Each **concept** is a named claim that a specialist can check



`black wingtip`

bird expert ✓



`spindle present`

sleep scorer ✓

"…I was charged twice…"

`duplicate charge`

support agent ✓

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

PART 4

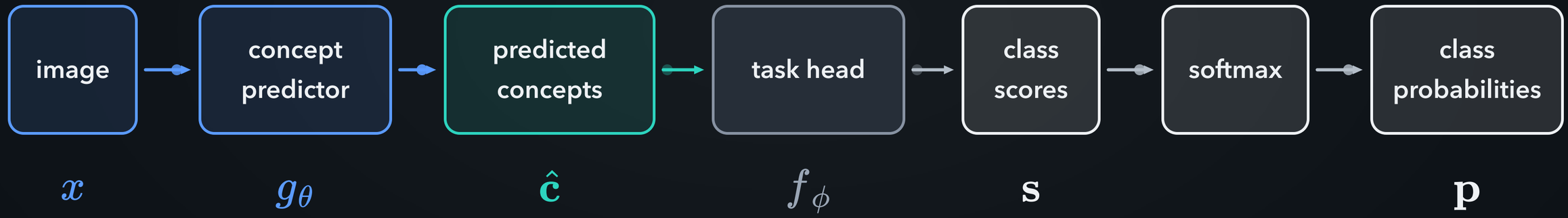
# From concepts to a prediction

Follow one image through the task head

# From concepts to predictions: the arithmetic



## Follow the image all the way to class probabilities




task head: a linear layer or a multilayer perceptron (MLP)

our worked example uses one linear layer

# From concepts to predictions: the arithmetic

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Three classes and three concepts

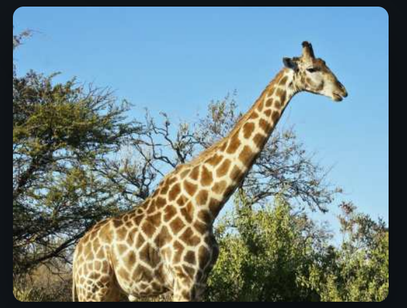
| | | | |
|---|---|---|---|
| classes |  zebra |  horse |  giraffe |
| concepts | stripes | horse-like shape | long neck |

Photo: Yathin S Krishnappa, Wikimedia Commons, CC BY-SA 3.0
Photo: Dietmar Rabich, Wikimedia Commons, CC BY-SA 4.0
Photo: Hans Hillewaert, Wikimedia Commons, CC BY-SA 4.0

ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic



## The concept predictor turns this image into three estimates



input image

concept predictor

$g_\theta$

$\hat{\mathbf{c}}$

| | |
|---|---|
| stripes | 0.1 |
| horse-like shape | 0.9 |
| long neck | 0.1 |

assumed outputs for this photo

$$\hat{\mathbf{c}} = g_\theta(x)$$

the hat means predicted

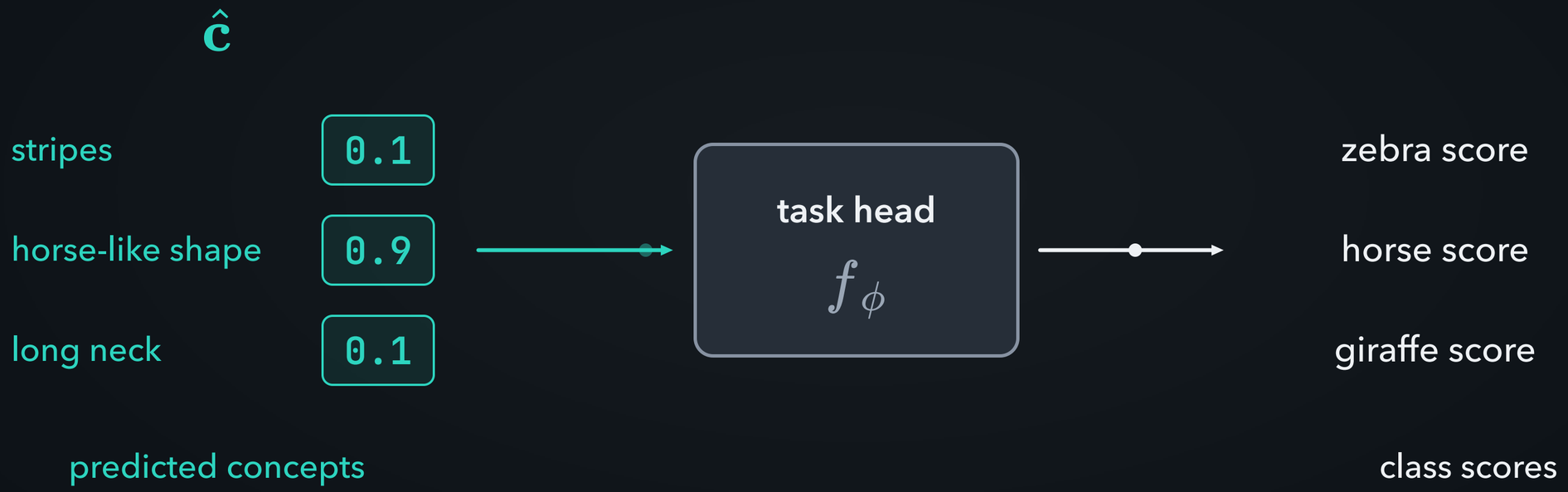
Photo: Yathin S Krishnappa, Wikimedia Commons, CC BY-SA 3.0

ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic




## The task head turns predicted concepts into class scores




The task head reads only the concepts, not the image

ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic




## The concept values describe this image; the weights belong to the layer

values for this image

| | |
|---|---|
| stripes | 0.1 |
| horse-like shape | 0.9 |
| long neck | 0.1 |

$\hat{c}$

change with the image

weights inside the linear layer

| | stripes | horse-like shape | long neck |
|---|---|---|---|
| zebra | 3 | 1 | 0 |
| horse | -1 | 2 | -1 |
| giraffe | -1 | 0 | 3 |

$W$

fixed during prediction

this example has no bias

ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic




## Inside the layer: multiply each concept value by its weight, then add

| | stripes | | horse-like shape | | long neck | | |
|---|---|---|---|---|---|---|---|
| zebra weights | 3 | × | 1 | × | 0 | | |
| this image's values | 0.1 | | 0.9 | | 0.1 | | |
| contributions | 0.3 | + | 0.9 | + | 0 | = | 1.2 |
| | | | | | | | zebra score |

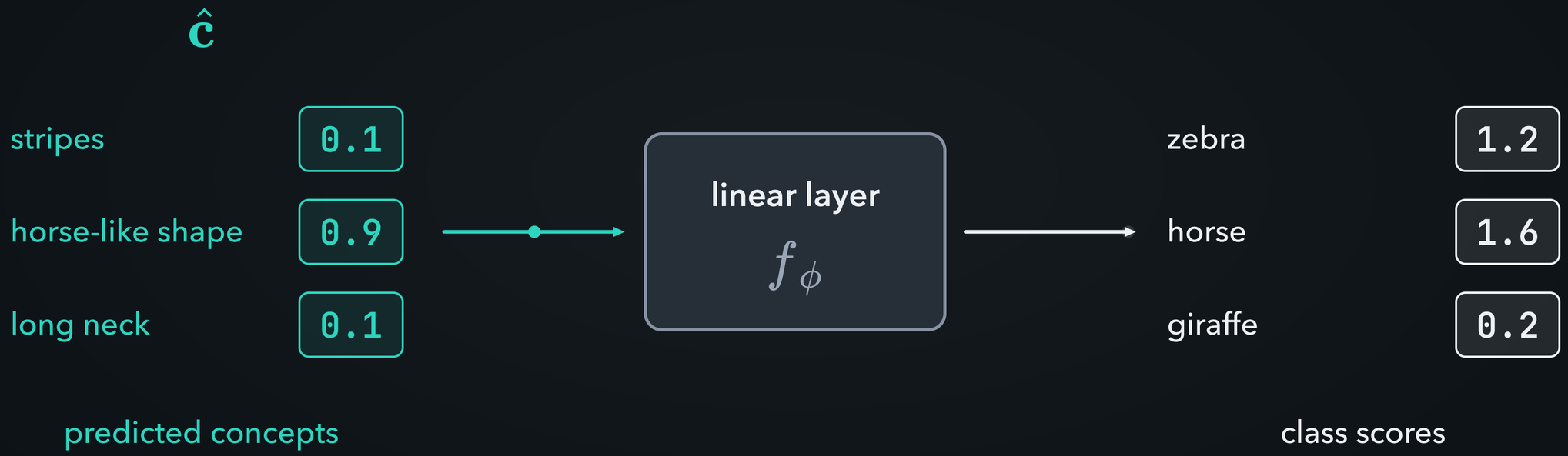
$$\text{zebra score} = 3 \times 0.1 + 1 \times 0.9 + 0 \times 0.1 = 1.2$$

ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic




## The same layer produces all three class scores




scores = layer weights × predicted concept values

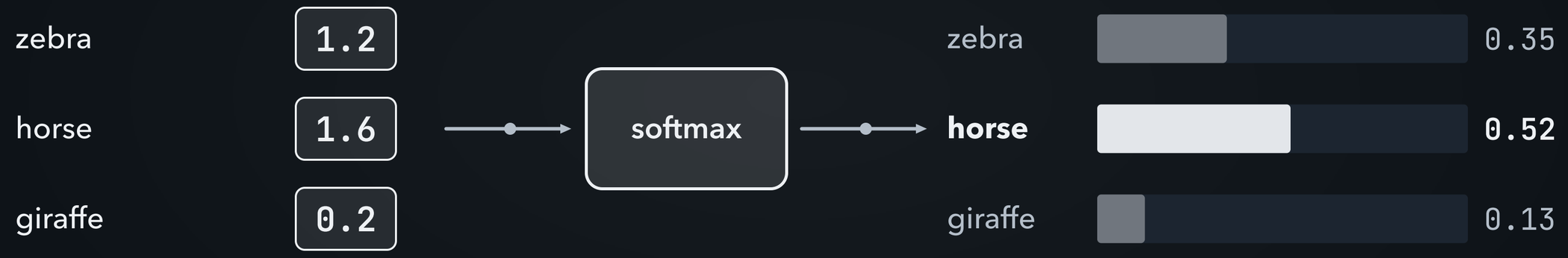
$$\mathbf{s} = f_\phi(\hat{\mathbf{c}}) = W\hat{\mathbf{c}}$$

ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic



## Softmax turns the scores into class probabilities




$$p_k = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

probabilities sum to one

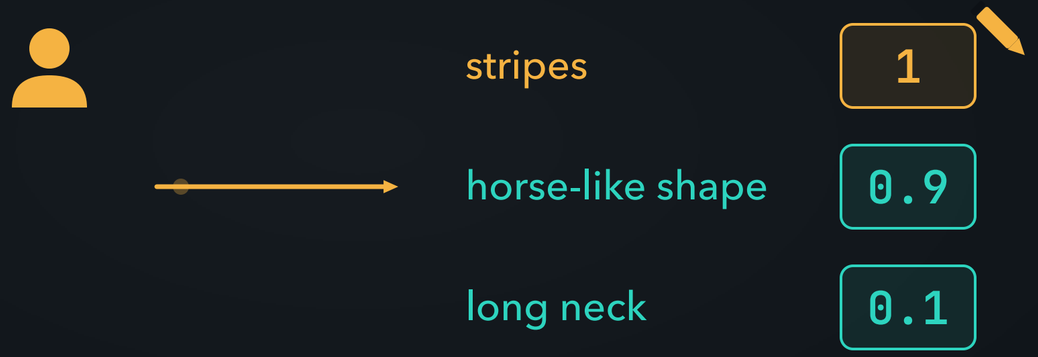
ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic

$x$ $g_\theta$ $\hat{c}$ $f_\phi$ $\hat{y}$

## A person corrects one predicted concept






$$\text{stripes: } 0.1 \rightarrow 1$$

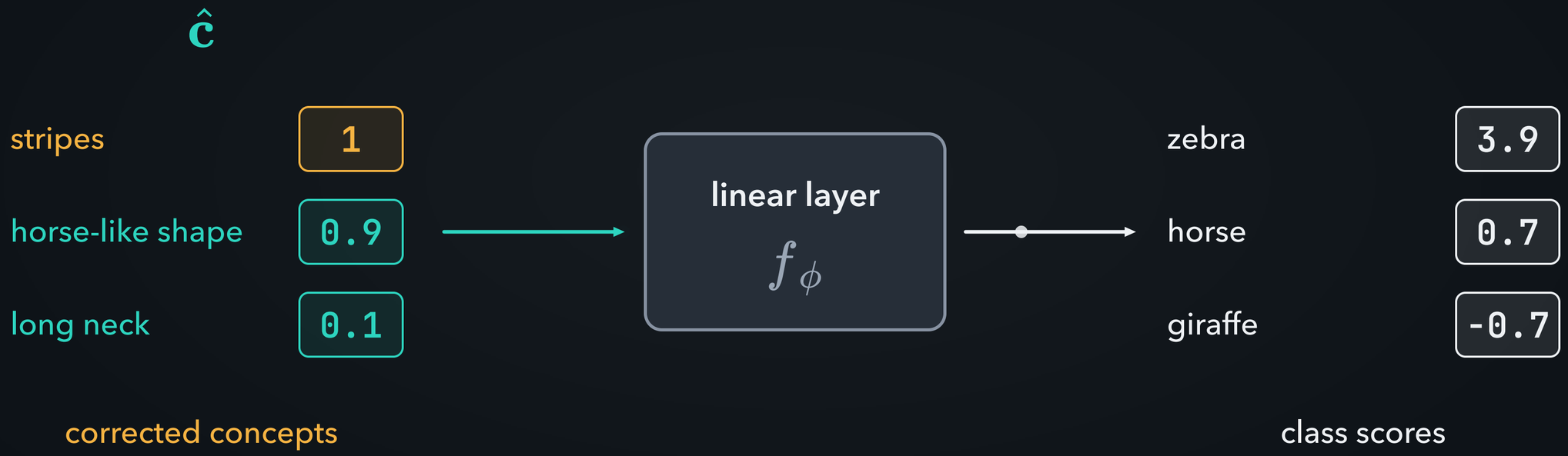
Photo: Yathin S Krishnappa, Wikimedia Commons, CC BY-SA 3.0

ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic




## Pass the corrected concepts through the same layer



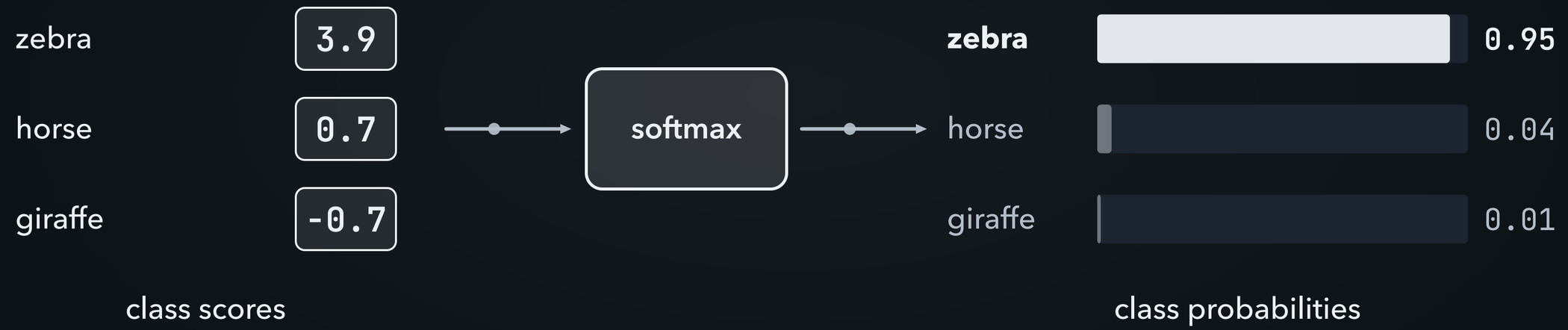

same image, same predictor, same weights; one corrected concept

ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic

$x$ $g_\theta$ $\hat{c}$ $f_\phi$ $\hat{y}$

## After softmax, zebra has the highest probability




before: horse 0.52

after: zebra 0.95

ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic




## Why did the zebra score rise?

| change in stripes | | zebra's stripes weight | | change in zebra score |
|---|---|---|---|---|
| $0.9$ | × | $3$ | = | $+2.7$ |

$$\text{zebra score: } 1.2 + 2.7 = 3.9$$

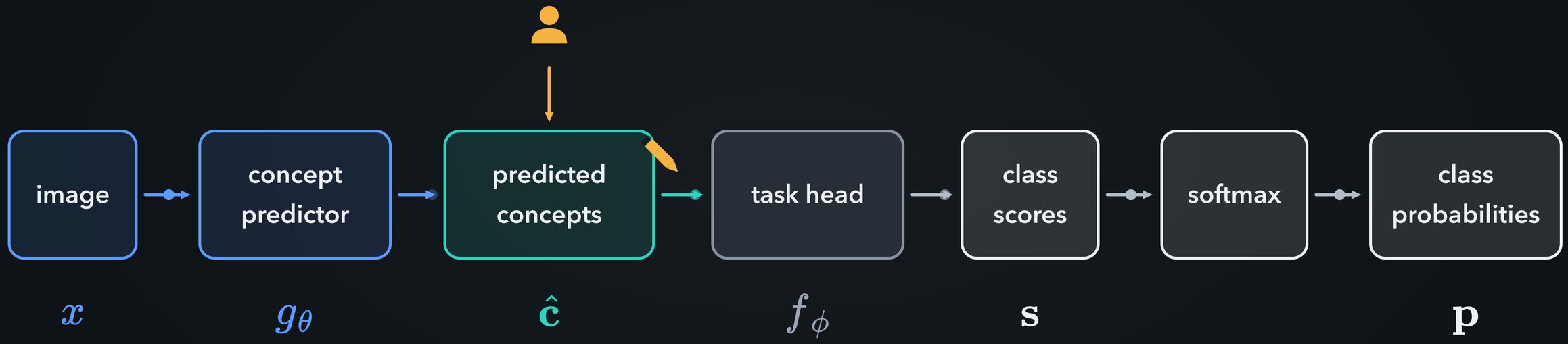
all other contributions stay the same

ILLUSTRATIVE VALUES · hand-chosen concept outputs and weights; arithmetic computed from these values

# From concepts to predictions: the arithmetic

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The concept vector is the point where a person can intervene




correct a concept, then run the same task head again

no retraining

PART 5

# Learning the concepts

What do the labels teach each part of the model?

# How does the model learn the concepts?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Where do the concept values come from?

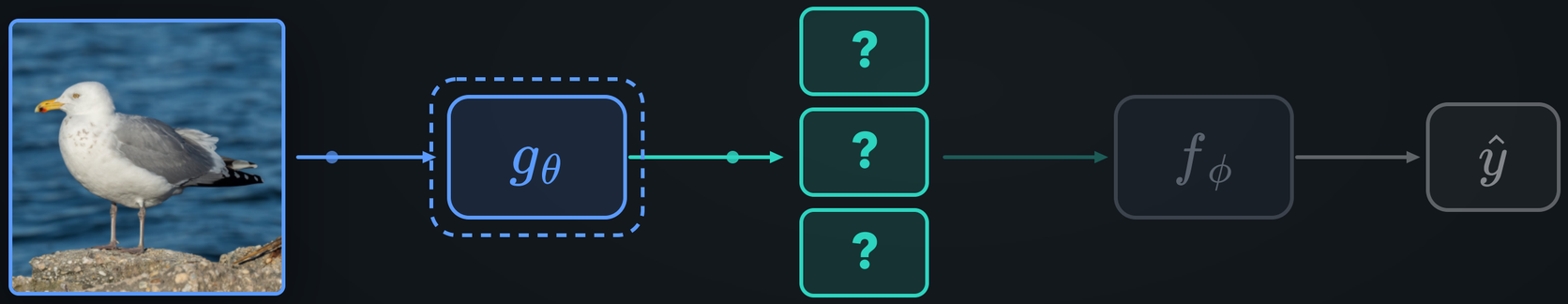



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

photo — concept predictor — concept values

learned from labelled examples

# How does the model learn the concepts?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Each training example carries three things



| | |
|---|---|
| hooked bill | 1 |
| grey wing | 1 |
| yellow wing | 0 |
| white throat | 1 |

... 112 labels in all

**Herring Gull**

input $\boldsymbol{x}_i$

concept labels, 0 or 1 $\mathbf{c}_i$

**task label** $y_i$

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

`DATASET · Herring Gull's labels in CUB, 4 of 112; our photo`

# How does the model learn the concepts?




## The concept loss compares each predicted concept with its label

for each concept, compare the prediction $\hat{c}_k$ with the label $c_k$

loss for concept k — label 1: a low prediction costs — label 0: a high prediction costs

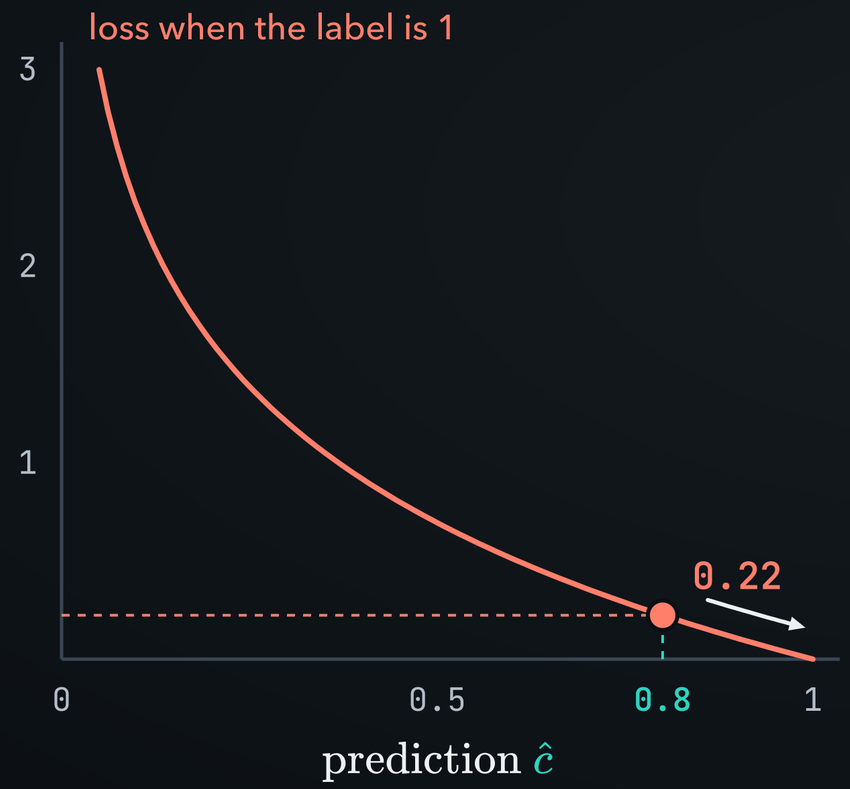
$$\ell_k = -c_k \log \hat{c}_k - (1 - c_k) \log(1 - \hat{c}_k)$$

binary cross-entropy, averaged over the concepts: $L_c = \frac{1}{K} \sum_k \ell_k$

# How does the model learn the concepts?



## One term of the concept loss, worked out




grey wing: label 1, prediction 0.8

$$\ell = -\left[1 \cdot \log 0.8 + 0 \cdot \log(1 - 0.8)\right]$$

$$= -\log 0.8 \approx 0.22$$

closer to 1, smaller loss

ILLUSTRATIVE VALUES · a hand-chosen prediction

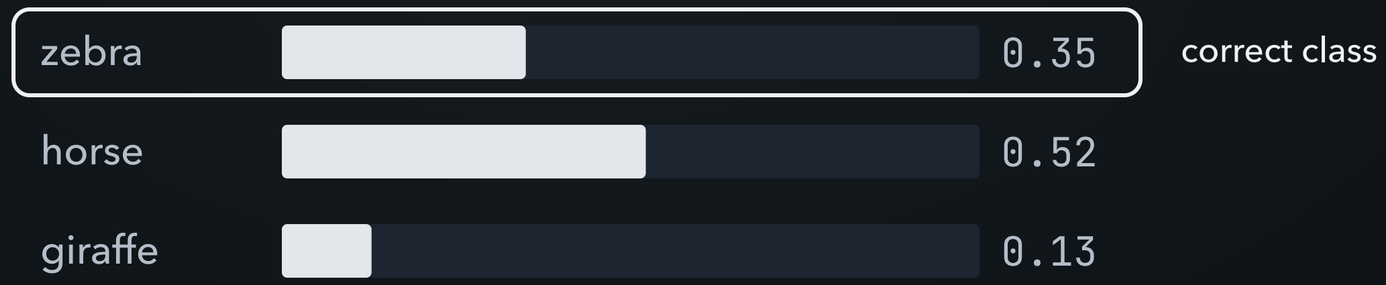
# How does the model learn the concepts?




## The task loss compares the prediction with the task label

task loss: minus the log of the share given to the correct **class**

$$L_y = -\log \hat{y}_{y_i}$$




the zebra photo of Part 4: $L_y = -\log 0.35 \approx 1.05$

ILLUSTRATIVE VALUES · shares from the Part 4 worksheet

PART 5

# How does the model learn the concepts?




## A weighted sum combines the two losses

total loss — weight × concept loss — weight × task loss

$$L = \lambda_c L_c + \lambda_y L_y$$

$\lambda_c$ and $\lambda_y$ set how much each objective counts

PART 5

# How does the model learn the concepts?



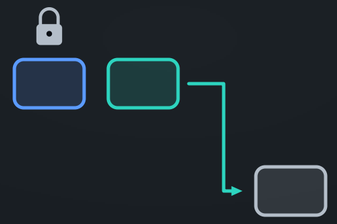

## Three ways to train the concept predictor and the task head

**INDEPENDENT**



each part on its own labels

**SEQUENTIAL**



predictor first, then the head

**JOINT**



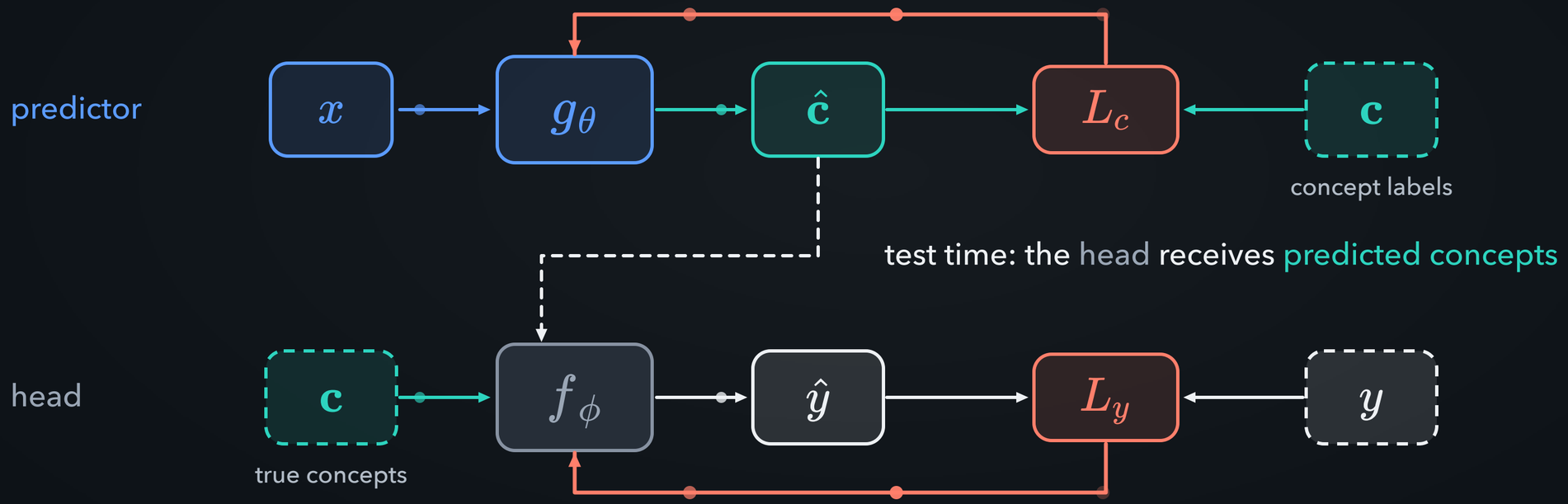
everything at once

Koh et al., ICML 2020

# How does the model learn the concepts?



## Independent: each part is trained on its own labels

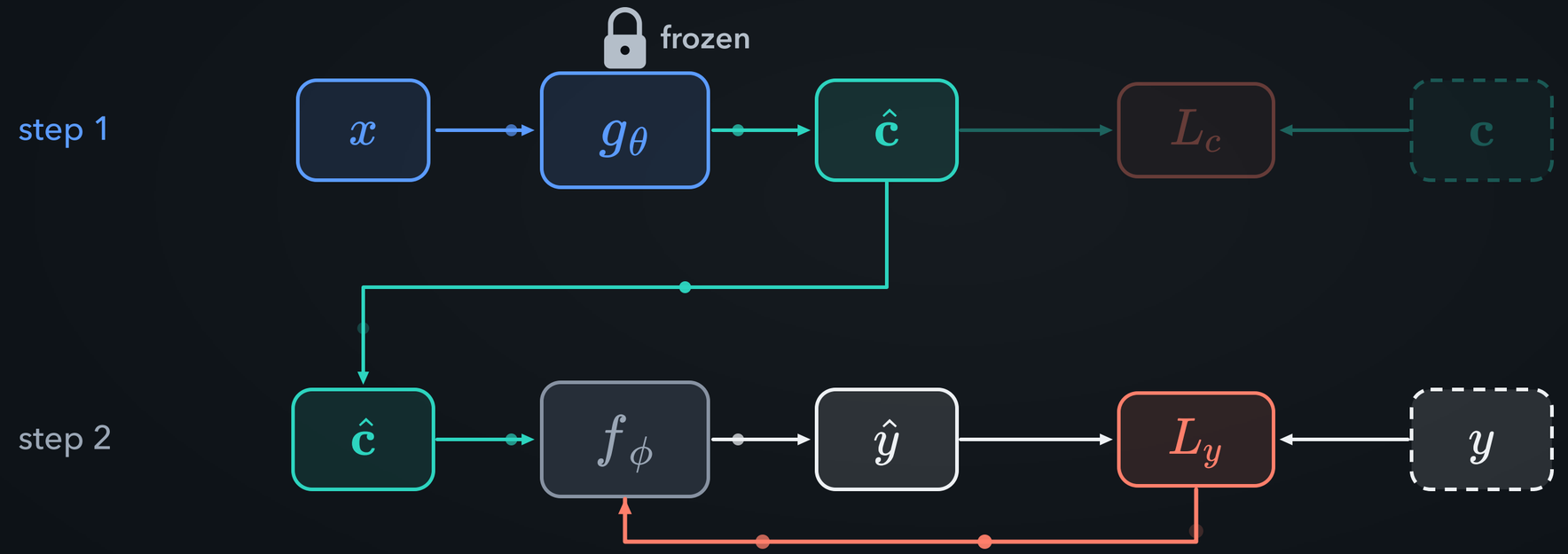



coral: the gradient of each loss, flowing back to what it trains

# How does the model learn the concepts?



## Sequential: freeze the predictor, then train the head on its outputs

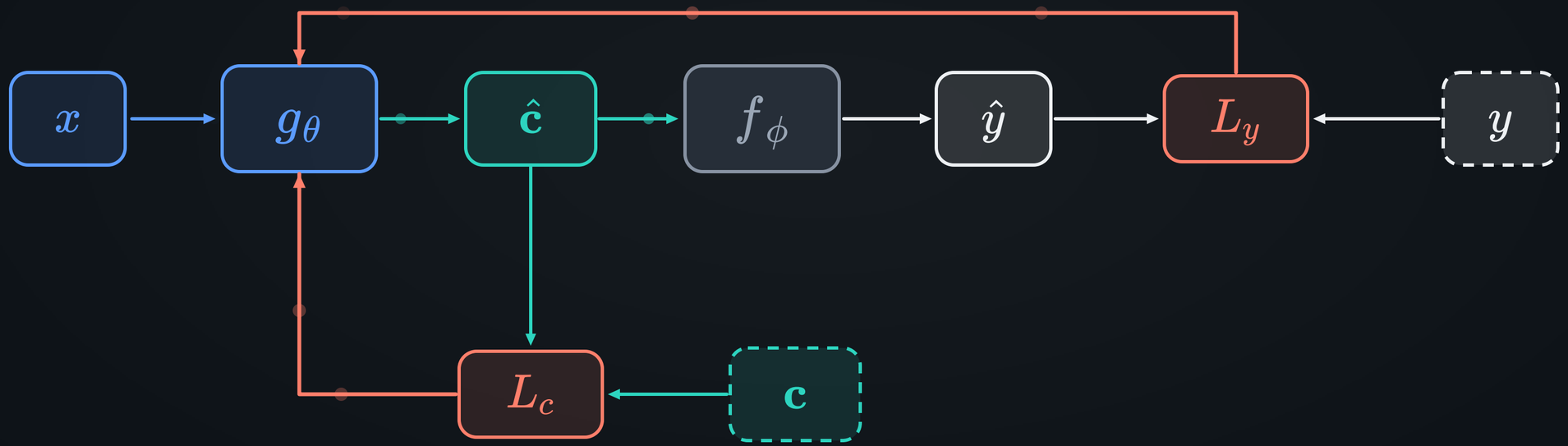



the head learns from the predictor's actual mistakes

# How does the model learn the concepts?



## Joint: train the whole system at once, with both losses

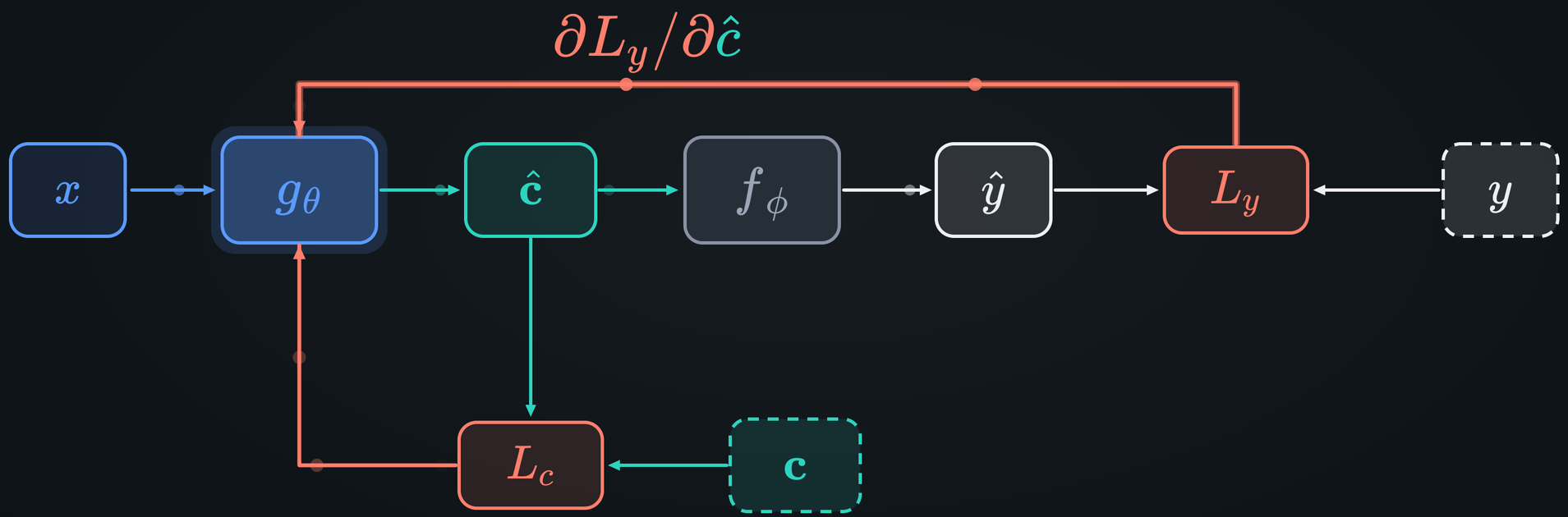



both losses train the whole system

# How does the model learn the concepts?




## In joint training, the task loss also shapes the concept predictor



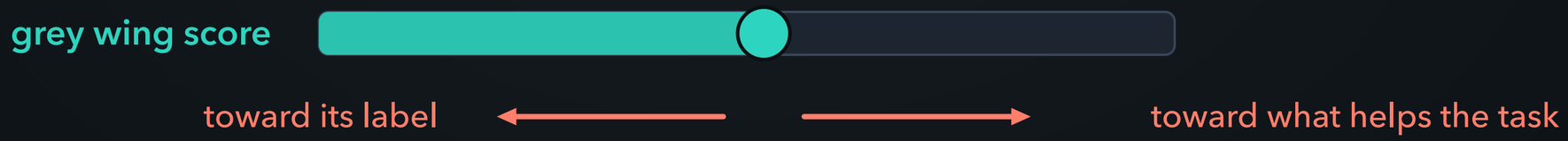

the concept predictor is shaped by both losses

# How does the model learn the concepts?




## A concept score can be pulled two ways

can improve **task accuracy**, can hurt concept fidelity




**whether it happens has to be measured**

# How does the model learn the concepts?




## Our measured run: joint training, $\lambda_c = 2$, $\lambda_y = 1$

concept loss, weight 2 task loss, weight 1

$$L = 2\,L_c + 1\,L_y$$

joint training, in the experiment of Part 11

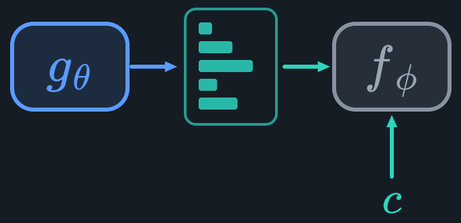
MEASURED · settings of the trained shape model, Part 11

# The three schemes on CUB




## Does the training scheme matter on real photographs?

**independent**



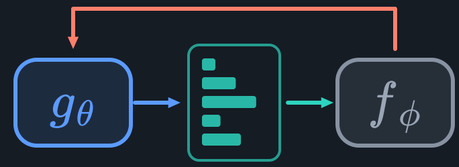

head fitted on true concepts

**sequential**




predictor frozen, head fitted on its predictions

**joint**




both trained together on both losses

our small run on CUB: 5,994 training photos, 5,794 test photos

PART 5

# The three schemes on CUB

$x$ $g_\theta$ $\hat{c}$ $f_\phi$ $\hat{y}$

## Our setup: frozen CLIP features → 112 concepts → 200 species

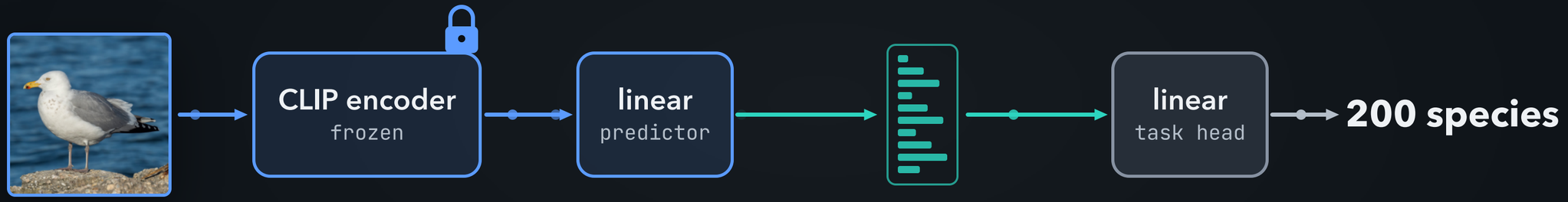



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

$x$ $e(x) \in \mathbb{R}^{512}$ $g_\theta$ $\hat{\mathbf{c}} \in [0,1]^{112}$ $f_\phi$ $\hat{y}$

CUB official split: 5,994 training photos, 5,794 test photos

MEASURED · our run: frozen CLIP ViT-B/32 + linear models, CUB official split, one run

# The three schemes on CUB



## One photo through the model: features, then concepts, then species




Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

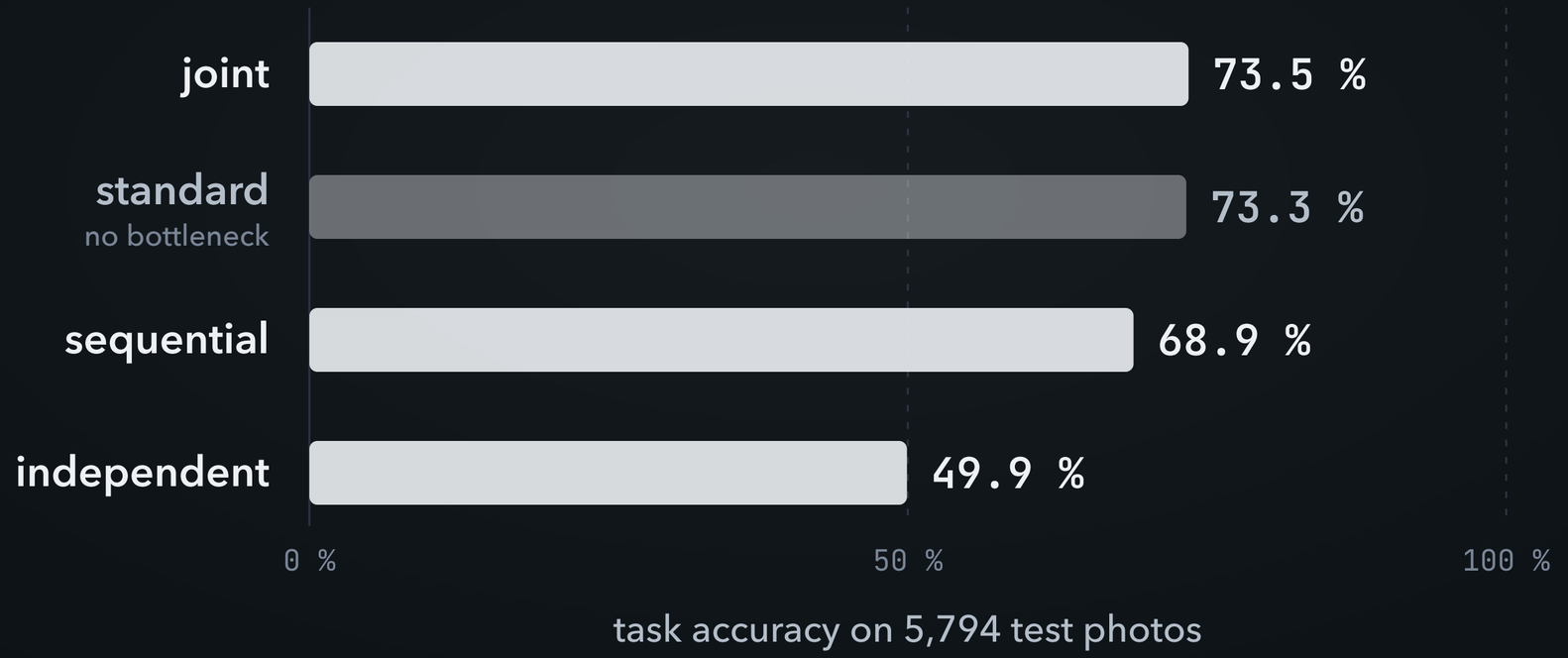
COMPUTED · REAL MODEL · our gull photo: CLIP ViT-B/32 feature, our CUB model

PART 5

# The three schemes on CUB

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Joint training is the most accurate, independent the least

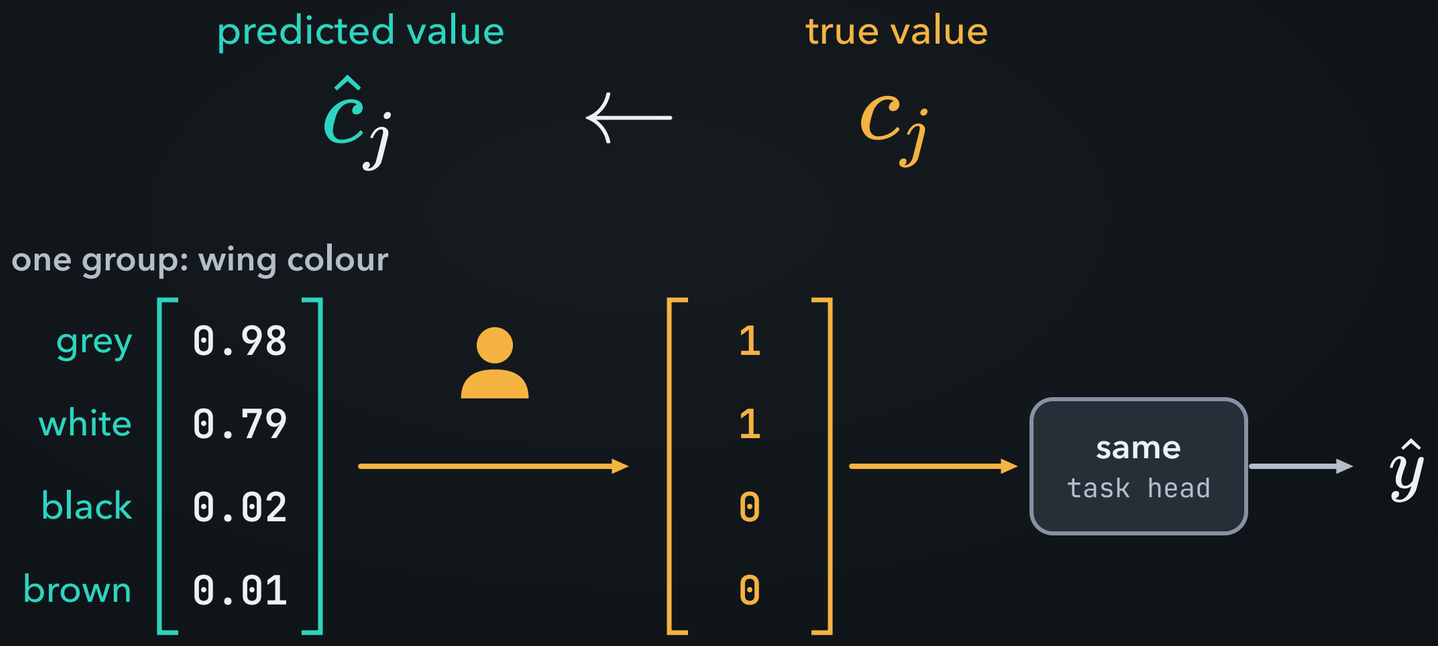



MEASURED · our CUB run, one run

# The three schemes on CUB



## An intervention replaces predicted values with true values



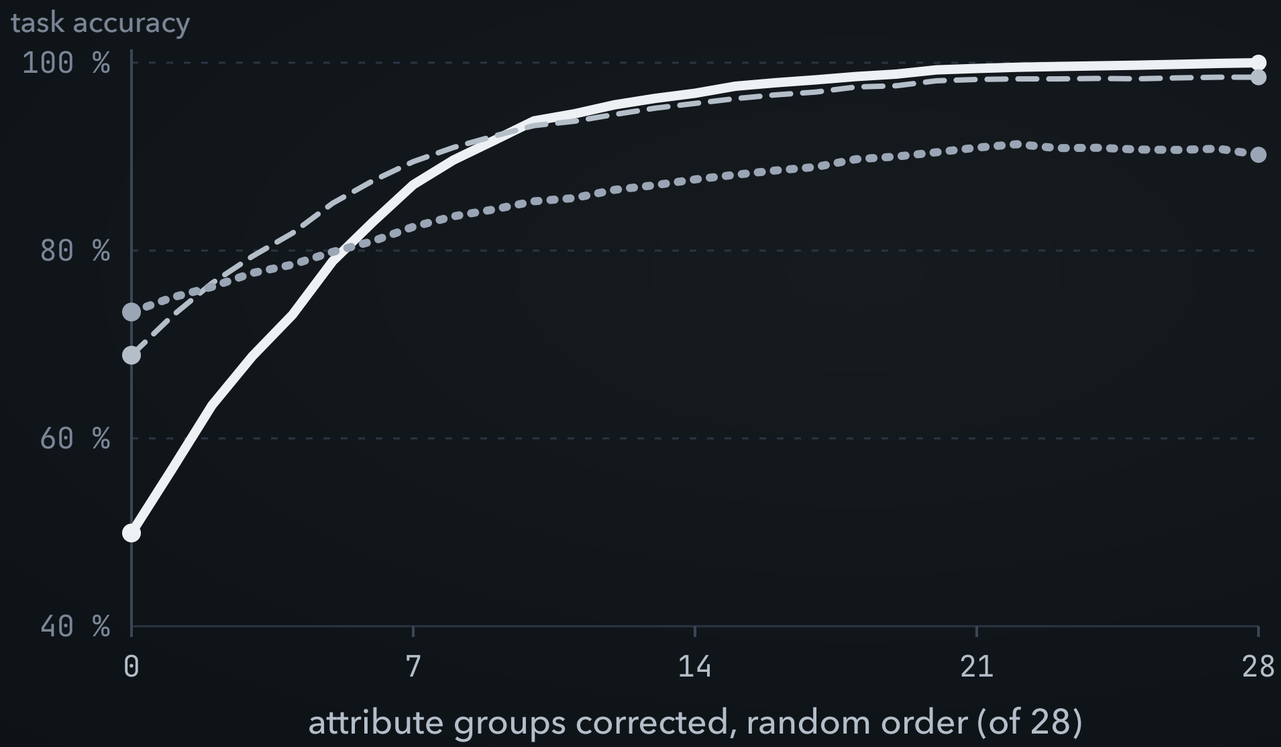

COMPUTED · REAL MODEL · our model on our gull photo; true values: Herring Gull's CUB labels

PART 5

# The three schemes on CUB

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## As more groups are corrected, the independent model gains most




| | |
|---|---|
| independent | 100.0 % |
| sequential | 98.5 % |
| joint | 90.2 % |

all 28 groups true: every species has its own pattern, so **independent** reaches 100.0 %

MEASURED · our CUB run, one run, averaged over random orders

PART 6

# Human interventions

What changes when a person corrects a concept?

# Human interventions

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## A person can fix a concept the model got wrong



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

wingtip black 0.20

the photo and the head stay the same; only a value changes

ILLUSTRATIVE VALUES · hand-written concept value (Part 2)

# Human interventions




## An intervention has four steps

1. predict concept values
2. a person checks one and corrects it
3. the fixed head recomputes
4. a new **prediction**

$$\hat{\mathbf{c}} = g_\theta(x) \qquad \hat{c}_j \leftarrow c_j \qquad f_\phi \text{ unchanged} \qquad \hat{y}_{\text{new}} = f_\phi(\hat{\mathbf{c}}_{\text{edited}})$$

# Human interventions



## Wing colour is one group: correct it together




Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

wing colour is one categorical concept: grey → 0, black → 1, together

ILLUSTRATIVE VALUES · hand-written concept values and head

# Human interventions




## Changing only one value could leave an impossible state

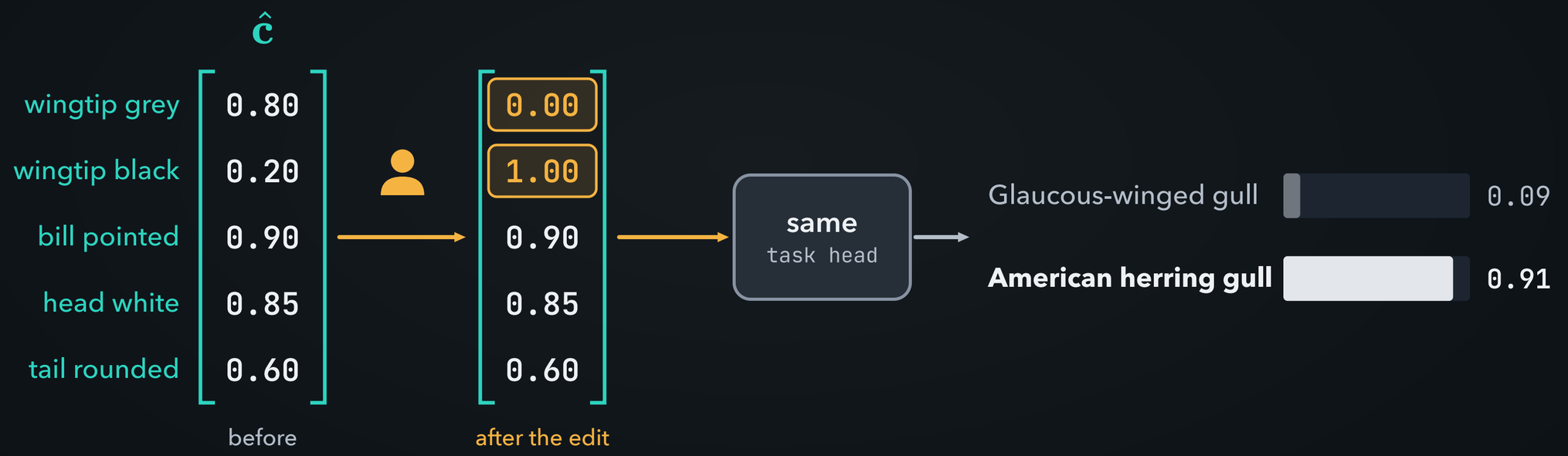



a wingtip has one colour, so the group changes as one

# Human interventions



## Same head, new values: herring gull rises




ILLUSTRATIVE VALUES · hand-written concept values and head

# Human interventions

$x$ $g_\theta$ $\hat{c}$ $f_\phi$ $\hat{y}$

## The same mechanism works on a signal

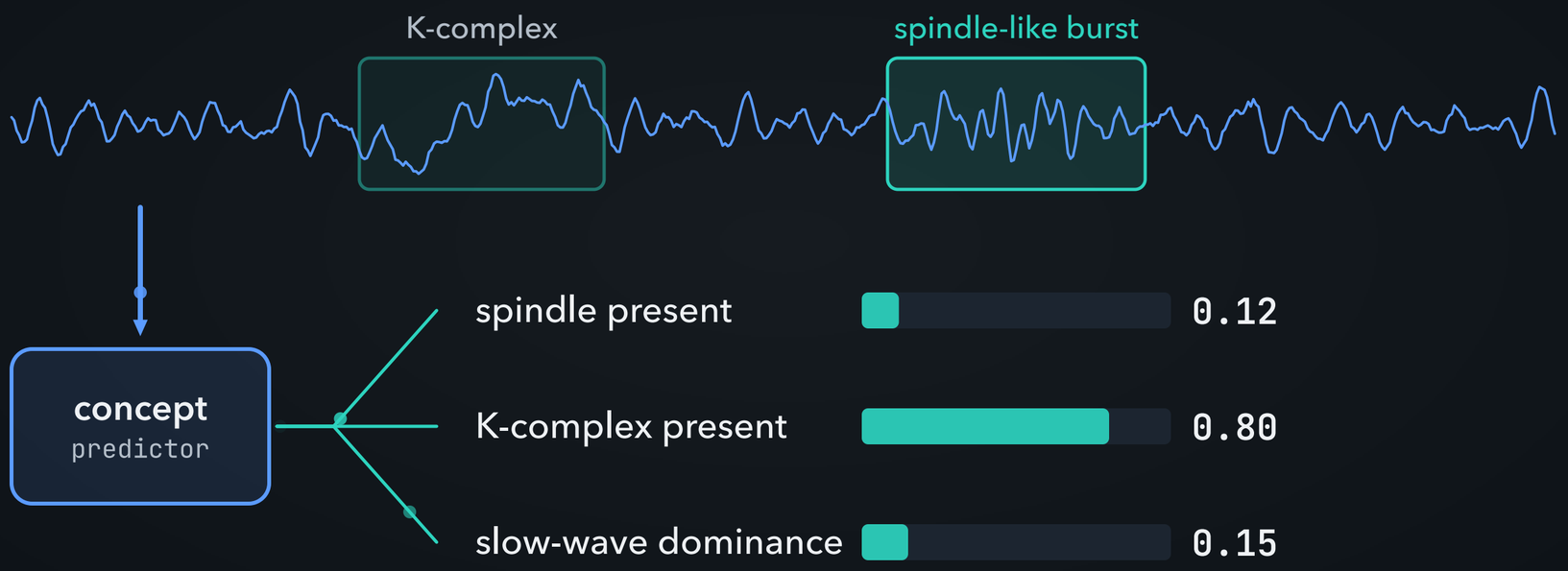



SCHEMATIC · authored trace, not a recording

# Human interventions

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The concept predictor gives spindle a low value

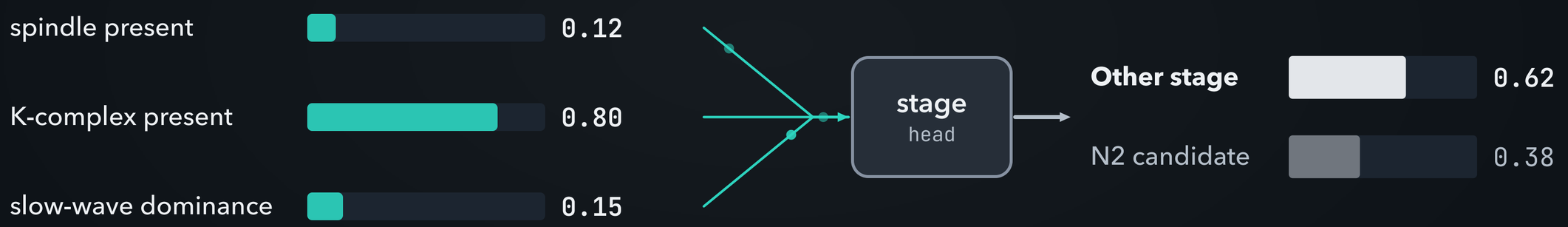



ILLUSTRATIVE VALUES · hand-written concept values; schematic trace

# Human interventions

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## An illustrative stage head favours other stage

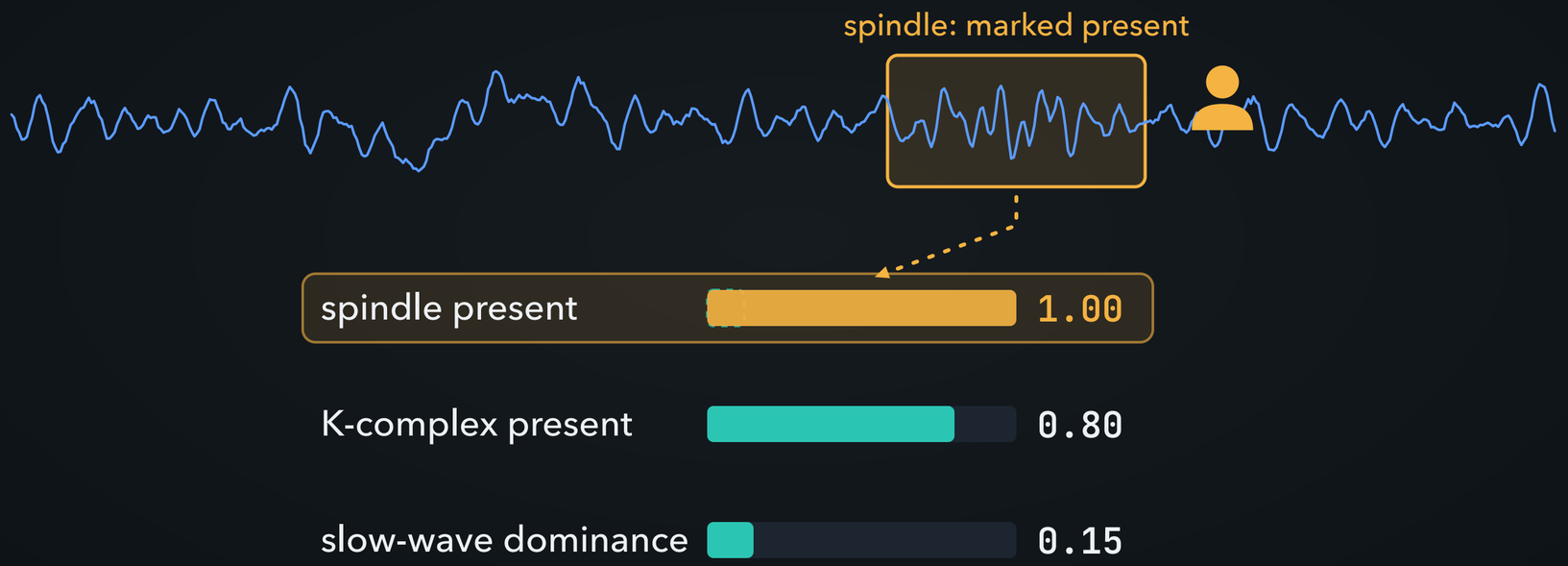



ILLUSTRATIVE VALUES · hand-written stage head

# Human interventions

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## A sleep expert marks the spindle as present

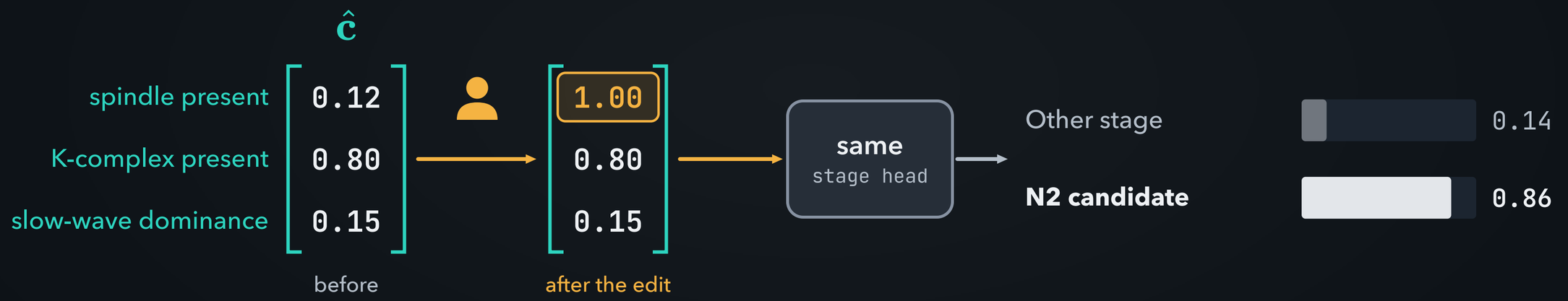



ILLUSTRATIVE VALUES · hand-written concept values; schematic trace

# Human interventions



## Same head, new input: N2 candidate rises



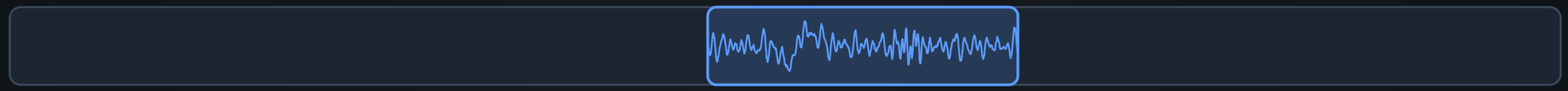

ILLUSTRATIVE VALUES · hand-written stage head

# Human interventions

$x$ $g_\theta$ $\hat{c}$ $f_\phi$ $\hat{y}$

## A demonstration of the mechanism, not a clinical result

real staging: scoring rules applied to 30 s epochs



0 s — our 6 s trace — 30 s

this stage head was written by hand

SCHEMATIC · authored EEG trace

ILLUSTRATIVE VALUES · hand-written stage head

PART 7

# When a correct edit hurts

A better concept value need not give a better answer

PART 7

# A correct edit can still hurt




## Does a correct edit always help?

| gull, Part 2 | zebra, Part 4 | EEG, Part 6 | a correct edit |
|---|---|---|---|
| **herring gull** 0.29 → 0.91 | **zebra** 0.35 → 0.95 | **N2 candidate** 0.38 → 0.86 | ? |
| ✓ | ✓ | ✓ | ✗ |

so far, every correction helped

PART 7

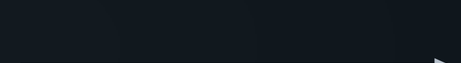
# A correct edit can still hurt

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## A support message that belongs in billing

SUPPORT MESSAGE

The app works, but I was charged twice. Please refund the extra payment.



BILLING QUEUE

where it belongs

CONSTRUCTED EXAMPLE · an invented message

PART 7

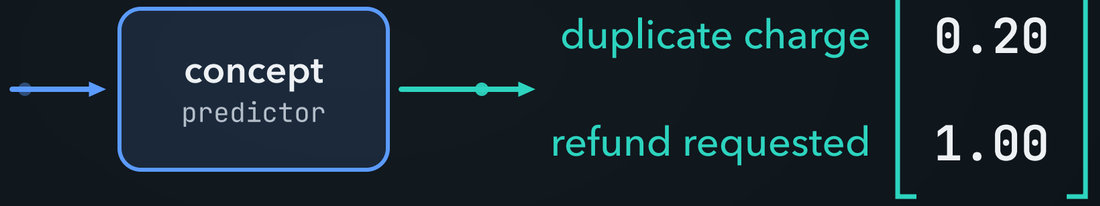
# A correct edit can still hurt

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The concept predictor reads two concepts from the message

SUPPORT MESSAGE

The app works, but I was charged twice. Please refund the extra payment.

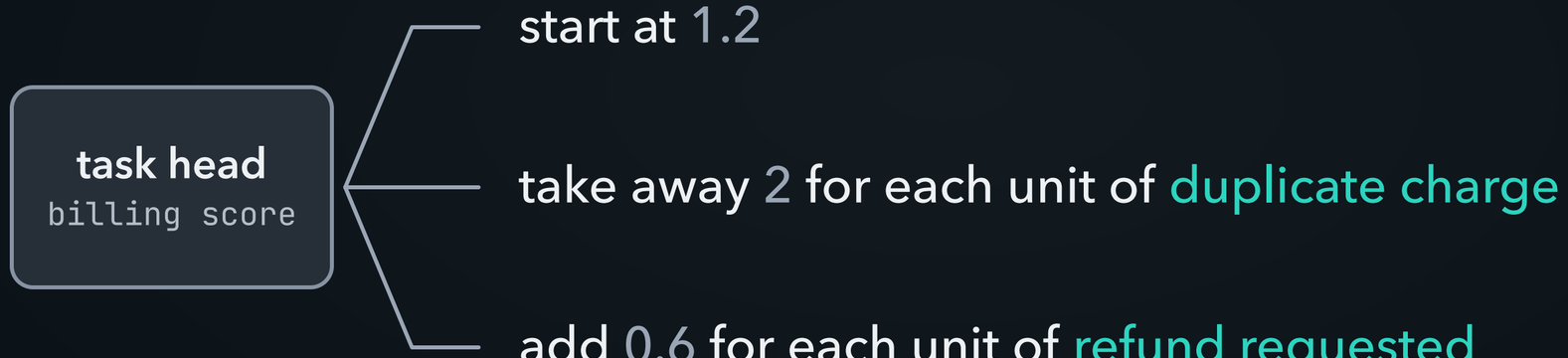



ILLUSTRATIVE VALUES · hand-written concept values

# A correct edit can still hurt



## The task head, in words




CONSTRUCTED EXAMPLE · a hand-written head

# A correct edit can still hurt

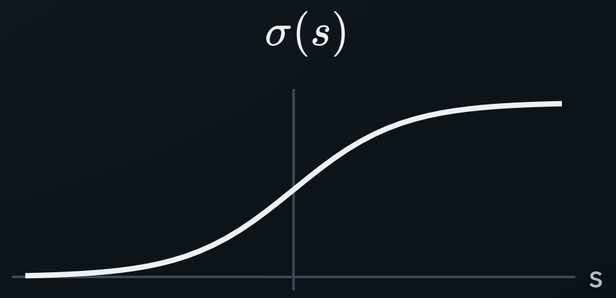


## The same head in symbols

score start duplicate charge refund requested

$$s = 1.2 - 2\,\hat{c}_{\text{dup}} + 0.6\,\hat{c}_{\text{refund}}$$

a sigmoid turns the score $s$ into the **billing share**



PART 7

# A correct edit can still hurt

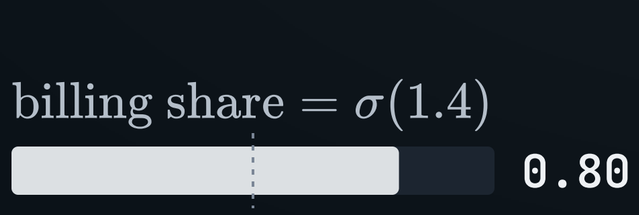
$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

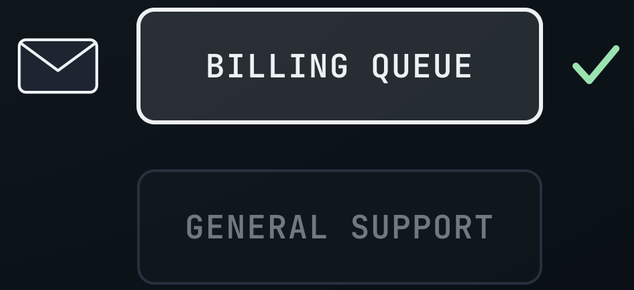
## Before the edit, the message goes to billing

$$s = 1.2 - 0.4 + 0.6 = 1.4$$

$-2 \times 0.20$

$0.6 \times 1.00$




CONSTRUCTED EXAMPLE · hand-written head

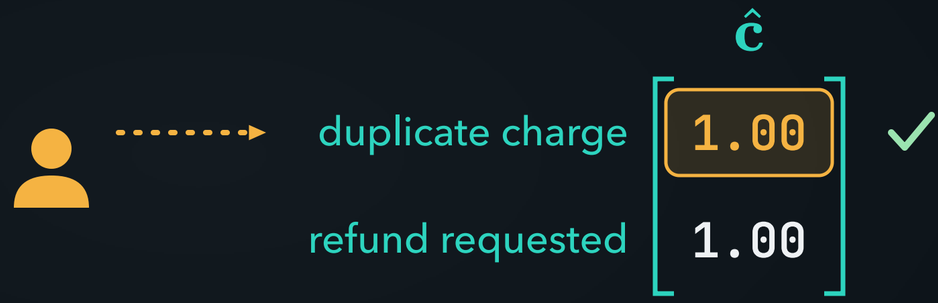
# A correct edit can still hurt




## A reviewer **correctly** sets duplicate charge to 1

SUPPORT MESSAGE

The app works, but I was charged twice. Please refund the extra payment.




matches the message: charged twice

PART 7

# A correct edit can still hurt




## After the edit, the message leaves billing

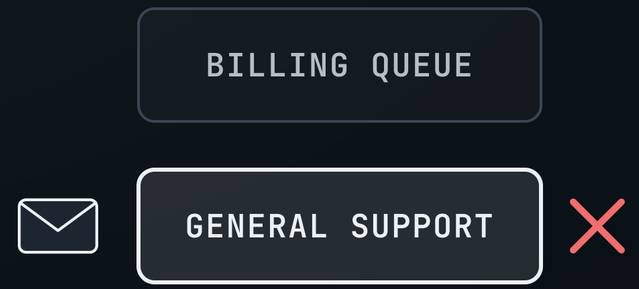
$$s = 1.2 \quad -2 \quad +0.6 \quad = -0.2$$

$-2 \times 1.00$

$0.6 \times 1.00$

$\text{billing share} = \sigma(-0.2)$




CONSTRUCTED EXAMPLE · hand-written head

# A correct edit can still hurt




## The fault is in the head: a weight with the wrong sign

$$s = 1.2 \quad -2\,\hat{c}_{\text{dup}} \quad +0.6\,\hat{c}_{\text{refund}}$$

pushes **billing** down as duplicate charge rises

✓ the edit: correct

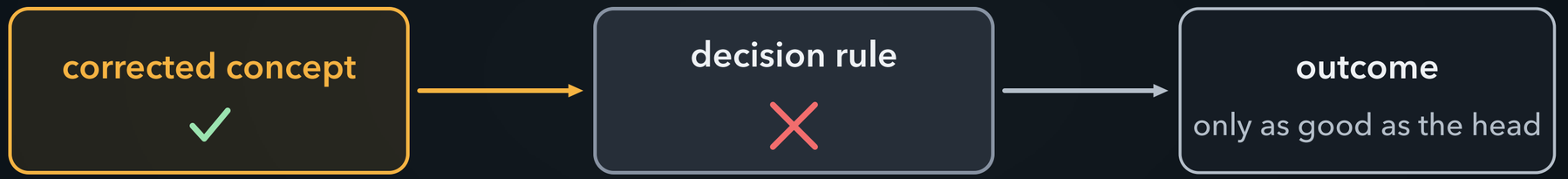
✗ the weight: wrong sign

CONSTRUCTED EXAMPLE · deliberately faulty head

# A correct edit can still hurt



## Correcting a concept does not repair the decision rule




measure the effect of an intervention; do not assume it

# A correct edit can still hurt




## A constructed case: the failure can happen

$$s = 1.2 - 2\,\hat{c}_{\text{dup}} + 0.6\,\hat{c}_{\text{refund}}$$

built faulty on purpose: the failure **can** happen

it says nothing about how often it does

CONSTRUCTED EXAMPLE · deliberately faulty head: existence, not frequency

PART 8

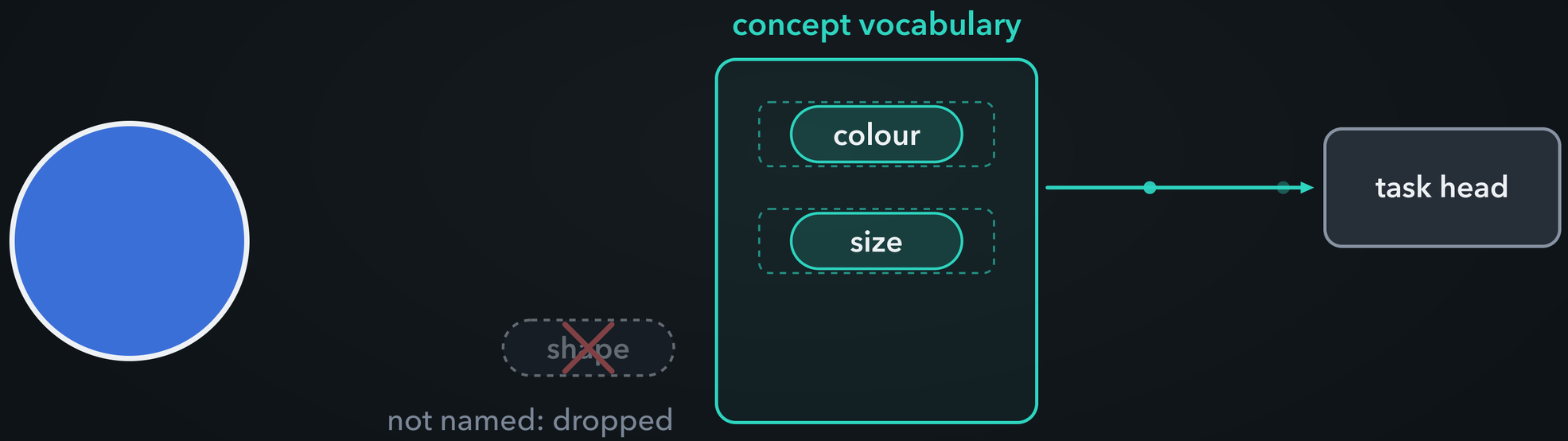
# An incomplete concept vocabulary

What if a useful distinction has no name?

# What if the concepts are incomplete?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The bottleneck can pass on only what the vocabulary names



# What if the concepts are incomplete?



## Two objects from two different classes




fill: the object's own colour

CONSTRUCTED EXAMPLE · two-object example

# What if the concepts are incomplete?



## The vocabulary names only colour and size

vocabulary: colour, size




both: blue = 1, large = 0




CONSTRUCTED EXAMPLE · two-object example

# What if the concepts are incomplete?



**Both objects get the same concept vector**

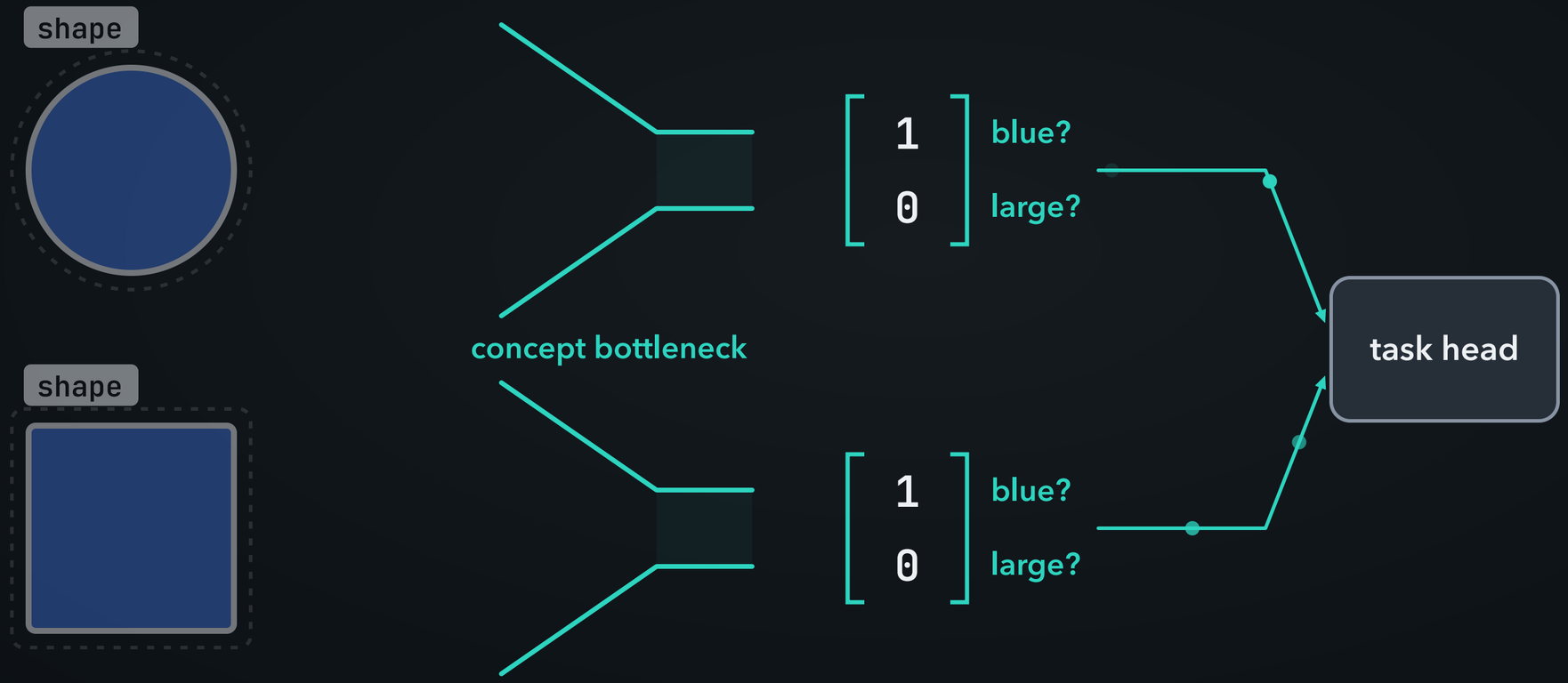



$$c(x_1) = c(x_2), \quad y_1 \neq y_2$$

# What if the concepts are incomplete?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Shape never reaches the task head

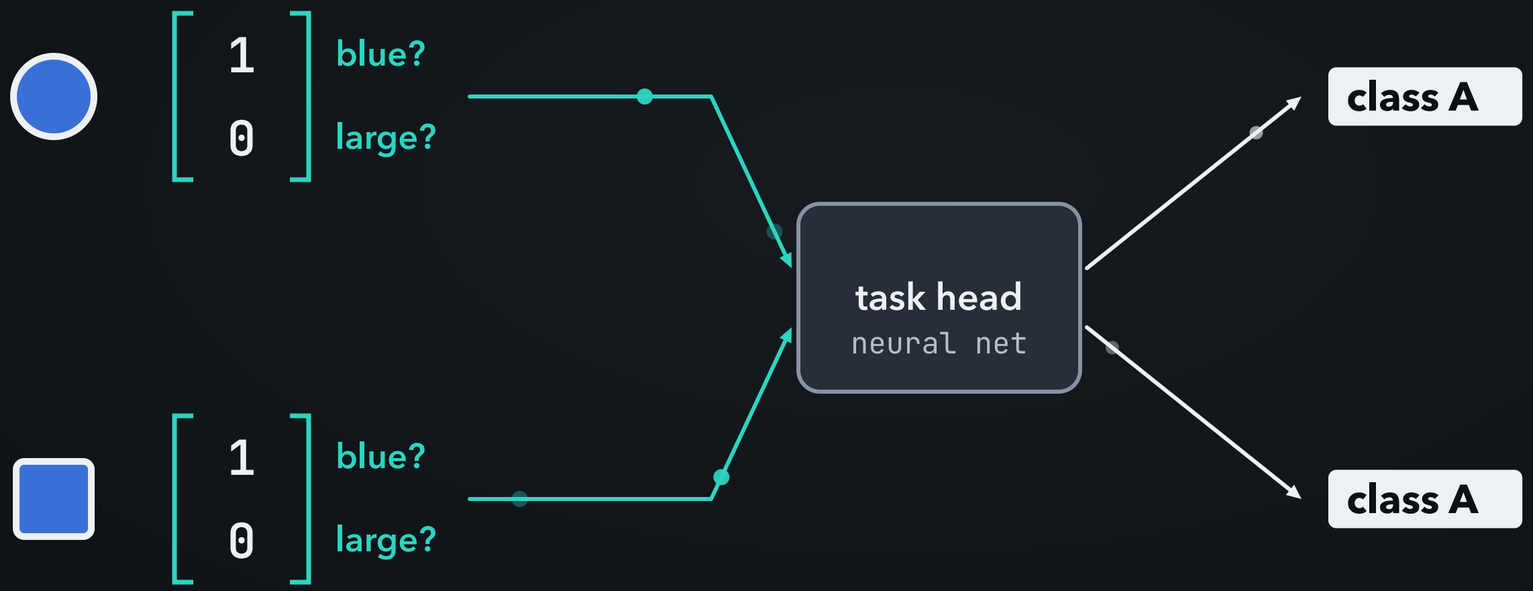



shape: in the image, not in the vector

# What if the concepts are incomplete?



## The same input to the head gives the same output

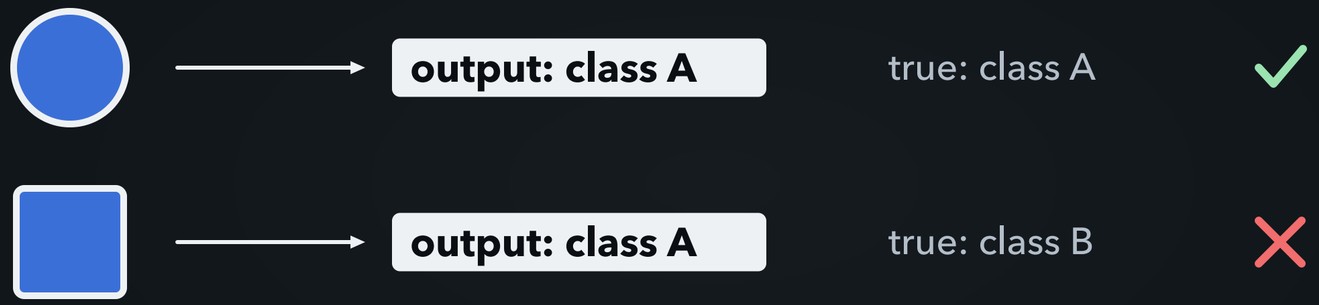



no head can recover what never reached it

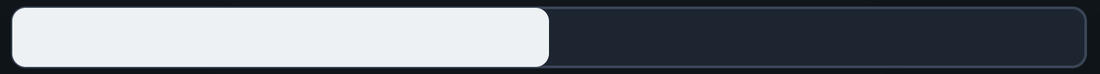
# What if the concepts are incomplete?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The best possible accuracy is one half




best possible accuracy



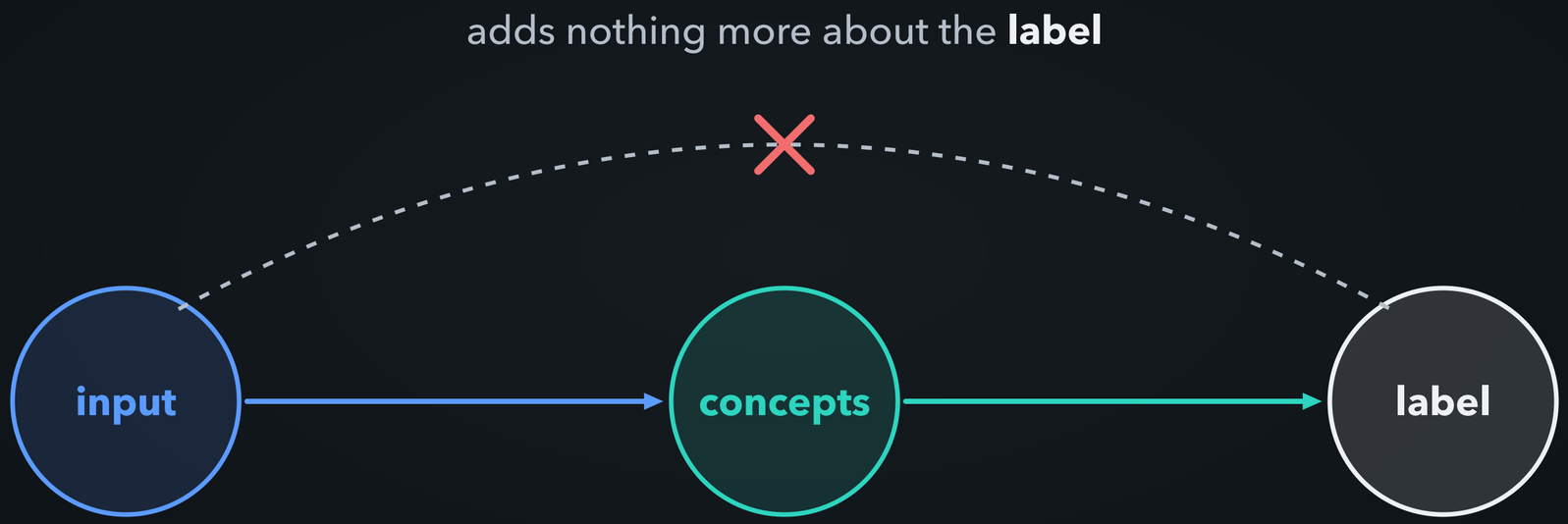
50%

CONSTRUCTED EXAMPLE · two-object example

# What if the concepts are incomplete?



## Sufficiency: the concepts say all the input says about the label




ideal: **sufficiency**

# What if the concepts are incomplete?



## Sufficiency in symbols

**label** independent of input given concepts

$$Y \not\perp\!\!\!\perp X \mid C$$

here it fails

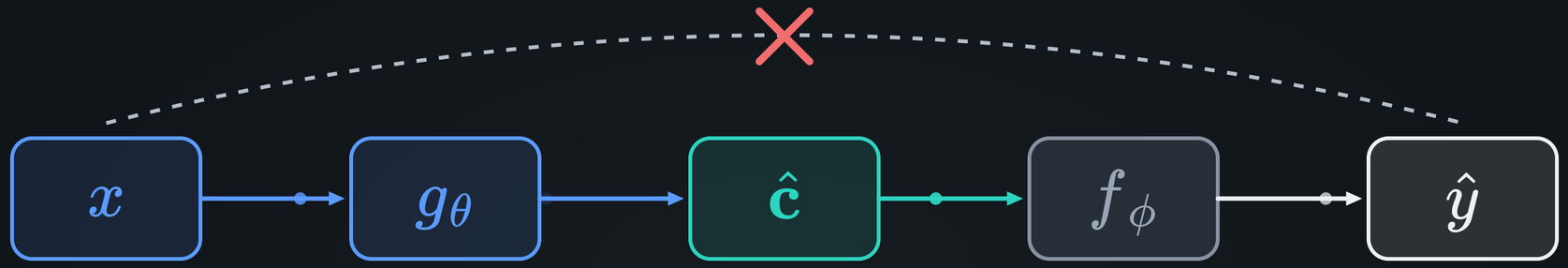


the **label** depends on shape, which $C$ lacks

# What if the concepts are incomplete?



## A strict CBM guarantees something weaker




the **prediction** depends on the input only through the concepts

$$\hat{y} = f_\phi(\hat{\mathbf{c}}), \qquad \hat{\mathbf{c}} = g_\theta(x)$$

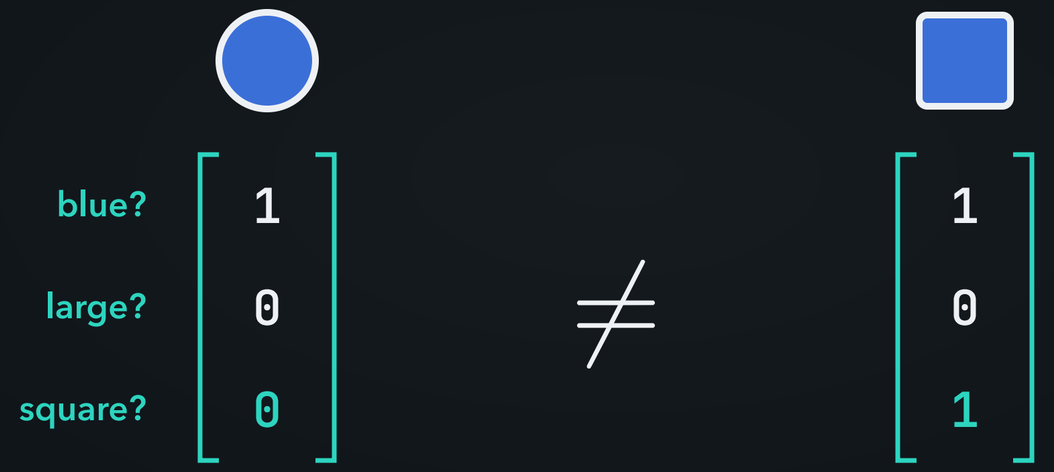
whether the true **label** does: a property of the vocabulary

# What if the concepts are incomplete?



## The fix is a richer vocabulary: add shape

vocabulary: colour, size, + shape




the two vectors now differ

CONSTRUCTED EXAMPLE · two-object example

PART 10

# Modern concept bottleneck architectures

Follow the ladder of supervision

# Modern concept bottleneck architectures




## Labelling every concept on every image is the main cost



| | | | | | | |
|---|---|---|---|---|---|---|
| stripes | 1 | 1 | 0 | 0 | 0 | 0 |
| horse-like shape | 1 | 1 | 1 | 1 | 0 | 0 |
| long neck | 0 | 0 | 0 | 0 | 1 | 1 |

36 Wikimedia Commons photographs (CC BY-SA / CC BY / public domain)

a person labels every concept on every image

the main cost, which newer methods try to reduce

# Modern concept bottleneck architectures



## One training record: an image, its concept labels and a class

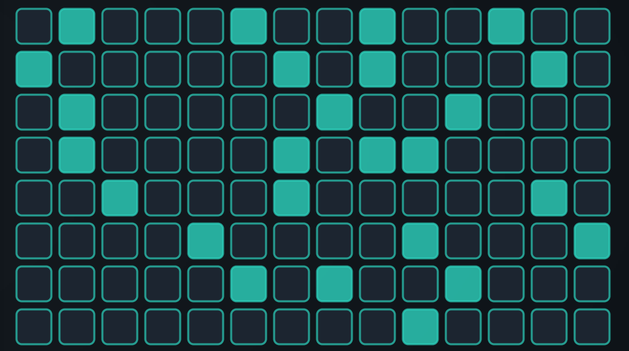


Photo: Diego Delso, Wikimedia Commons, CC BY-SA 4.0

```
{
  "image": "zebra_12.jpg",
  "concepts": {
    "stripes": 1,
    "horse_like_shape": 1,
    "long_neck": 0
  },
  "label": "zebra"
}
```

CONSTRUCTED EXAMPLE · a teaching record

CUB birds, as used by Koh et al. 2020



112 concept labels per photograph

DATASET · Herring Gull's labels in CUB

# Modern concept bottleneck architectures




## Rung one: images, concept labels and classes

rung 1 · task data

$$\{x,\ c,\ y\}$$

images, concept labels, **class labels**

**CBM, concept embedding model (CEM)**

rung 2

rung 3

less supervision

# Modern concept bottleneck architectures




## Rung two: keep class labels, drop per-image concept labels

rung 1 · task data

$\{x, c, y\}$

images, concept labels, **class labels**

**CBM, concept embedding model (CEM)**

drop

concept labels

rung 2 · task data

$\{x, y\}$

images and **class labels**

**post-hoc CBM**

rung 3

less supervision

# Modern concept bottleneck architectures




## Rung three: no training for this task

rung 1 · task data

$\{x, c, y\}$

images, concept labels, **class labels**

**CBM, concept embedding model (CEM)**

drop

concept labels

rung 2 · task data

$\{x, y\}$

images and **class labels**

**post-hoc CBM**

drop

task training

rung 3 · task data

$\varnothing$

no training for this task

**classification by description**

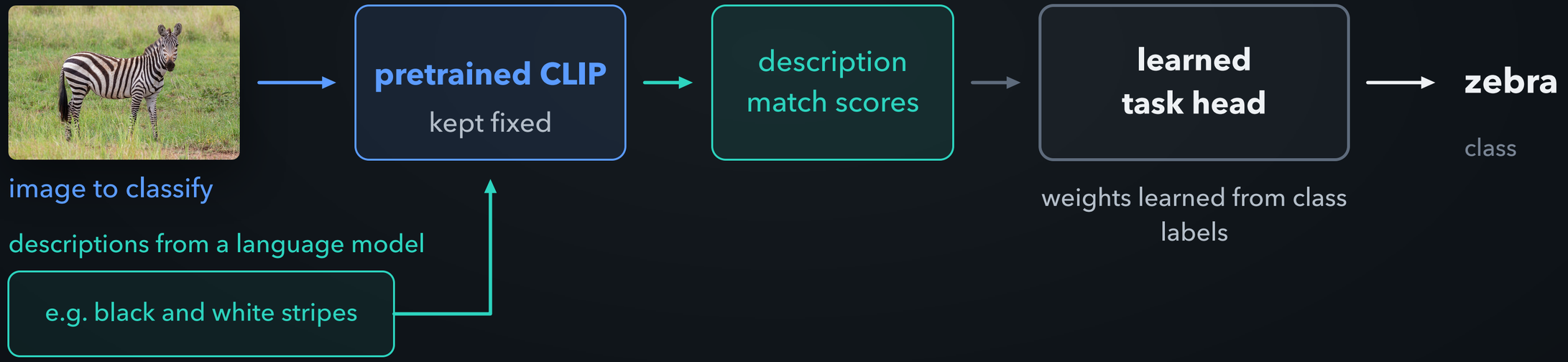
less supervision

# Modern concept bottleneck architectures

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Rung two: class labels teach the task head

Example: CLIP phrases + a learned head




Rung two: **class labels** teach us how to combine the scores.

`SCHEMATIC · example path; no measured scores shown`

Photo: Diego Delso, Wikimedia Commons, CC BY-SA 4.0

# Modern concept bottleneck architectures



## Rung three: combine the match scores with a fixed rule

Classification by description

no task-labelled examples
no fitting



image to classify

**pretrained CLIP**
kept fixed

description
match scores

**fixed**
**class averages**

**zebra**

highest average

average the description
scores within each class

descriptions from a language model

e.g. black and white stripes

Rung three: combine the scores with a **fixed average**.

`SCHEMATIC · example path; no measured scores shown`

Photo: Diego Delso, Wikimedia Commons, CC BY-SA 4.0

# Modern concept bottleneck architectures




## Fewer concept labels, so check the concepts another way

| | rung 1 | rung 2 | rung 3 |
|---|---|---|---|
| concept labels for each task image |  | none | none |
| someone checks each concept against the photo |  | ? | ? |

without per-image concept labels, check the concepts another way

# Modern concept bottleneck architectures

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## For each variant, ask: is the concept really in the image?




Photo: Diego Delso, Wikimedia Commons, CC BY-SA 4.0

who checks that a named concept is really visible in the image?

we return to this question in the recap and evaluation

`black and white stripes`

# Modern concept bottleneck architectures



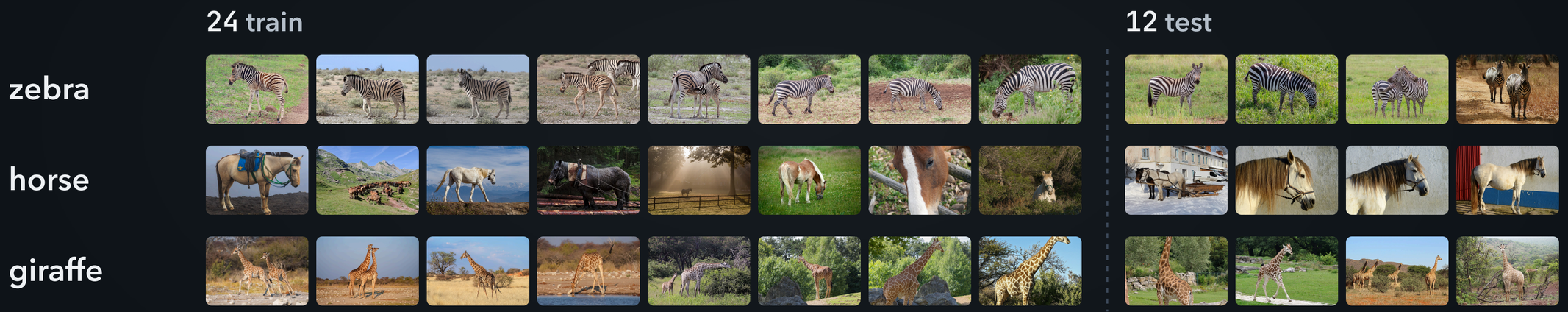

## A variant can change four things

| what changes | plain CBM |
|---|---|
| concept source | people label every concept |
| detector | a trained concept predictor |
| representation | one number per concept |
| what reaches the head | only the concept values |

# Modern concept bottleneck architectures




## One running example: photographs of zebras, horses and giraffes




36 Wikimedia Commons photographs (CC BY-SA / CC BY / public domain)

the three classes from Part 4

# Modern concept bottleneck architectures



## For each variant: how the path from image to concepts to class changes

$$x \longrightarrow \hat{c} \longrightarrow \hat{y}$$

| each variant, in symbols | | |
|---|---|---|
| concept predictor | $g_\theta$ | · · · |
| what reaches the head | $\hat{c}$ | · · · |
| task head | $f_\phi$ | · · · |
| what is trained | | · · · |
| data needed | | · · · |

PART 10 · RUNG ONE

# Concept labels and class labels

Next: Concept Embedding Models (CEMs)

Training uses an image, its concept labels and its class label.

**Concept and class labels** | Class labels | No task fitting

# Concept embedding models

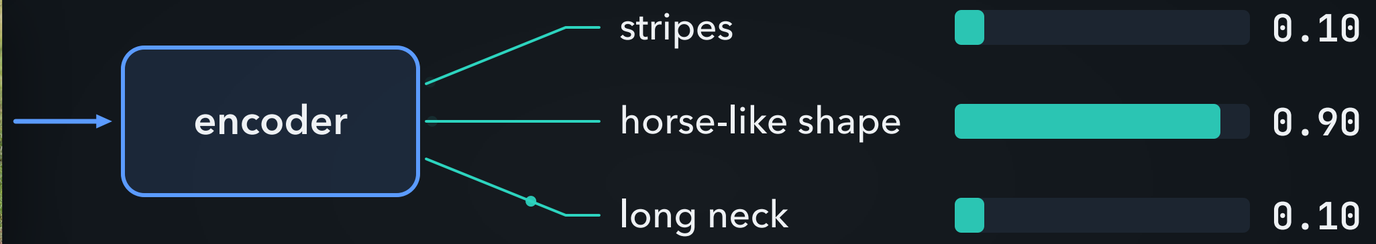


## One number per concept is a narrow channel

Concept Embedding Model (CEM)



Photo: Diego Delso, Wikimedia Commons, CC BY-SA 4.0




one number per concept: a narrow channel

Part 8: an incomplete vocabulary



`blue 1, large 0 for both`

same values, so best accuracy 0.50

ILLUSTRATIVE VALUES · hand-chosen values

# Concept embedding models




## What does a prototype mean here?

An embedding is a short list of learned numbers.

stripes present

$$\begin{bmatrix} 0.82 \\ 0.30 \end{bmatrix}$$

stripes absent

$$\begin{bmatrix} 0.22 \\ 0.74 \end{bmatrix}$$

Both vectors come from the current image.

These are not stored reference photos or class averages.

The coordinates are learned features, without individual names.

ILLUSTRATIVE VALUES · hand-chosen vector values

# Concept embedding models



## Learned layers turn the image into the two vectors

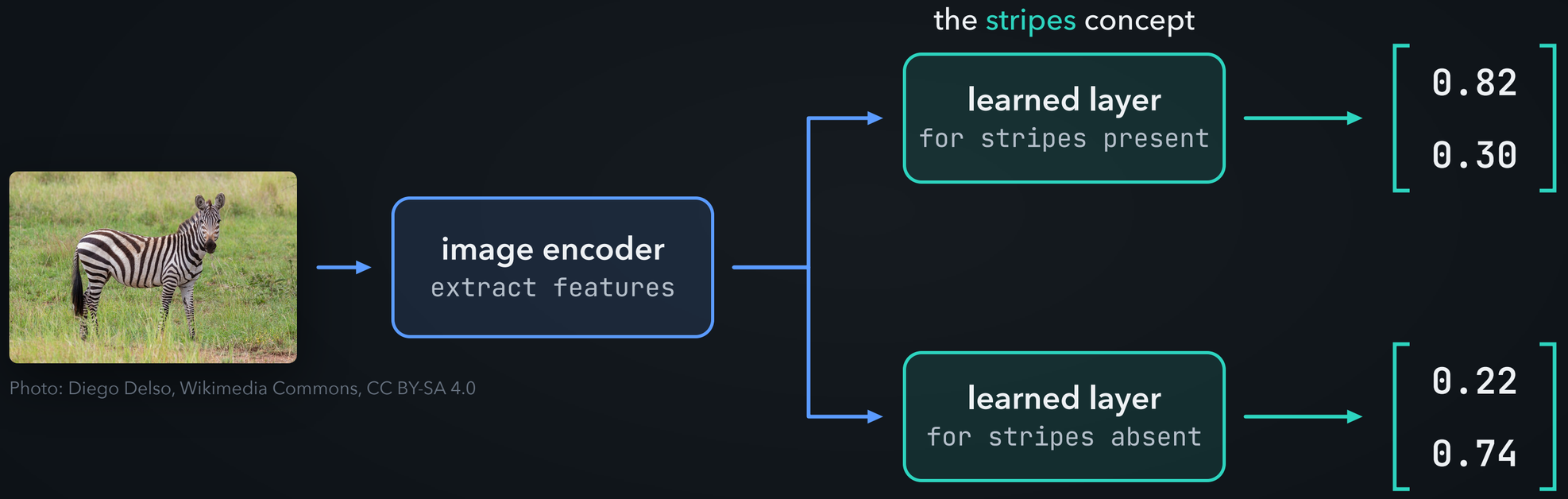



Photo: Diego Delso, Wikimedia Commons, CC BY-SA 4.0

Training learns the **weights** of these layers.

Keep the weights fixed: a new image can still produce different vectors.

ILLUSTRATIVE VALUES · hand-chosen outputs, not a trained run

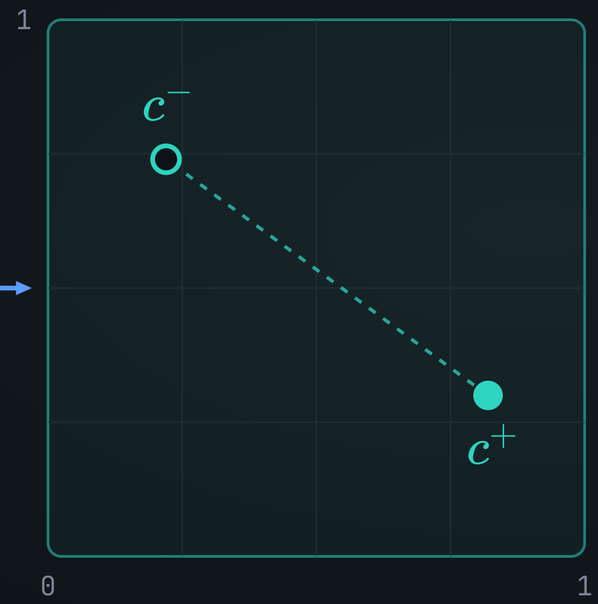
# Concept embedding models



## Each concept gets two embeddings: active and inactive



encoder




stripes: two embeddings in one plane

$$c^+ = \begin{bmatrix} 0.82 \\ 0.30 \end{bmatrix}$$ active

$$c^- = \begin{bmatrix} 0.22 \\ 0.74 \end{bmatrix}$$ inactive

one pair for every concept, computed from the image

embedding: a short list of numbers (here two)

ILLUSTRATIVE VALUES · hand-chosen 2-D embeddings

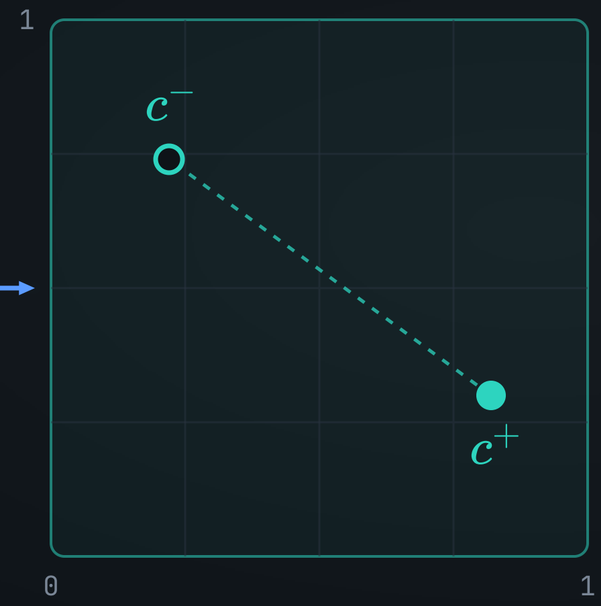
# Concept embedding models



## A learned scorer predicts the probability p



encoder




stripes: two embeddings in one plane

a learned scorer reads both vectors

and predicts the probability of stripes




$$p = 0.1$$

suppose it misses the stripes, as in Part 4

ILLUSTRATIVE VALUES · hand-chosen 2-D embeddings

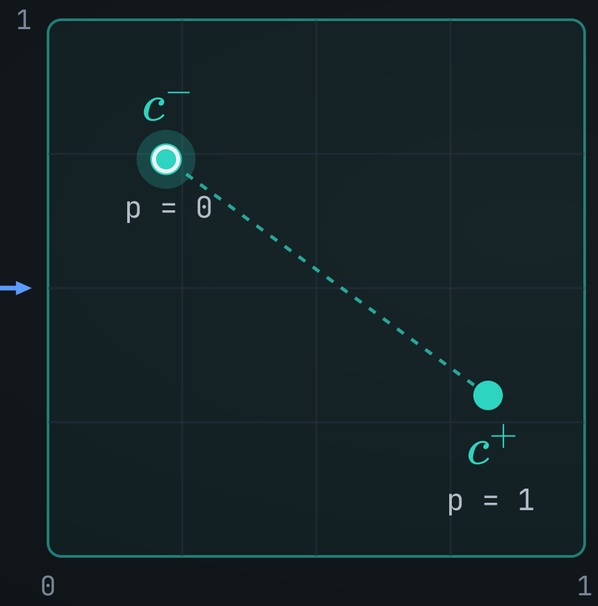
# Concept embedding models



## The concept embedding mixes the two, weighted by p



encoder




stripes: two embeddings in one plane

mixed = p × active + (1 − p) × inactive

$$\hat{c} = p\,c^+ + (1-p)\,c^-$$

the mixed point lies on the segment from inactive to active

ILLUSTRATIVE VALUES · hand-chosen 2-D embeddings

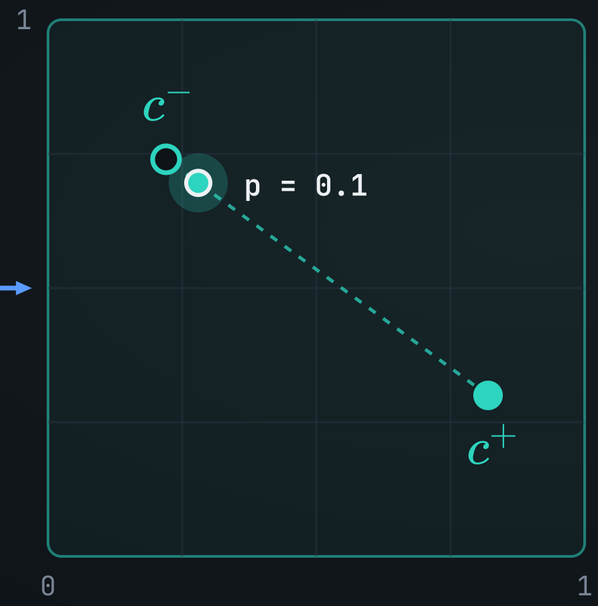
# Concept embedding models



## With p = 0.1, the mixed vector sits close to inactive



encoder




stripes: two embeddings in one plane

$\hat{c}$ for stripes, with p = 0.1:

first entry:

$$0.1 \times 0.82 + 0.9 \times 0.22 = 0.28$$

$$0.1 \begin{bmatrix} 0.82 \\ 0.30 \end{bmatrix} + 0.9 \begin{bmatrix} 0.22 \\ 0.74 \end{bmatrix} = \begin{bmatrix} 0.28 \\ 0.70 \end{bmatrix}$$

close to inactive

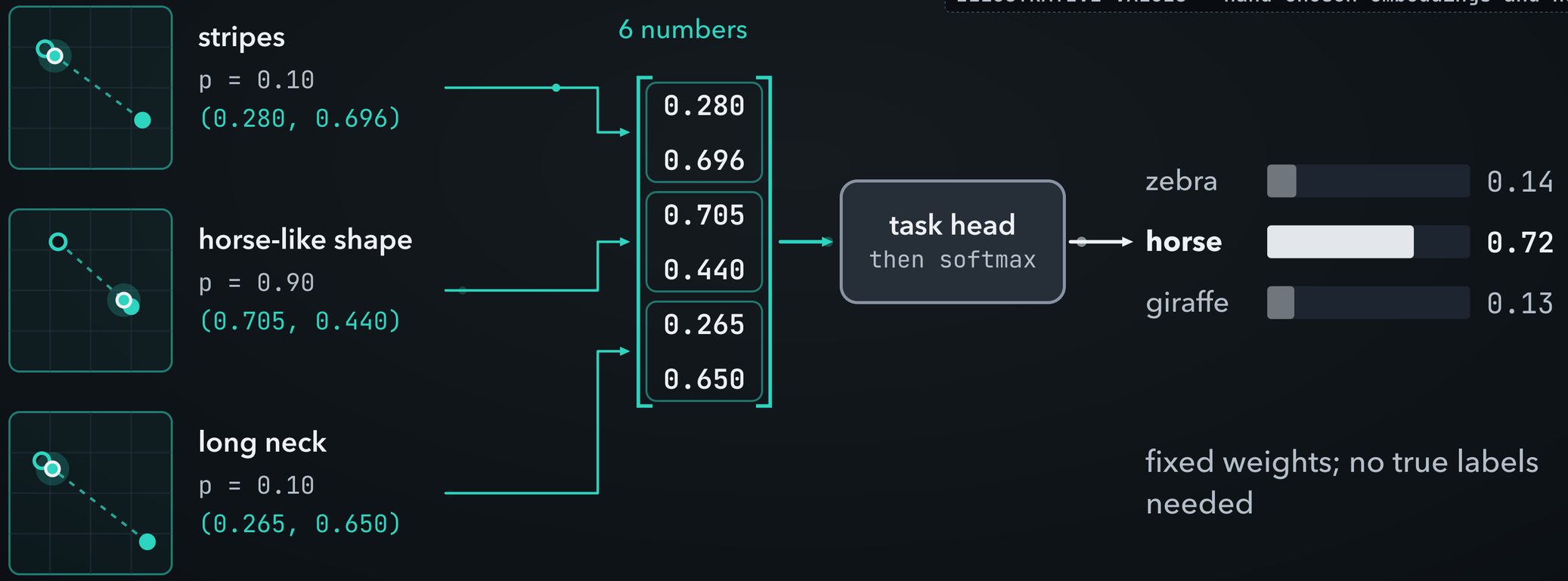
ILLUSTRATIVE VALUES · hand-chosen 2-D embeddings

# Concept embedding models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## At test time, the head reads the mixed vectors

ILLUSTRATIVE VALUES · hand-chosen embeddings and head



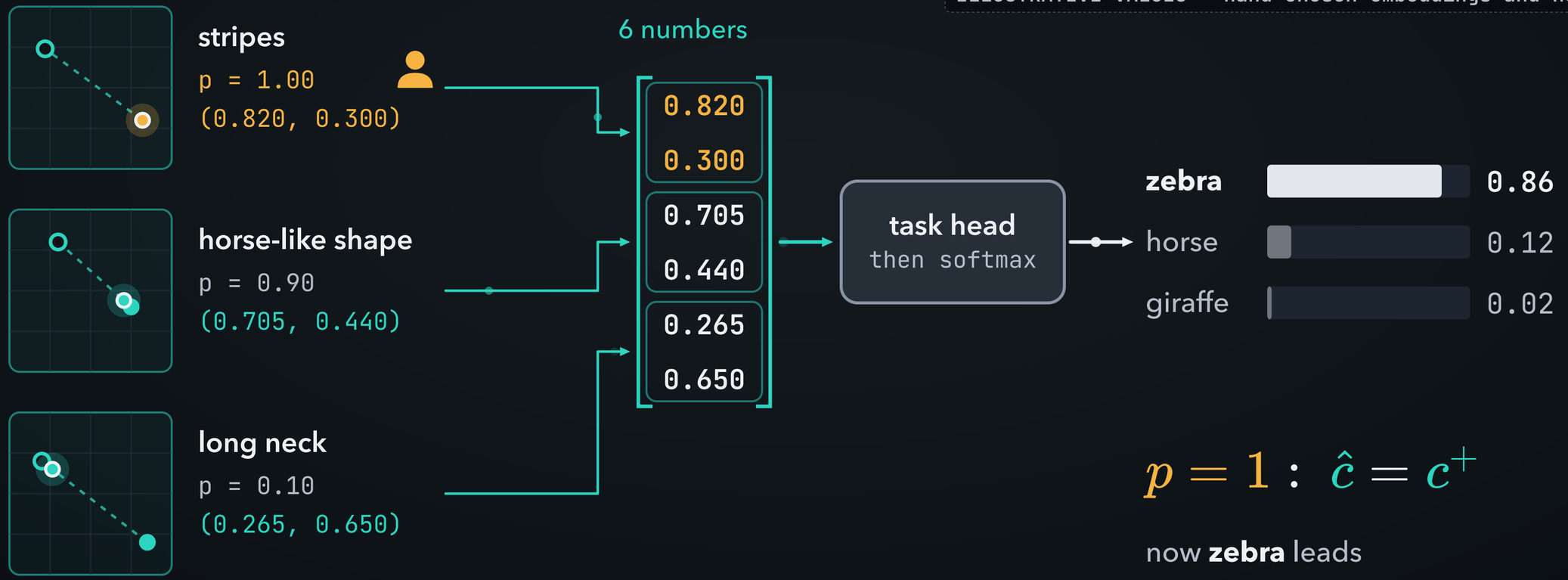

fixed weights; no true labels needed

# Concept embedding models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## An optional correction changes p, with weights fixed

ILLUSTRATIVE VALUES · hand-chosen embeddings and head




$p = 1 : \ \hat{c} = c^{+}$

now **zebra** leads

# Concept embedding models

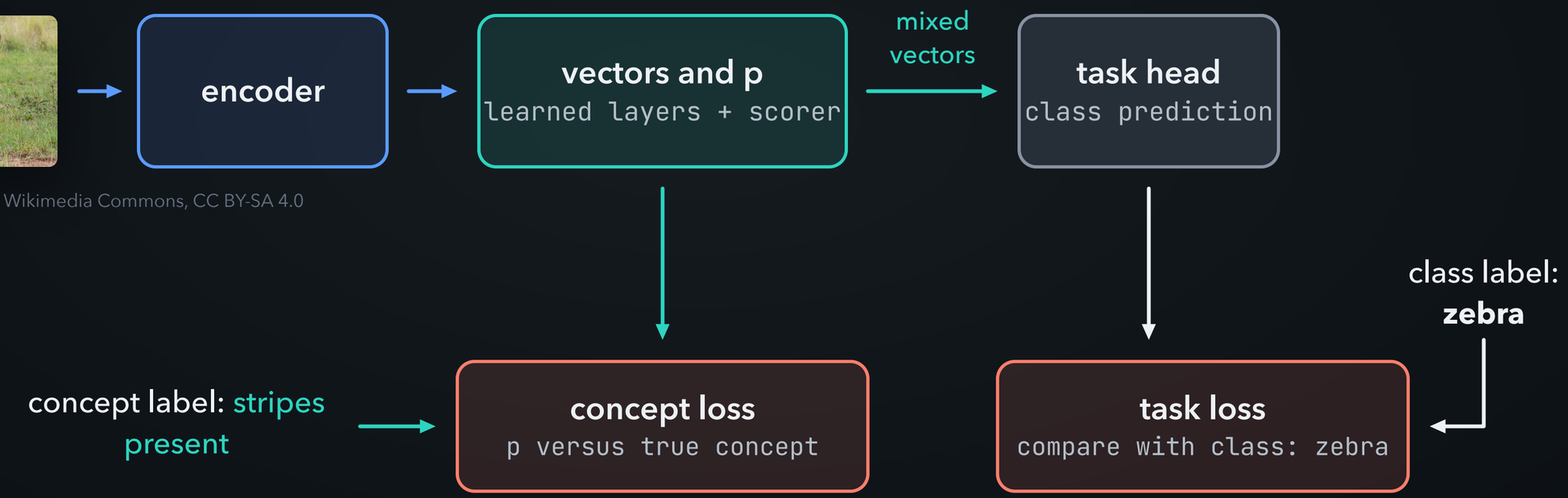


## Training uses concept labels and class labels

Training example: photo + concept labels + **class label**



Photo: Diego Delso, Wikimedia Commons, CC BY-SA 4.0




Backpropagation updates the encoder, vector layers, scorer and head.

# Concept embedding models




## Training learns the weights. Testing uses them

| what changes | training | test |
|---|---|---|
| **input** | image, concept labels, class label | new image |
| **network weights** | updated using both losses | kept fixed |
| **vectors and probabilities** | computed from this image | computed from the new image |

Test-time vectors are newly computed; the learned weights stay fixed.

# Concept embedding models




## Concept embedding models in symbols

**Concept embedding models (Espinosa Zarlenga et al., NeurIPS 2022)** x → ĉ → ŷ

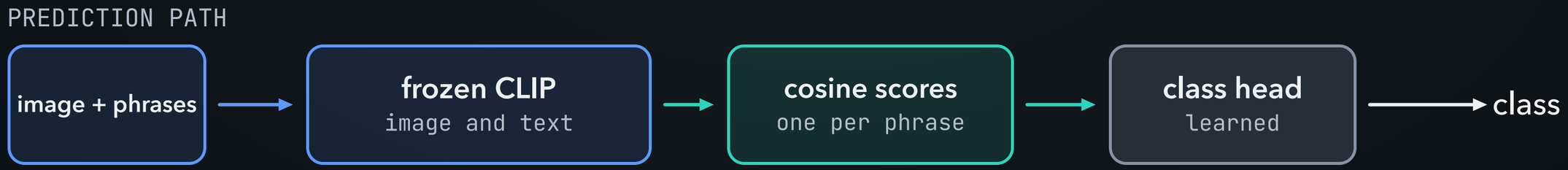
| | |
|---|---|
| concept predictor | $g_\theta : x \mapsto (p_j,\ c_j^+,\ c_j^-)$ |
| what reaches the head | $\hat{c}_j = p_j\, c_j^+ + (1 - p_j)\, c_j^- \in \mathbb{R}^m$ |
| task head | $\hat{y} = f_\phi(\hat{c}_1, \ldots, \hat{c}_k)$ |
| what is trained | $g_\theta,\ f_\phi$ (joint training) |
| data needed | $\{(x,\ c,\ y)\}$ images, concept labels, class labels |

$j$: one concept · $k$: number of concepts · $m$: numbers in each embedding

PART 10 · RUNG TWO

# Use CLIP scores with a learned head

Post-hoc concept bottleneck: the CLIP phrase example




Fit the class head using **class labels**; keep CLIP fixed.

Next: follow the phrase bank, image scores and class prediction.

SCHEMATIC · predict the class through named concepts

# Post-hoc concept bottleneck models

$x$ $g_\theta$ $\hat{c}$ $f_\phi$ $\hat{y}$

## Start with animal classes; ask for visible concepts

Post-hoc Concept Bottleneck Model (PCBM): the CLIP phrase route

Animal classes: zebra, horse, giraffe

**Ask a large language model (LLM)**

“Suggest visible properties that distinguish these animals. Avoid class names.”

→

black and white stripes

a long neck

a flowing mane

The LLM proposes words. CLIP will score the photos.

ILLUSTRATIVE VALUES · example prompt and possible suggestions; not a recorded LLM response

# Post-hoc concept bottleneck models




## Review the phrases, then fix one shared concept bank

Keep visible properties; reject class names, duplicates and weak matches.

“zebra” → reject: class name

“striped coat” → merge with “black and white stripes”

black and white stripes

a long neck

a flowing mane

brown patches on a pale coat

hooves

a saddle

green grass

a wooden fence

Every image is scored against this same list.

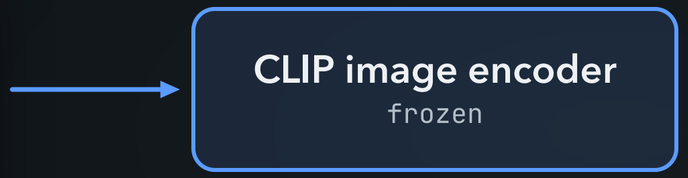
DATASET · saved bank: hand-chosen, not LLM-generated; review examples illustrative

# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## CLIP compares a photo with a concept phrase

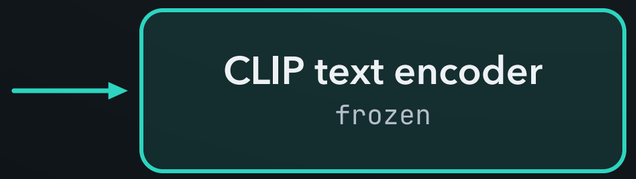














cosine(image vector, phrase vector) = 0.274

A match score, not a probability of stripes.

Photo: Diego Delso, Wikimedia Commons, CC BY-SA 4.0

COMPUTED · REAL MODEL · frozen CLIP ViT-B/32; saved photo and exact phrase

# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## One image gives one score per concept



image encoder
512 features

cosine similarity
with each phrase

| | |
|---|---|
| black and white stripes | 0.274 |
| long neck | 0.243 |
| flowing mane | 0.234 |
| brown patches on a pale coat | 0.257 |
| hooves | 0.243 |
| saddle | 0.223 |
| green grass | 0.217 |
| wooden fence | 0.211 |

8 named scores; not probabilities

Next: standardise scores, then train the class head.

Photo: Diego Delso, Wikimedia Commons, CC BY-SA 4.0

COMPUTED · REAL MODEL · frozen CLIP ViT-B/32; saved phrase vectors

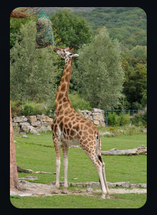
# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Use the same concept prompts for every image

Three held-out photos

raw cosine similarity: three prompts from the same bank

| | “black and white stripes” | “a long neck” | “a flowing mane” |
|---|---|---|---|
|  zebra | 0.274 | 0.243 | 0.234 |
|  horse | 0.168 | 0.199 | 0.186 |
|  giraffe | 0.208 | 0.301 | 0.218 |

Same encoder. Same prompts. One score per image and concept.

COMPUTED · REAL MODEL · frozen CLIP ViT-B/32; cosine scores, not probabilities

Commons photos: Diego Delso (CC BY-SA 4.0); George Chernilevsky (Public domain); Quartl (CC BY-SA 3.0)

# Post-hoc concept bottleneck models




## Each prompt gets its own training average and spread

Score all 24 training photos with the same prompts.

| | “black and white stripes” | “a long neck” | “a flowing mane” |
|---|---|---|---|
| training average | 0.215 | 0.239 | 0.227 |
| training spread<br>standard deviation | 0.042 | 0.037 | 0.023 |

Calculate down each prompt column, across training images.

The three test photos do not set the scale.

COMPUTED · REAL MODEL · saved training split; each prompt has its own mean and spread

# Post-hoc concept bottleneck models




## Standardise each score: subtract the mean, divide by the spread




standardised score = (score − training mean) ÷ training spread

$$z = \frac{0.274 - 0.215}{0.042} \approx 1.41$$

COMPUTED · REAL MODEL · stripes scores, mean and spread over the 24 training photos

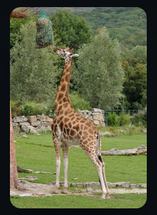
# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The same photos and prompts, now on a common scale

Three held-out photos

raw cosine → standardised score

| | “black and white stripes” | “a long neck” | “a flowing mane” |
|---|---|---|---|
|  zebra | 0.274 → **+1.41** | 0.243 → **+0.10** | 0.234 → **+0.32** |
|  horse | 0.168 → **-1.10** | 0.199 → **-1.08** | 0.186 → **-1.80** |
|  giraffe | 0.208 → **-0.16** | 0.301 → **+1.70** | 0.218 → **-0.38** |

0 = training average · positive = above · negative = below

Below average does not mean absent.

COMPUTED · REAL MODEL · training statistics reused; values rounded for display

Commons photos: Diego Delso (CC BY-SA 4.0); George Chernilevsky (Public domain); Quartl (CC BY-SA 3.0)

# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## For this zebra, stripes scores highest




each bar: (score − training mean) ÷ training spread

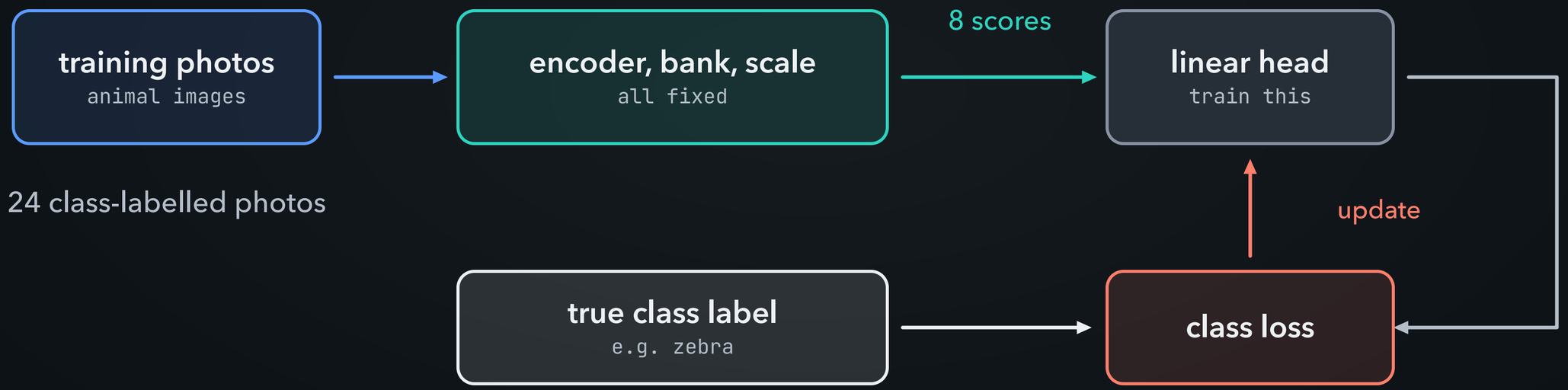
stripes: 1.41 spreads above its average

COMPUTED · REAL MODEL · CLIP ViT-B/32 on 36 Commons photos, teaching example

# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Train the task head using concept scores and class labels




Class labels teach the head which concepts predict each animal.

SCHEMATIC · animal example: CLIP phrase bank, then a sparse task head

# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Sparse: only 4 of 24 weights are non-zero

| | zebra | horse | giraffe |
|---|---|---|---|
| black and white stripes | +1.49 | 0 | 0 |
| long neck | 0 | -0.27 | +1.02 |
| flowing mane | 0 | +0.28 | 0 |
| brown patches on a pale coat | 0 | 0 | 0 |
| hooves | 0 | 0 | 0 |
| saddle | 0 | 0 | 0 |
| green grass | 0 | 0 | 0 |
| wooden fence | 0 | 0 | 0 |

8 concepts × 3 classes = 24 weights

**4 / 24**

weights are non-zero

L1 penalty: the size of every weight is added to the loss

COMPUTED · REAL MODEL · L1 logistic head fit on 24 photos

# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## The zebra score: stripes score times weight, plus a constant

stripes score × weight + constant = **zebra score**

$$1.406 \times 1.486 - 0.054 \approx 2.035$$




class scores

shares, after the softmax

COMPUTED · REAL MODEL · CLIP ViT-B/32 on 36 Commons photos

# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## At test time: one new image, with all weights fixed



encoder
512 features

cosine + scale
8 concept scores

trained head
class scores

softmax

Input: one image.

No concept labels. No class label.

Everything is saved. Nothing is fitted again.

| | | |
|---|---|---|
| **zebra** | | 0.77 |
| horse | | 0.11 |
| giraffe | | 0.11 |

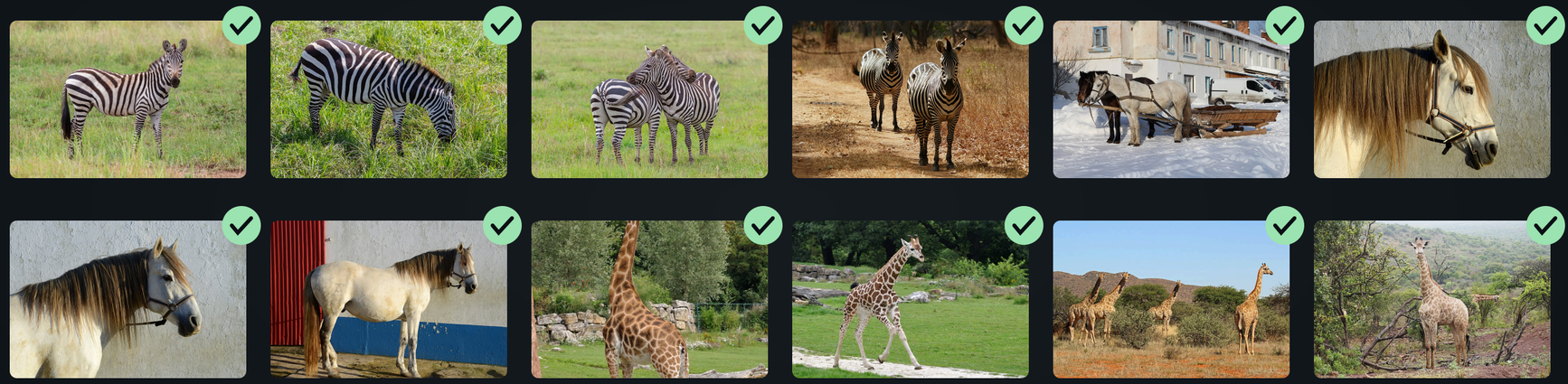
Photo: Diego Delso, Wikimedia Commons, CC BY-SA 4.0

COMPUTED · REAL MODEL · same held-out zebra; fixed encoder, bank, scale and head

# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## All 12 held-out photos are classified correctly



36 Wikimedia Commons photographs (CC BY-SA / CC BY / public domain)

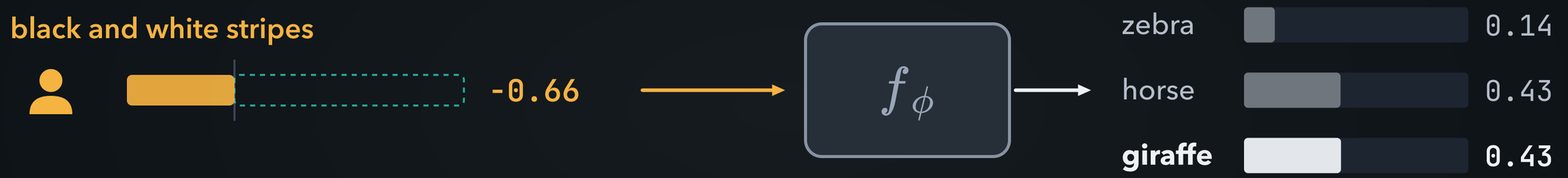
**12 of 12** held-out photos correct

a tiny teaching example, not a benchmark

# Post-hoc concept bottleneck models



## Edit the stripes score, and zebra falls




set to −0.66, its mean on non-zebra training photos

$$-0.659 \times 1.486 - 0.054 \approx -1.034$$

the new zebra logit; the photo and the head stay the same

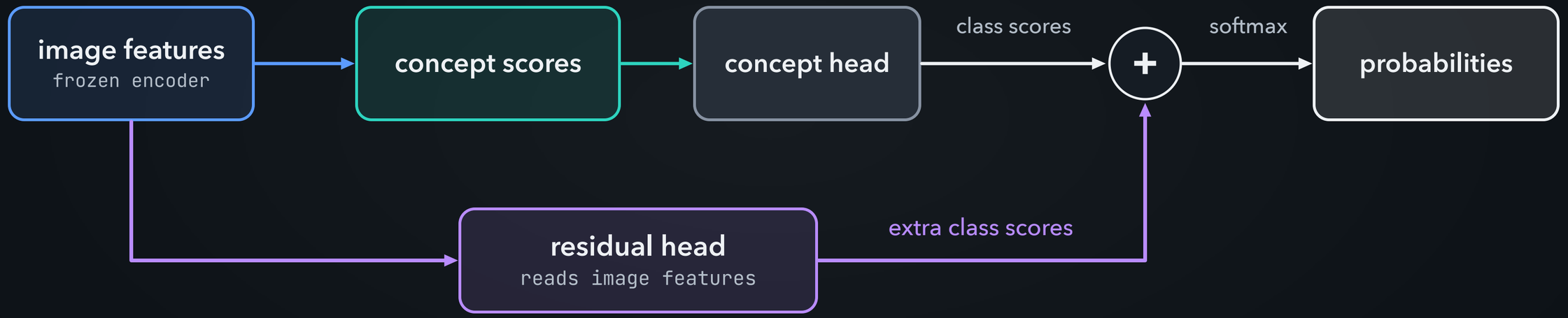
COMPUTED · REAL MODEL · the same head, one score edited

# Post-hoc concept bottleneck models

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## PCBM-h adds an extra path for evidence the concepts miss

Hybrid Post-hoc Concept Bottleneck Model (PCBM-h)




Worked photo: zebra class score, before softmax

$$2.03 + 0.31 \approx \mathbf{2.34}$$

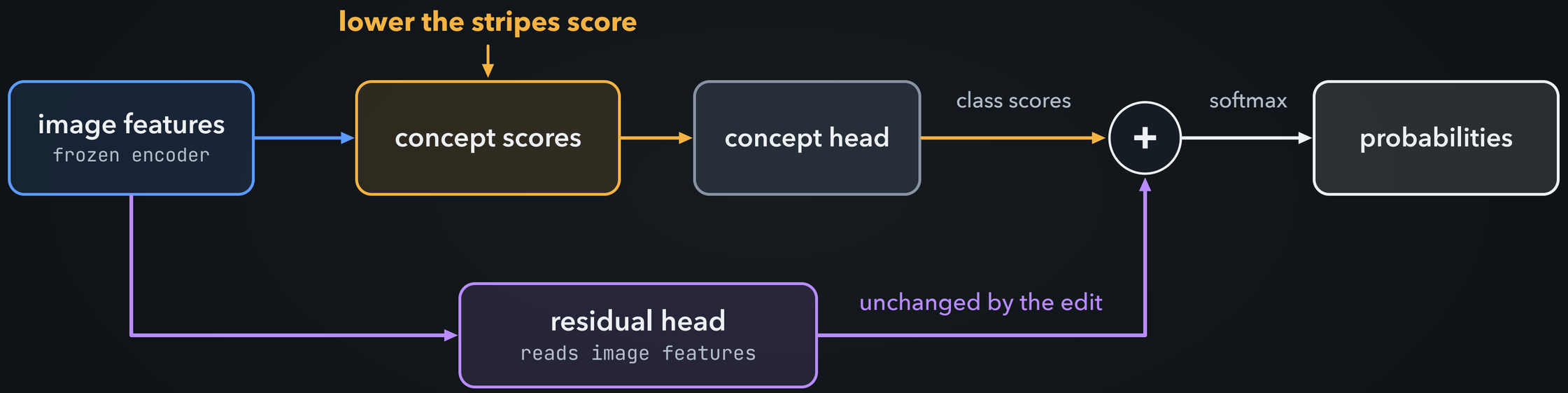
Train the extra head with class labels; keep the concept model fixed.

COMPUTED · REAL MODEL · saved CLIP animal example; scores rounded for display

# Post-hoc concept bottleneck models

$x$ $g_\theta$ $\hat{c}$ $f_\phi$ $\hat{y}$

## An edit leaves the extra path unchanged




Zebra probability after the same stripes edit

concept model: **0.14**

with residual: **0.20**

The prediction can change, but some evidence bypasses the concepts.

COMPUTED · REAL MODEL · same held-out zebra photo; same edit; residual contribution fixed

# Post-hoc concept bottleneck models




## Compared with the plain CBM: the concepts come from phrases

| what changes | plain CBM | post-hoc CBM |
|---|---|---|
| concept source | people label every concept | **a concept bank of phrases** |
| detector | a trained concept predictor | **frozen CLIP + cosine similarity** |
| representation | one number per concept | one number per concept |
| what reaches the head | only the concept values | **concept scores, plus a residual in PCBM-h** |

3 of 4 changed; each concept is still one number

# Post-hoc concept bottleneck models




## Post-hoc concept bottleneck models in symbols

**Post-hoc CBM (Yuksekgonul et al., ICLR 2023)** x → ĉ → ŷ

| | |
|---|---|
| concept predictor | $\hat{c}_j = \text{standardise}(\cos(e(x), v_j))$ |
| what reaches the head | $\hat{c} = (\hat{c}_1, \ldots, \hat{c}_k)$ standardised cosine scores |
| task head | $\hat{y} = f_\phi(\hat{c})$ sparse linear $+ r(e(x))$ PCBM-h |
| what is trained | $f_\phi$ only (and $r$ in PCBM-h) |
| data needed | $\{(x, y)\}$ + concept bank |

$e$: image vector · $v_j$: phrase vector · $r$: residual

PART 10 · RUNG THREE

# No fitting on the new task

Next: Classification by Description (CBD)

The pretrained model stays fixed; a fixed rule combines description scores.

Concept and class labels

Class labels

**No task fitting**

# Classification by description




## Rung three: no fitting on the new task

DATA EACH VARIANT NEEDS

$\{x, c, y\}$ CBM, CEM → $\{x, y\}$ PCBM → ∅ **by description**

# Classification by description (CBD)

Menon and Vondrick, ICLR 2023

no task concept labels, no task **class labels**, no fitting

# Classification by description



## Each class gets a checklist, and CLIP ticks it softly

photo



**Herring Gull**

| | |
|---|---|
| ☑ a white body with black wingtips | 0.9 |
| ☑ a yellow beak and legs | 0.8 |
| ☑ a red spot on its beak | 0.7 |
| average | 0.80 |

**Ivory Gull**

| | |
|---|---|
| ☑ a white bird | 0.8 |
| ☐ a black bill | 0.2 |
| ☐ pink legs | 0.3 |
| average | 0.43 |

**Ring-billed Gull**

| | |
|---|---|
| ☐ red eyes | 0.2 |
| ☑ webbed feet | 0.6 |
| ☑ a long, black tail | 0.4 |
| average | 0.40 |

best average: **wins**

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

ILLUSTRATIVE VALUES · real descriptors; tick strengths drawn for teaching

# Classification by description




## A language model describes every class

**language model**
writes descriptors

```
{ "Herring Gull": [
    "a large bird with a wingspan of up to 2.3 meters",
    "a white body with black wingtips",
    "a yellow beak and legs",
    "a red spot on its beak",
    "webbed feet"
]}
```

descriptor: a short phrase about what the **class** looks like

DATASET · Menon and Vondrick's released CUB descriptors

# Classification by description



## CLIP scores the photo against each descriptor



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

each score: a soft tick

- a large bird with a wingspan of up to 2.3 meters
- a white body with black wingtips
- a yellow beak and legs
- a red spot on its beak
- webbed feet

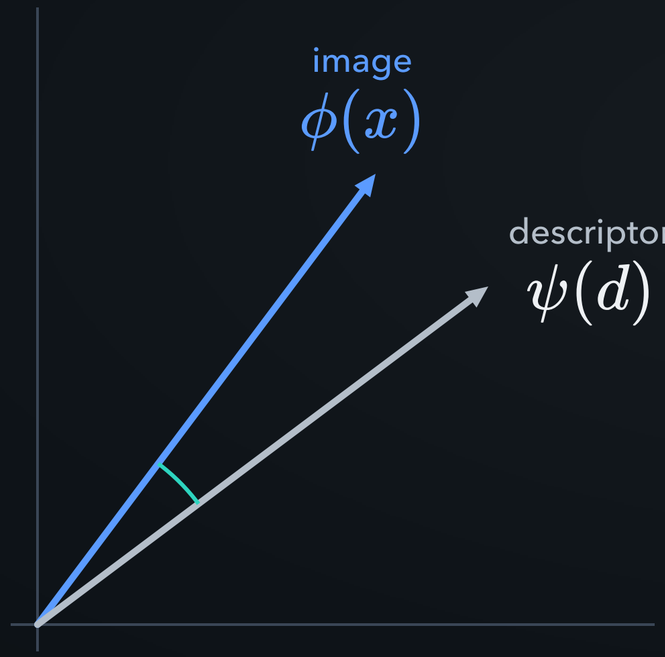
**class score**: the average

The **highest average** wins.

# Classification by description



## One descriptor score: a cosine between two vectors




entries (0.6, 0.8) and (0.8, 0.6)

$$\hat{c}_d = \cos\left(\phi(x),\ \psi(d)\right)$$
$$= 0.6 \times 0.8 + 0.8 \times 0.6 = 0.96$$

both vectors have length 1, so the cosine is the sum of products

near 1: they point the same way; smaller: they point apart

ILLUSTRATIVE VALUES · two-entry vectors for teaching; CLIP's have 512 entries

# Classification by description




## The class score averages its descriptor scores

**class score** — average over the class's descriptors — descriptor score

$$\hat{y}_k = \frac{1}{|D_k|} \sum_{d \in D_k} \hat{c}_d$$

$D_k$: the list of descriptors of class k, and $|D_k|$ how many there are

Nothing is fitted to the target data.

# Classification by description

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Every score comes with its reasons



bars start at 0.30




$$\hat{y}_{\text{Herring}} = \tfrac{1}{5}\big(0.324 + 0.339 + 0.339 + 0.340 + 0.328\big) = 0.334$$

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

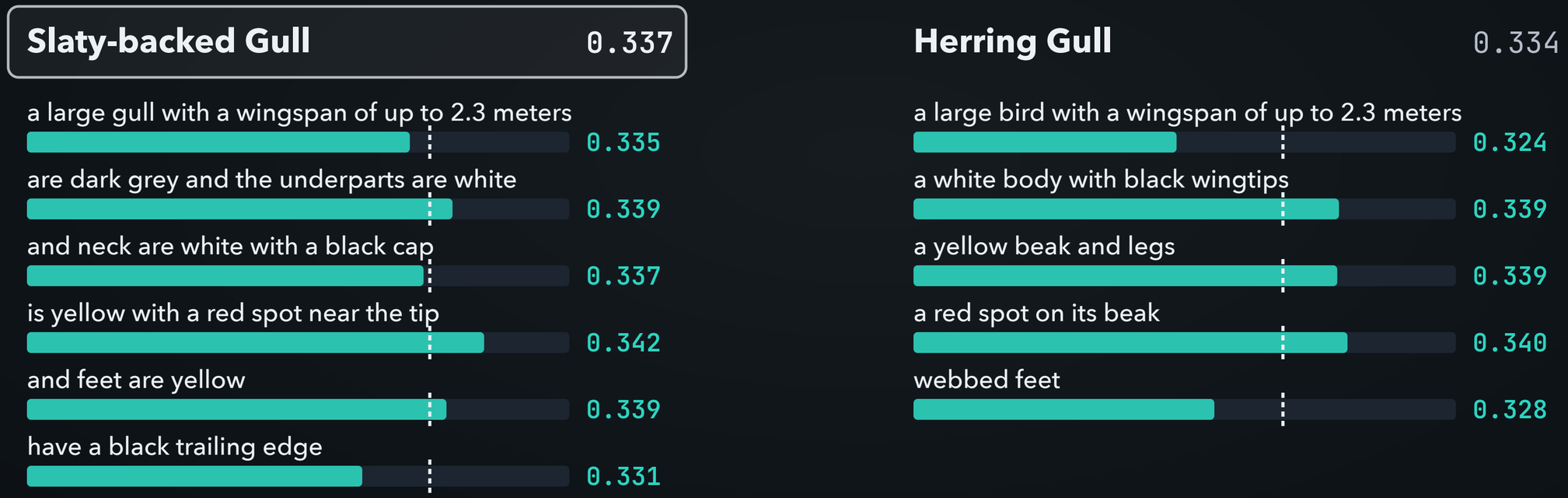
COMPUTED · REAL MODEL · CLIP ViT-B/32 on our gull photo, released CUB descriptors

# Classification by description

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Slaty-backed gull edges ahead

**class scores**: **Slaty-backed 0.337** · Herring 0.334 · Glaucous-winged 0.328 · Western 0.326 · California 0.324




**Slaty-backed Gull** edges ahead by 0.003

bars start at 0.30; dashed line: the average

Our photo shows a **herring gull**: a near miss.

COMPUTED · REAL MODEL · CLIP ViT-B/32 on our gull photo, released CUB descriptors

# Classification by description

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Attractive when there are no labels at all




no **labelled images**

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

classes we already score: Slaty-backed, Herring, Glaucous-winged, …

**+ one more class**

```
{ "Ivory Gull": [
    "a white bird",
    "black wingtips",
    "a black bill",
    "pink legs",
    "yellow eyes"
]}
```

its checklist is scored at once

**no retraining**

DATASET · Menon and Vondrick's released CUB descriptors

# Classification by description




## A descriptor can be wrong

The language model says:

```
{ "California Gull": [
    "a white bird with a black head and bill",
    ...
]}
```

**It does not have a black head.**

The wrong descriptor is still averaged into the score, one of 7.

DATASET · Menon and Vondrick's released CUB descriptors

# Classification by description



## A match score is not a fact check

Our herring gull photo



White head, yellow bill

From the California Gull checklist:

"a white bird with a black head and bill"

These words do not match this photo.

CLIP

Match score

**0.325**

Similarity, not a probability

California Gull

**Class average**

still includes this score

**Check the words**

Correct and visible in the photo?

**Check the score**

Does changing that feature change its score?

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

COMPUTED · REAL MODEL · CLIP ViT-B/32 on this photo; released CUB descriptor

# Classification by description




## From rung two to rung three: the head is no longer learned

| | rung two: learned head | | by description |
|---|---|---|---|
| the task head | a linear layer | → | **a fixed average** |
| what is trained | the head | → | **nothing** |
| data needed | images and **class labels** | → | **none** |

# Classification by description




## Classification by description in symbols

**Classification by description (Menon and Vondrick, ICLR 2023)** x → ĉ → ŷ

| | |
|---|---|
| concept predictor | $\text{CLIP: image encoder } \phi,\ \text{text encoder } \psi$ |
| what reaches the head | $\hat{c}_d = \cos\big(\phi(x),\ \psi(d)\big),\quad d \in D_k$ |
| task head | $\hat{y}_k = \frac{1}{\lvert D_k \rvert} \sum_{d \in D_k} \hat{c}_d \ \ \text{(fixed)}$ |
| what is trained | nothing |
| data needed | none: a language model writes the descriptors $D_k$ |

The cost: nothing guarantees that the descriptors are correct, or grounded.

# Comparing the variants

## One table: one row per variant, one column at a time

| variant | data needed | concepts from | what the head reads | trained |
|---|---|---|---|---|
| **CBM**<br>Koh et al. 2020 | | | | |
| **CEM**<br>Espinosa Zarlenga et al. 2022 | | | | |
| **PCBM**<br>Yuksekgonul et al. 2023 | | | | |
| **CBD** (by description)<br>Menon and Vondrick 2023 | | | | |

# Comparing the variants

## Column one: the supervision each variant needs

key: $x$ an image, $c$ its concept labels, $y$ its class label

| variant | data needed | concepts from | what the head reads | trained |
|---|---|---|---|---|
| **CBM**<br>Koh et al. 2020 | $\{(x, c, y)\}$ | | | |
| **CEM**<br>Espinosa Zarlenga et al. 2022 | $\{(x, c, y)\}$ | | | |
| **PCBM**<br>Yuksekgonul et al. 2023 | | | | |
| **CBD** (by description)<br>Menon and Vondrick 2023 | | | | |

# Comparing the variants

## Higher rungs drop the concept labels

key: $x$ an image, $c$ its concept labels, $y$ its class label

| variant | data needed | concepts from | what the head reads | trained |
|---|---|---|---|---|
| **CBM**<br>Koh et al. 2020 | $\{(x, c, y)\}$ | | | |
| **CEM**<br>Espinosa Zarlenga et al. 2022 | $\{(x, c, y)\}$ | | | |
| **PCBM**<br>Yuksekgonul et al. 2023 | $\{(x, y)\}$<br>+ a concept bank | | | |
| **CBD** (by description)<br>Menon and Vondrick 2023 | none for new task | | | |

# Comparing the variants

## Column two: where the concepts come from

key: $x$ an image, $c$ its concept labels, $y$ its class label

| variant | data needed | concepts from | what the head reads | trained |
|---|---|---|---|---|
| **CBM**<br>Koh et al. 2020 | $\{(x, c, y)\}$ | human labels | | |
| **CEM**<br>Espinosa Zarlenga et al. 2022 | $\{(x, c, y)\}$ | human labels | | |
| **PCBM**<br>Yuksekgonul et al. 2023 | $\{(x, y)\}$<br>+ a concept bank | a concept bank | | |
| **CBD** (by description)<br>Menon and Vondrick 2023 | none for new task | a language model | | |

# Comparing the variants

## Column three: what the task head reads

key: $x$ an image, $c$ its concept labels, $y$ its class label

| variant | data needed | concepts from | what the head reads | trained |
|---|---|---|---|---|
| **CBM**<br>Koh et al. 2020 | $\{(x,\ c,\ y)\}$ | human labels | $\hat{c}_j \in [0, 1]$<br>one number per concept | |
| **CEM**<br>Espinosa Zarlenga et al. 2022 | $\{(x,\ c,\ y)\}$ | human labels | $\hat{c}_j \in \mathbb{R}^m$<br>a vector per concept | |
| **PCBM**<br>Yuksekgonul et al. 2023 | $\{(x,\ y)\}$<br>+ a concept bank | a concept bank | scaled cosine scores<br>phrase example | |
| **CBD** (by description)<br>Menon and Vondrick 2023 | none for new task | a language model | $\cos(\phi(x),\ \psi(d))$<br>raw CLIP scores | |

# Comparing the variants

## Column four: what is trained

key: $x$ an image, $c$ its concept labels, $y$ its class label

| variant | data needed | concepts from | what the head reads | trained |
|---|---|---|---|---|
| **CBM**<br>Koh et al. 2020 | $\{(x,\ c,\ y)\}$ | human labels | $\hat{c}_j \in [0,1]$<br>one number per concept | $g_\theta,\ f_\phi$<br>predictor + head |
| **CEM**<br>Espinosa Zarlenga et al. 2022 | $\{(x,\ c,\ y)\}$ | human labels | $\hat{c}_j \in \mathbb{R}^m$<br>a vector per concept | $g_\theta,\ f_\phi$<br>predictor + head |
| **PCBM**<br>Yuksekgonul et al. 2023 | $\{(x,\ y)\}$<br>+ a concept bank | a concept bank | scaled cosine scores<br>phrase example | $f_\phi$<br>head only |
| **CBD** (by description)<br>Menon and Vondrick 2023 | none for new task | a language model | $\cos(\phi(x),\ \psi(d))$<br>raw CLIP scores | no task fitting |

# Comparing the variants

## Who supplies the concepts, and who checks them?

| variant | who supplies the concepts | who checks them |
|---|---|---|
| **CBM, CEM** | people label them |  **people** |
| **PCBM** | a concept bank | whoever built the bank |
| **CBD** | a language model | **review needed** |

# Comparing the variants

## Each rung up the ladder removes a cost and a check

$\{x,\ c,\ y\}$ CBM, CEM

$\{x,\ y\}$ PCBM

$\varnothing$ by description




Labour saved on labels has to be spent later, on checking.

SCHEMATIC · bar levels drawn for teaching

PART 11

# Evaluating a concept bottleneck

Check the concepts, predictions and interventions

# How should we evaluate a CBM?




## Four checks, beyond the final answer

| WHAT TO CHECK | THE QUESTION | LOOK AT |
|---|---|---|
| **Task accuracy** | Is the final answer right? | Held-out class labels |
| **Concept quality** | Are the named values right? | Concept labels + image evidence |
| **Faithfulness** | Does the answer use the concepts? | Inspect the path; test concept edits |
| **Intervention utility** | Do corrections help? | Accuracy after edits + human effort |

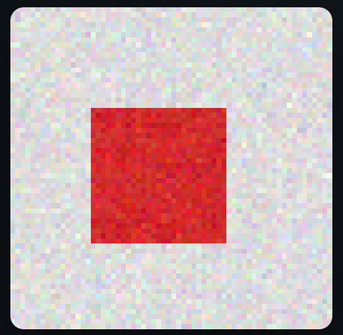
# How should we evaluate a CBM?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## One correction, the same trained head

Synthetic shapes: colour, shape and size determine the class.

test image



Truth: small red square

Before correction

large = 0.78 → same head → **large red square**

After: edit only large

large = 0 → same head → **small red square**

All other concept values stay the same. No retraining.

MEASURED · saved model output on one synthetic test image

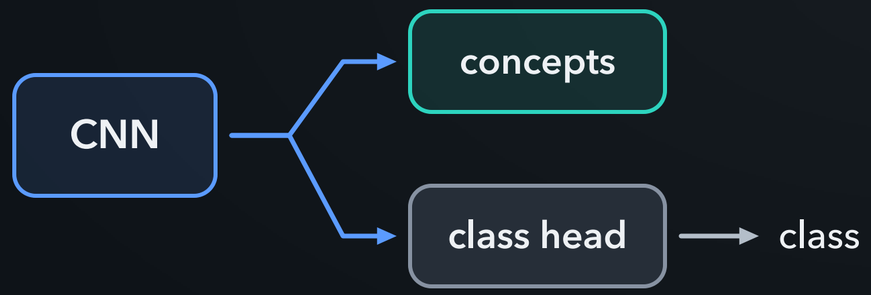
# How should we evaluate a CBM?

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Corrections help only if the head reads the concepts

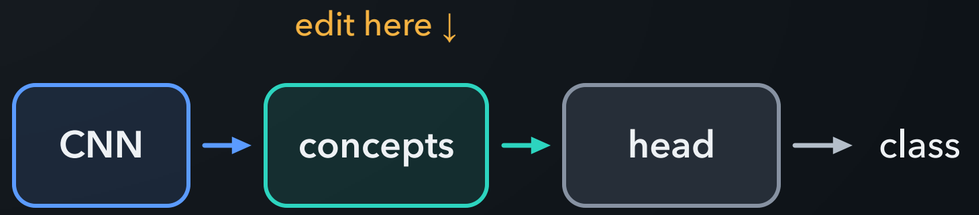
Task accuracy: before → after one concept correction per image

**Multi-task** **82.0% → 82.0%**




No path from concepts to class

**Strict CBM** **81.6% → 97.3%**




The correction reaches the class head

Oracle correction: true labels fix the largest concept error first.

Perfect corrections; human time and mistakes were not measured.

MEASURED · 1 seed · 512 synthetic test images · saved model outputs

PART 12

# Beyond images

The same idea with different inputs

# Beyond images



## Where else do concept bottlenecks fit?

SO FAR: PHOTOGRAPHS OF BIRDS



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

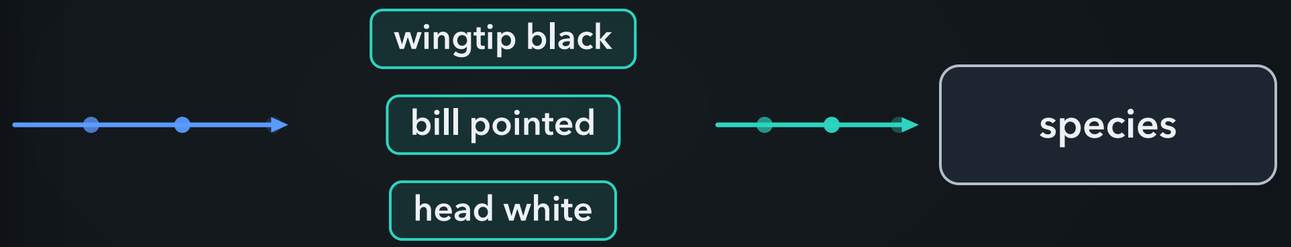



photo $x$ | concepts $\hat{\mathbf{c}}$ | **species** $\hat{y}$

they fit wherever experts reason with named findings

# Beyond images



## Medicine: a knee X-ray → clinical findings → a severity grade

input $x$



named concepts $\hat{\mathbf{c}}$

each finding is a small grade, 0 = none

| | |
|---|---|
| joint space narrowing | 1 |
| bone spurs | 2 |
| sclerosis | 1 |
| 7 more findings | ⋮ |

task $\hat{y}$

**osteoarthritis grade**

$$\hat{y} \in \{0, 1, 2, 3\}$$

$$\hat{y} = 2$$

0 none or doubtful · 3 severe

Knee X-ray: Koh et al. 2020, Fig. 1 (CC BY 4.0), Osteoarthritis Initiative

Koh et al., ICML 2020 · Osteoarthritis Initiative

ILLUSTRATIVE VALUES · values for teaching

# Beyond images



## A radiologist corrects a finding, and the grade is recomputed

input $x$



Knee X-ray: Koh et al. 2020, Fig. 1 (CC BY 4.0), Osteoarthritis Initiative

named concepts $\hat{\mathbf{c}}$

each finding is a small grade, 0 = none

| | |
|---|---|
| joint space narrowing | 2 |
| bone spurs | 2 |
| sclerosis | 1 |
| 7 more findings | ⋮ |

radiologist corrects a finding

task $\hat{y}$

**osteoarthritis grade**

$$\hat{y} \in \{0, 1, 2, 3\}$$

$$\hat{y} = 3$$

0 none or doubtful · 3 severe

Why it fits: radiologists already write their reports in these terms

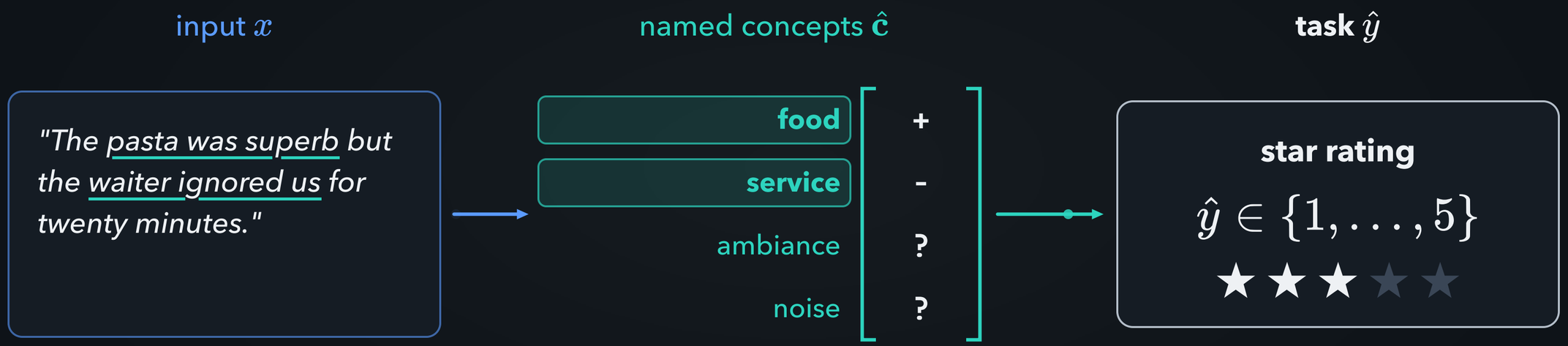
Koh et al., ICML 2020 · Osteoarthritis Initiative

ILLUSTRATIVE VALUES · values for teaching

# Beyond images



## Language: a review → four aspects → a star rating




+ praised · – criticised · ? not mentioned

Why it fits: reviewers already write about the food and the service

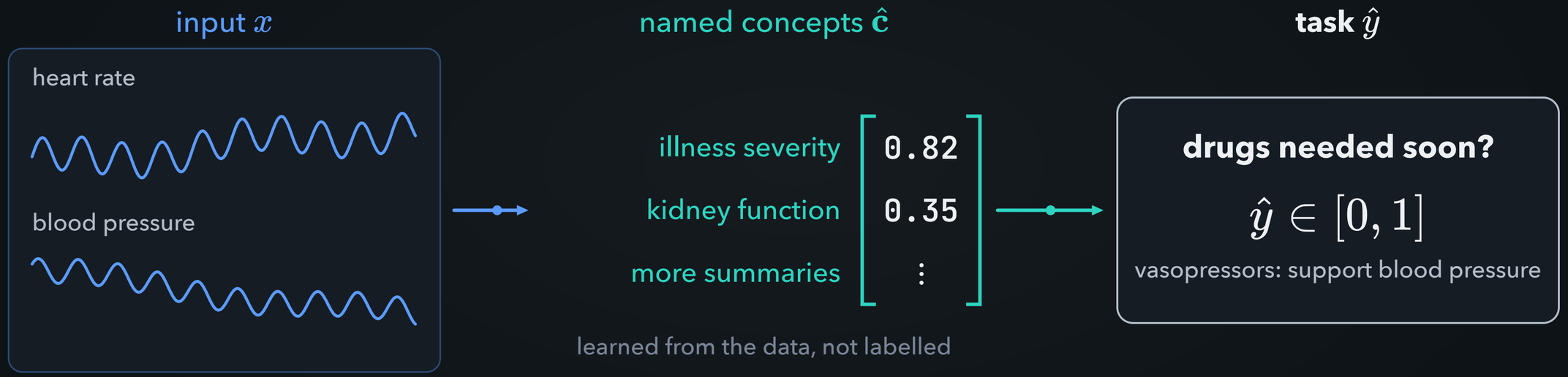
CEBaB, Abraham et al., NeurIPS 2022 · text bottleneck models, Ludan et al. 2023

`ILLUSTRATIVE VALUES · review and values for teaching`

# Beyond images



## Intensive care: vital signs → summary concepts → drugs needed soon?




Why it fits: doctors already sum up a patient in terms like illness severity and kidney function

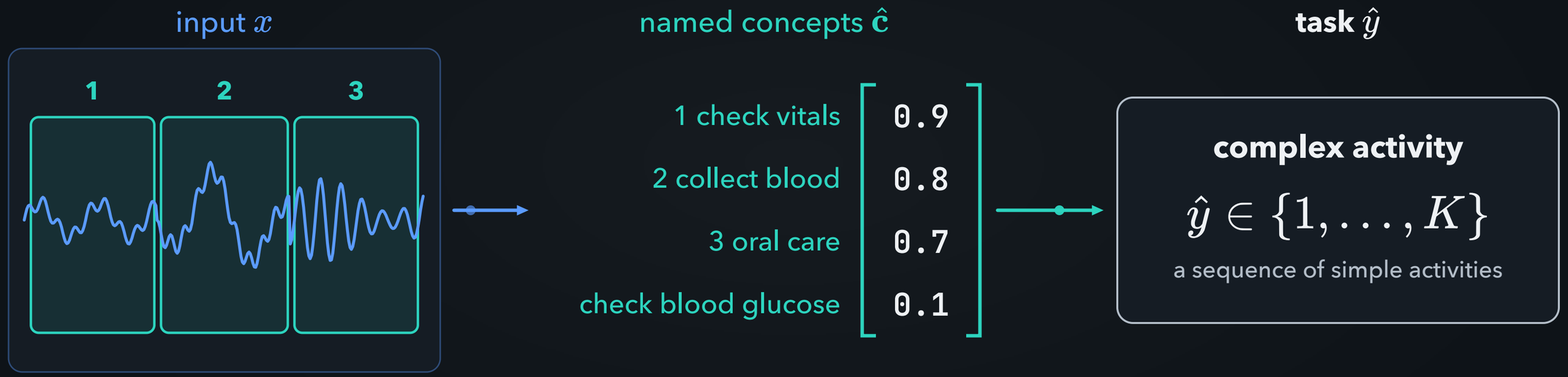
Wu et al., MLHC 2022 · MIMIC-III, 15,552 patients

ILLUSTRATIVE VALUES · schematic traces, values for teaching

# Beyond images



## Wearables: sensor signal → simple activities → complex activity




Why it fits: people already describe a **complex activity** as its simple steps

X-CHAR, Jeyakumar et al., IMWUT 2023 · concepts from its nurse data

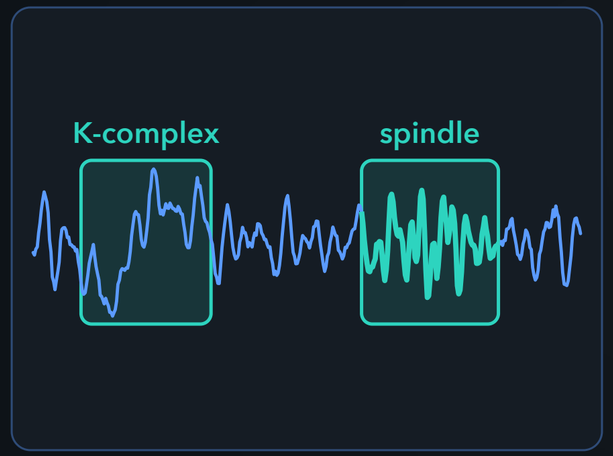
ILLUSTRATIVE VALUES · schematic trace, values for teaching

# Beyond images



## Sleep: an EEG → the patterns scorers use → a sleep stage

input $x$




named concepts $\hat{\mathbf{c}}$

| | |
|---|---|
| spindle | 1.00 |
| K-complex | 0.80 |
| slow waves | 0.15 |

**task $\hat{y}$**

**sleep stage**

$\hat{y} \in \{\mathrm{W}, \mathrm{N1}, \mathrm{N2}, \mathrm{N3}, \mathrm{R}\}$

N2 candidate: **0.86**

W wake · N1, N2, N3: stages one to three · R: REM

Why it fits: scorers already mark these patterns by hand

patterns from the AASM scoring manual (Berry et al.)

ILLUSTRATIVE VALUES · schematic EEG, the sleep head from Part 6

# Beyond images




## One recipe in every domain

**Medicine**
knee X-ray → grade

**Language**
review → stars

**Intensive care**
vital signs → drug need

**Wearables**
sensor signal → activity

**Sleep**
EEG → sleep stage

THE RECIPE

name the findings an expert checks

$$c_1, \ldots, c_k$$

predict them from the raw signal

$$\hat{\mathbf{c}} = g_\theta(x)$$

let the **decision** use them alone

$$\hat{y} = f_\phi(\hat{\mathbf{c}})$$

PART 13

# What can still go wrong?

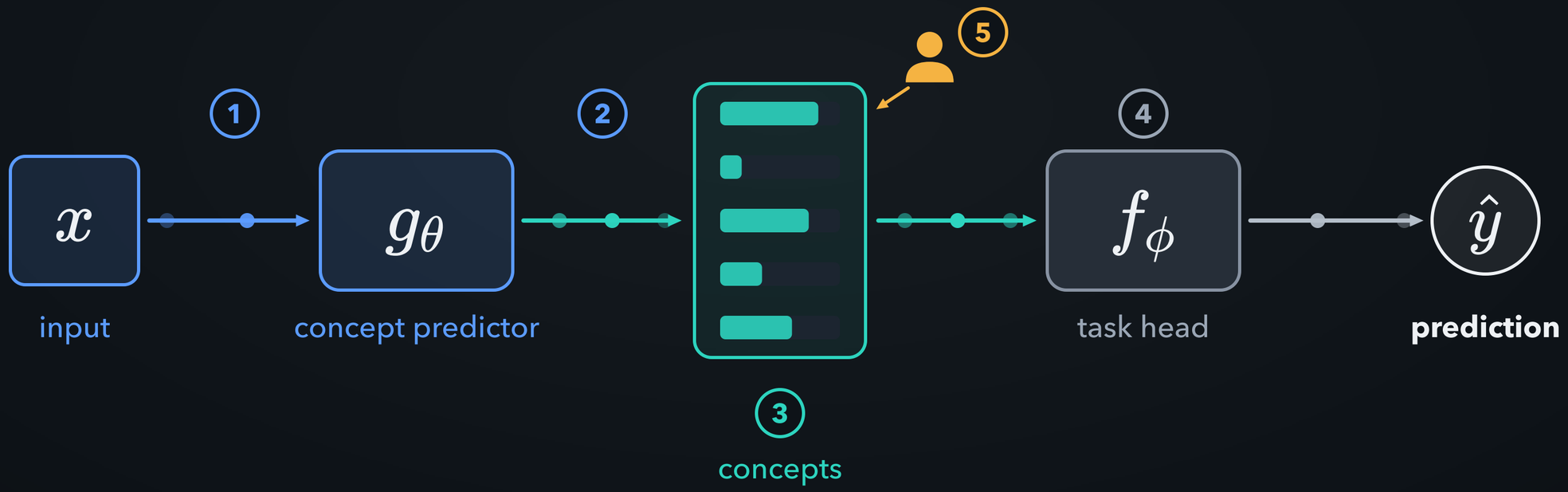
An editable interface needs careful checks

PART 13

# An interpretable interface is not the end of the story

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Follow the path from photo to concepts to answer: what can fail?



# An interpretable interface is not the end of the story

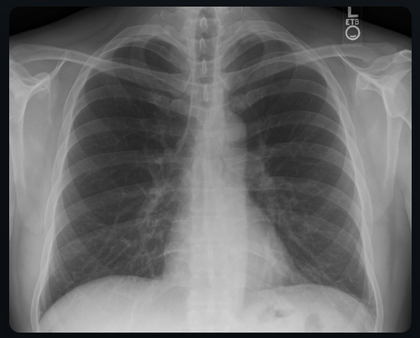



## 1 · Encoder: distribution shift




new inputs unlike the training inputs

$$p_{\text{new}}(x) \neq p_{\text{train}}(x)$$



Chest X-ray: Stillwaterising, Wikimedia Commons, CC0

# An interpretable interface is not the end of the story

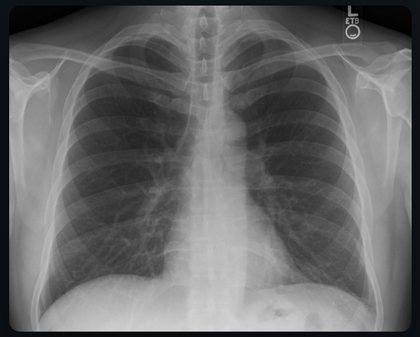



## A new hospital: the same model scores lower




new inputs unlike the training inputs

$$p_{\text{new}}(x) \neq p_{\text{train}}(x)$$



Chest X-ray: Stillwaterising, Wikimedia Commons, CC0

pneumonia detection, AUC (1 = perfect ranking, 0.5 = guessing)

| | |
|---|---|
| same hospitals | 0.931 |
| new hospital | 0.815 |

In practice: fine where it was built, worse at the new site

REPORTED · Zech et al., PLOS Medicine 2018 (Part 1)

# An interpretable interface is not the end of the story



## 2 · Concept detector: wrong grounding



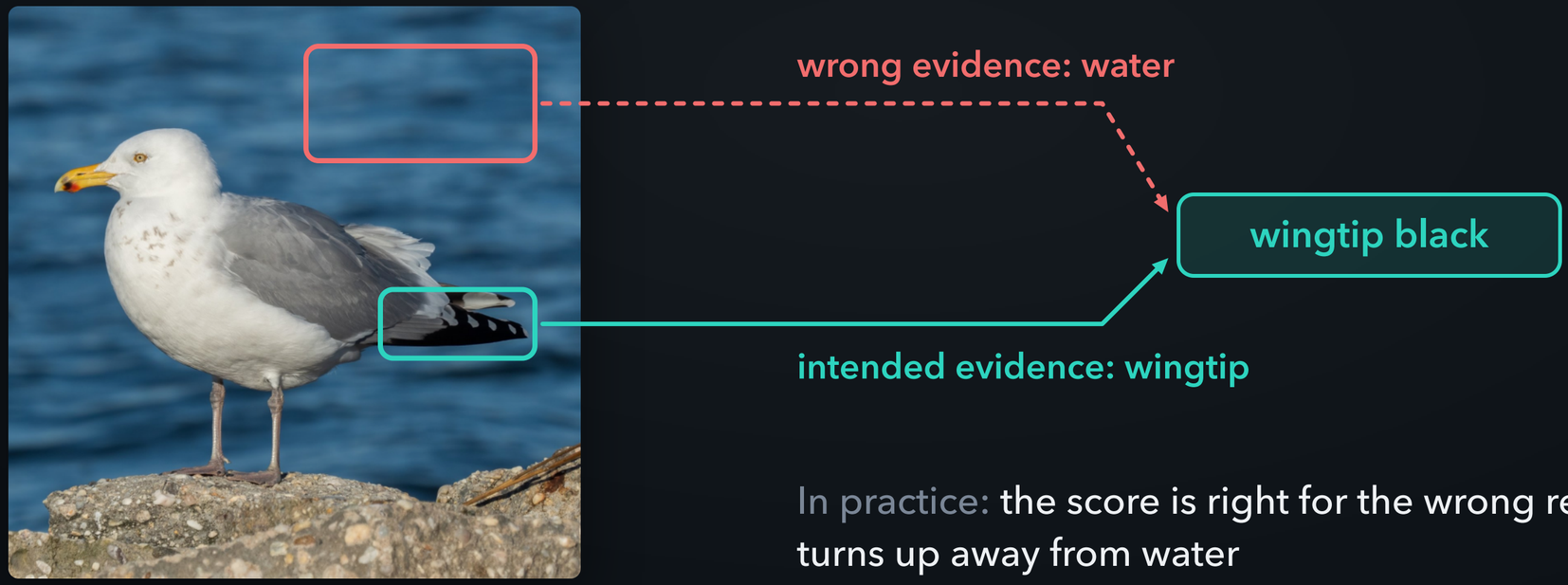




Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

In practice: the score is right for the wrong reason, until a gull turns up away from water

SCHEMATIC · regions drawn by hand

# An interpretable interface is not the end of the story




## 3 · Vocabulary: missing information







the same concept vector, different **classes**

$$\mathbf{c}(x_1) = \mathbf{c}(x_2), \quad y_1 \neq y_2$$

shape never reaches the head: best accuracy **50%**

In practice: two cases an expert tells apart look identical to the head

CONSTRUCTED EXAMPLE · the blue circle and square of Part 8

# An interpretable interface is not the end of the story




## 4 · Task head: the wrong relationship




$$\text{score} = 1.2 - 2\,\text{duplicate charge} + 0.6\,\text{refund requested}$$




a correct edit meets a wrong rule: billing falls from 0.80 to 0.45

In practice: the expert's fix is right, yet the decision moves the wrong way

CONSTRUCTED EXAMPLE · head built faulty on purpose, Part 7

# An interpretable interface is not the end of the story




## 5 · Intervention: wrong, or too costly

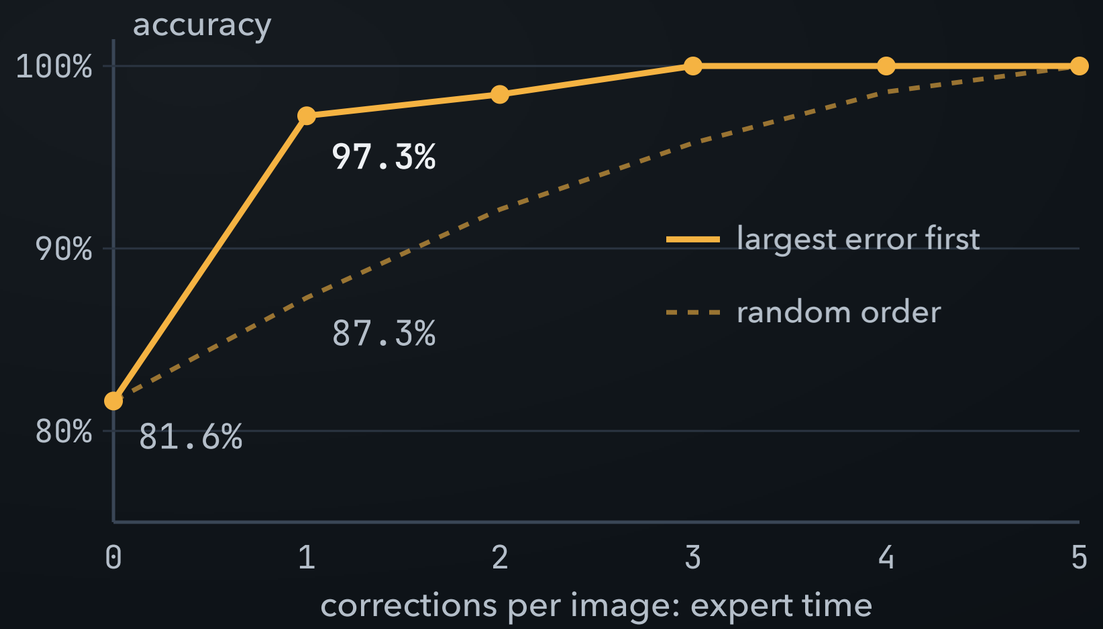





size: large 0.00

Part 11 took corrections from true labels: an expert who never errs

In practice: real experts misjudge too, and every check costs their time




MEASURED · Part 11: one seed, 512 synthetic test images, oracle corrections

# An interpretable interface is not the end of the story




## Five questions for any concept bottleneck

1. ☐ Are the concepts meaningful?
2. ☐ Are they grounded correctly?
3. ☐ Are they sufficient for the task?
4. ☐ Does the model actually use them?
5. ☐ Do corrections improve decisions?

# An interpretable interface is not the end of the story




## A 2026 survey maps the field onto the same parts

SURVEY · TMLR 2026

### What's in the Bottle? A Survey and Roadmap of Concept Bottleneck Models

P. Knab, D. Steinmann, C. Bartelt, K. Kersting, B. Schiele, T. Seidl, U. Schlegel, W. Stammer

input | concepts | output | training

open challenges include grounding, completeness, leakage and intervention reliability

RECAP

# Putting the pieces together

From the image to an editable explanation

# Final replay

## One last time: a black box maps the photo to a species






Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

ILLUSTRATIVE VALUES · hand-written concept values and head

# Final replay

## Open it: named concepts sit on the path to the prediction

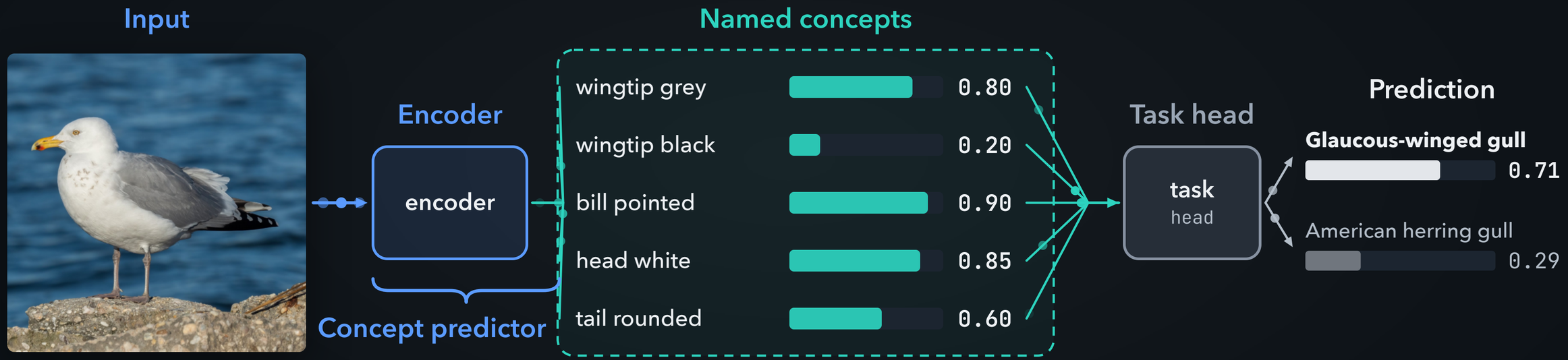



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

ILLUSTRATIVE VALUES · hand-written concept values and head

# Final replay

## A person corrects one concept, and the prediction follows

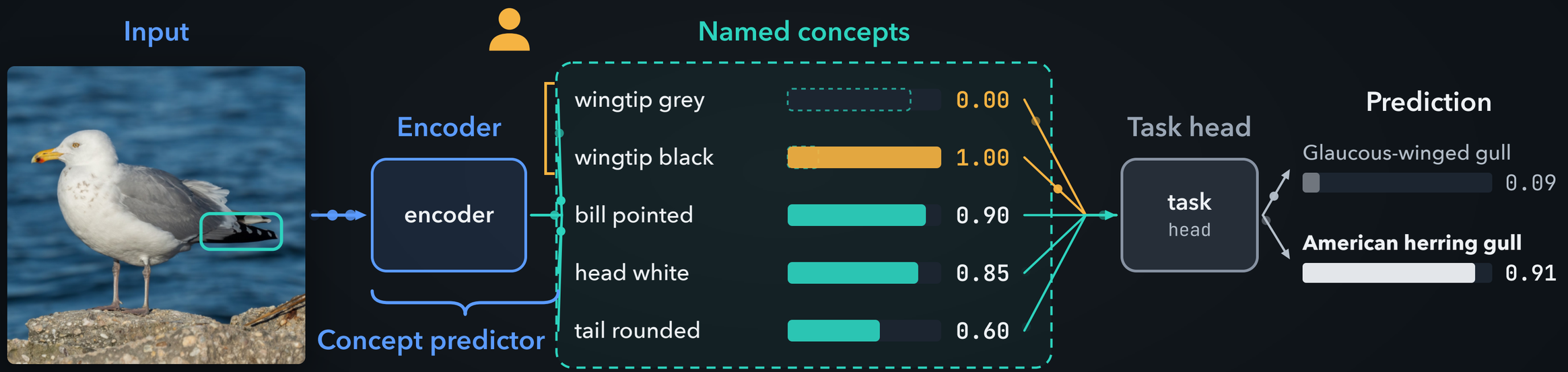



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

ILLUSTRATIVE VALUES · hand-written concept values and head

# Final replay

## The interface is only as good as its parts

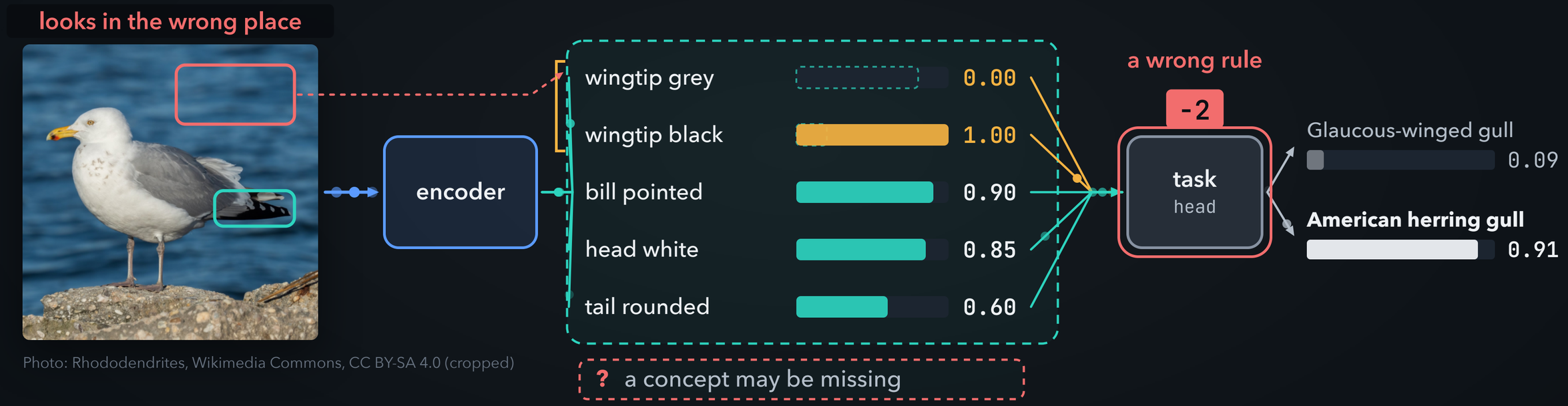



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

ILLUSTRATIVE VALUES · hand-written concept values and head

# Final replay

## Concepts that can be inspected, evaluated and corrected




Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

inspected evaluated corrected

ILLUSTRATIVE VALUES · hand-written concept values and head

## Final replay

**The interface must carry what the prediction depends on**

The challenge is to make sure the interface faithfully represents the information on which the prediction depends.

put simply: whatever the **answer** uses should be visible in the concepts

# Final replay

## What to remember

1. Post-hoc methods explain afterwards; a concept bottleneck makes the **answer** pass through named concepts
2. Fixing a concept helps only if the head reads it the right way
3. A soft score can carry more than its name says: extra information leaks
4. Newer variants need fewer labels, so check each concept looks at the right evidence

APPENDIX

# Testing concept use in a black box

Testing with Concept Activation Vectors (TCAV)

# Appendix: testing with concept activation vectors

## TCAV asks about a named concept: does the model use grey wings when it says herring gull?




Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

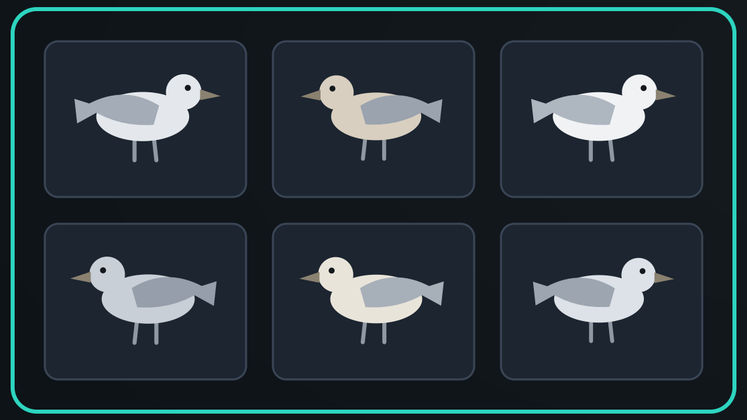
one more post-hoc method: it asks about concepts directly

TCAV: Kim et al., ICML 2018

# Appendix: testing with concept activation vectors

## Step 1: collect photos with grey wings and photos without

**with grey wings**



**without**



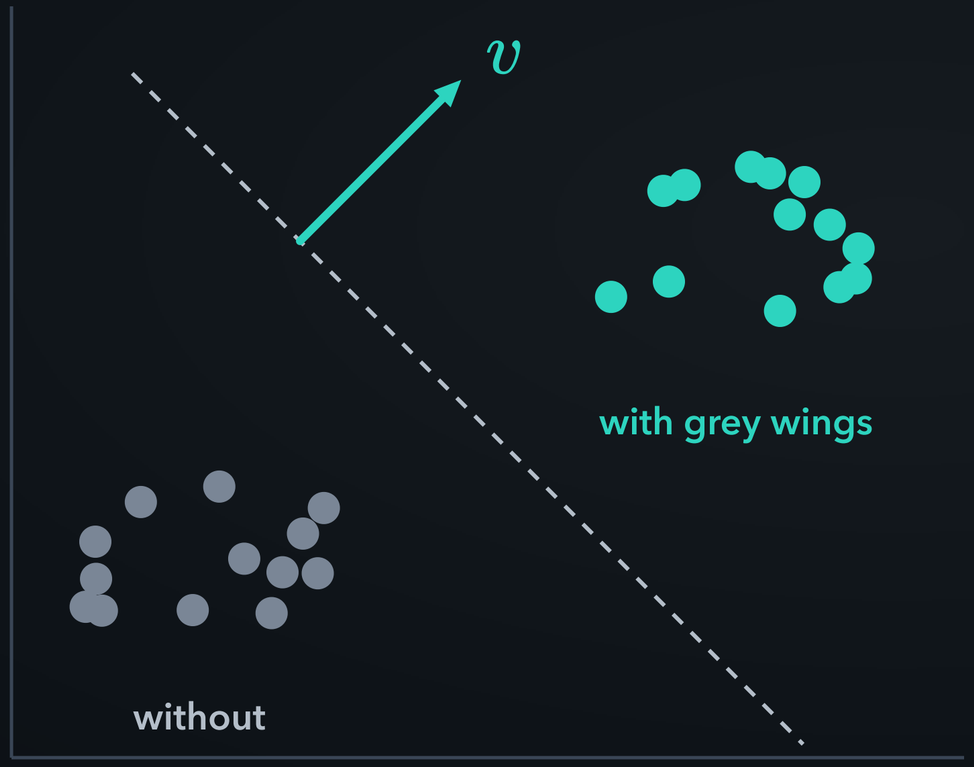
150 photos in each group

none is a **herring gull**

SCHEMATIC · drawings stand in for labelled bird photos (CUB, Part 2)

# Appendix: testing with concept activation vectors

## Step 2: the concept direction separates them inside the model




SCHEMATIC · 2 of 512 dimensions drawn

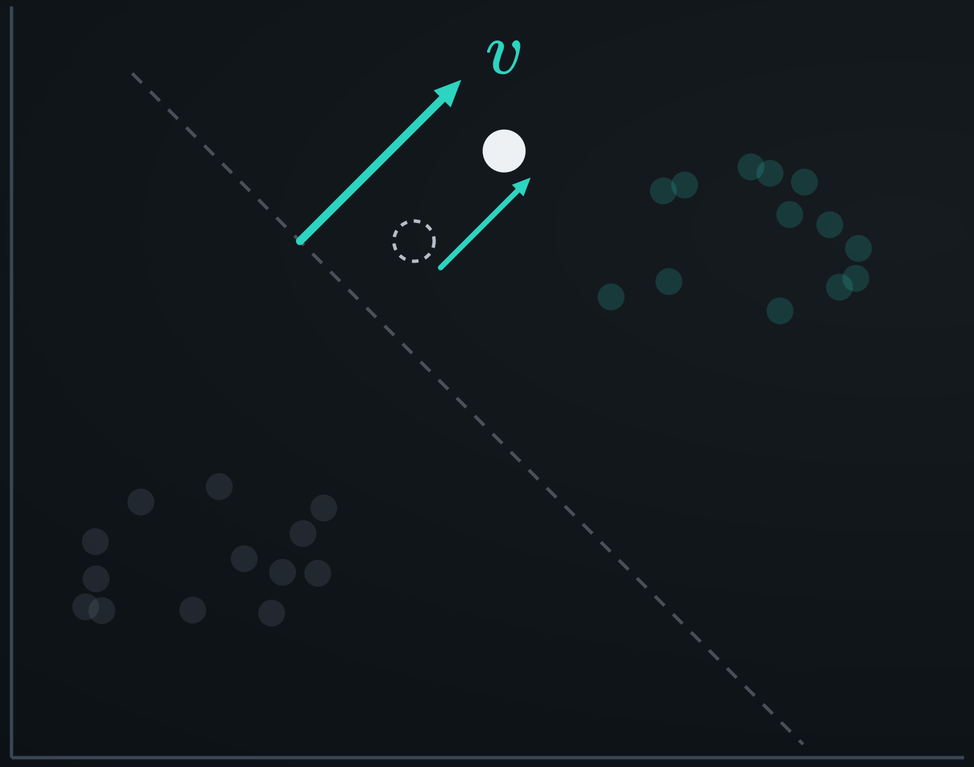
activations: the CLIP features

$$h(x) \in \mathbb{R}^{512}$$

$v$, the concept activation vector:
moving along it means more grey wing

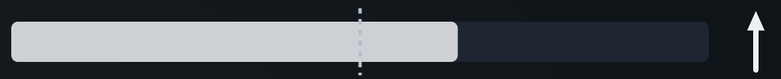
# Appendix: testing with concept activation vectors

## Step 3: nudge each herring gull photo along that direction




SCHEMATIC · one photo, 2 of 512 dimensions drawn

**herring gull** score



does the **score** go up?

$$\nabla_h \hat{y}_{\text{herring gull}} \cdot v > 0 \; ?$$

# Appendix: testing with concept activation vectors

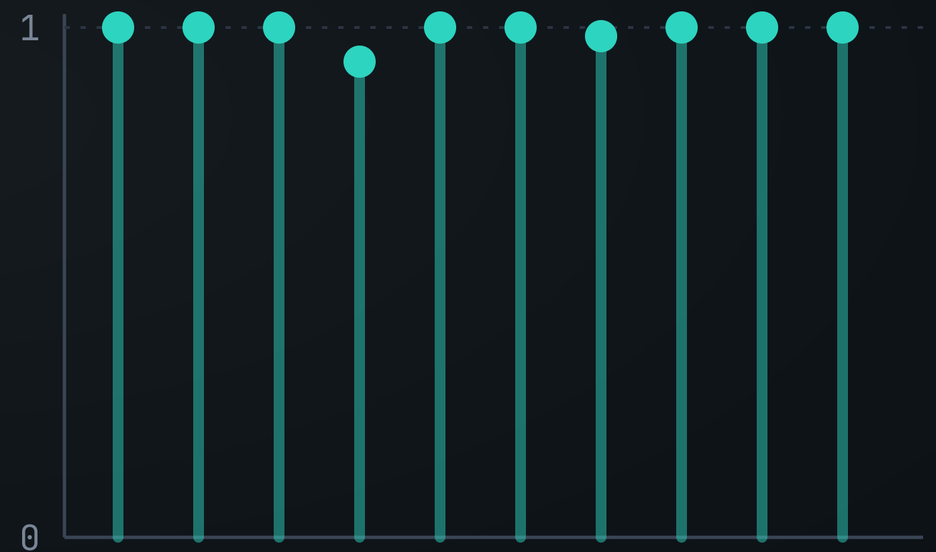
## The TCAV score: the share of photos whose score rises

the share of the **60 herring gull photos** whose **score** goes up

$$\mathrm{TCAV} = \frac{\#\{\, \nabla_h \hat{y} \cdot v > 0 \,\}}{60}$$

grey wings: between **0.93** and **1.00** in all 10 repeats

grey wings: one score per repeat

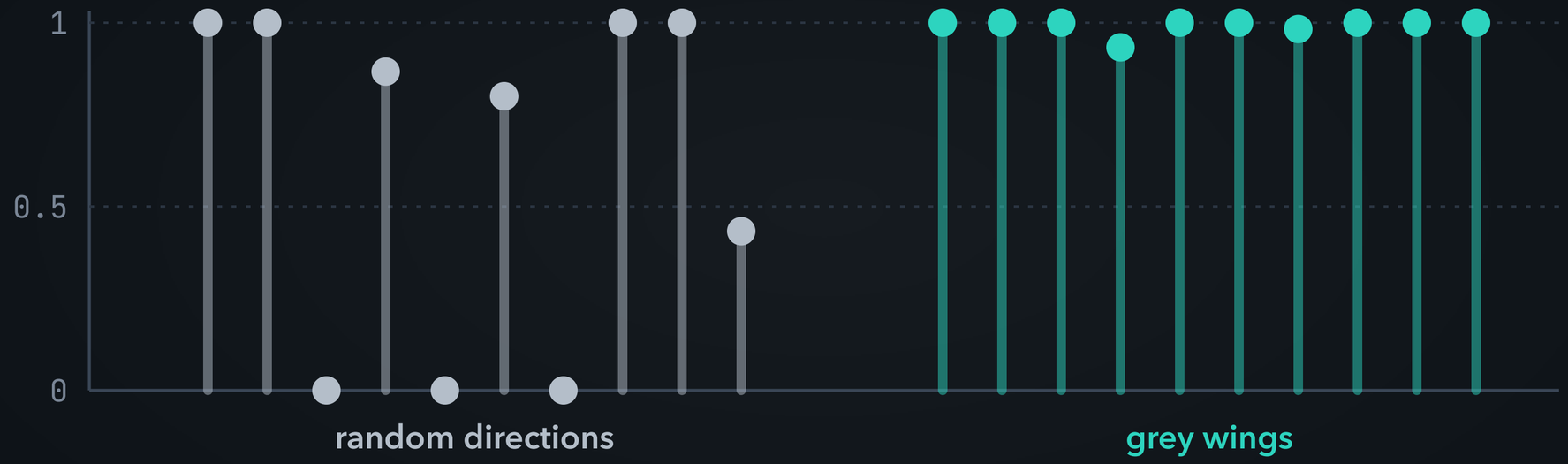



repeats 1 to 10, new example photos each time

COMPUTED · REAL MODEL · CLIP ViT-B/32 zero-shot herring gull score, CUB photos; Kim et al.'s method

# Appendix: testing with concept activation vectors

## Compare with random directions




a concept counts only if its score is steady and differs from random

it says what the model responds to, not how to fix it

COMPUTED · REAL MODEL · CLIP ViT-B/32 zero-shot herring gull score, CUB photos; Kim et al.'s method

APPENDIX · OPTIONAL FURTHER READING

# Grounding the concepts

Beyond the main lecture: do the concept names match the image?

# Grounding the concepts



## Why ground the concept targets in image regions?



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)




A CLIP score alone does not locate the concept in the image.

# Grounding the concepts

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

## Naming a concept is not the same as pointing at it

naming



`yellow bill`

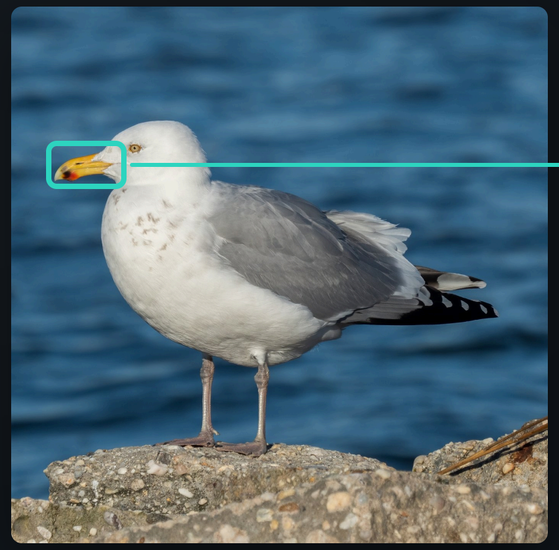
?

where?

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

a name, with no place in the image

pointing




✓ `yellow bill` box found: counts

✗ `loud laughing call` no box: not counted

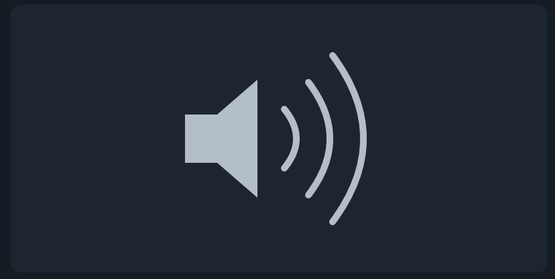
`ILLUSTRATIVE VALUES · box drawn for teaching`

# Grounding the concepts

$x$ — $g_\theta$ — $\hat{c}$ — $f_\phi$ — $\hat{y}$

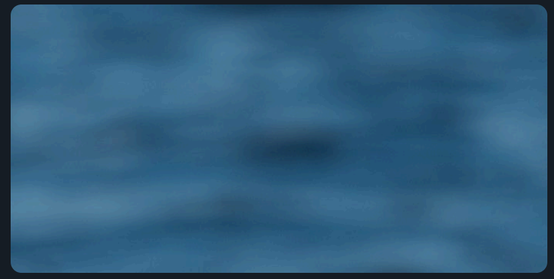
## Three ways a concept can fail

1 `loud laughing call`



not visual

2 `near the sea`



the setting, not the bird

3 `black head`



high score, yet absent

but this gull's head is white

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

`ILLUSTRATIVE VALUES · example concepts chosen for teaching`

# Grounding the concepts



## High class accuracy does not prove meaningful concepts

Control experiment: keep image knowledge, remove concept learning.




Never taught to detect yellow bill.

Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

With enough mixed features, the head can still **classify well**.

Class accuracy does not establish what each intermediate value means.

Random: how the weights are chosen. Fixed: whether training changes them.

REPORTED · Srivastava et al. 2024, Section 4 / Figure 3; bird pipeline schematic

# Grounding the concepts




## Random weights: draw the mixing coefficients by chance

Zoom in: the layer mixes features from the image encoder.

Draw the weights once by chance, without learning them from labels.

| image features | | weights | | contributions |
|---|---|---|---|---|
| 2 | × | 0.4 | = | 0.8 |
| 1 | × | -0.2 | = | -0.2 |

add the contributions

0.8 - 0.2 = 0.6

one mixed feature

Nothing taught this value to mean yellow bill.

ILLUSTRATIVE VALUES · chosen features and possible random weights; arithmetic computed

# Grounding the concepts




## Fixed means: training does not update these weights

During training of this baseline:

| image encoder | random mixing layer | class head |
|---|---|---|
| keep pretrained weights | keep the one draw | adjust using class labels |

**The same mixing weights can give different outputs:**

| photo | image features | same weights | mixed output |
|---|---|---|---|
| A | [2, 1] | [0.4, -0.2] | 0.6 |
| B | [1, 2] | [0.4, -0.2] | 0 |

**Check the concept meaning as well as the class accuracy.**

At test time, the trained class head stays fixed too.

ILLUSTRATIVE VALUES · same toy weights and two chosen feature vectors

# Grounding the concepts



## VLG-CBM asks a detector to find each concept

Vision-Language-Guided Concept Bottleneck Model (VLG-CBM)

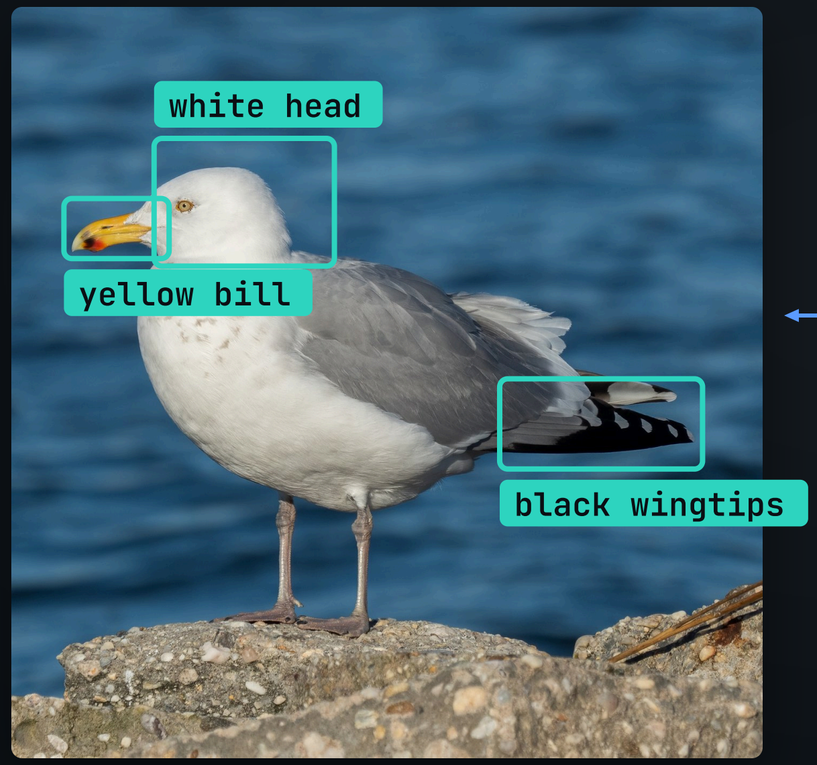



Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

concept names

each training image

open-vocabulary detector
one image at a time

a box where it is found

open-vocabulary: it searches for any phrase you give it, not only a fixed list of objects

ILLUSTRATIVE VALUES · boxes drawn for teaching, not a detector run

# Grounding the concepts



## A box means 1, no box means 0

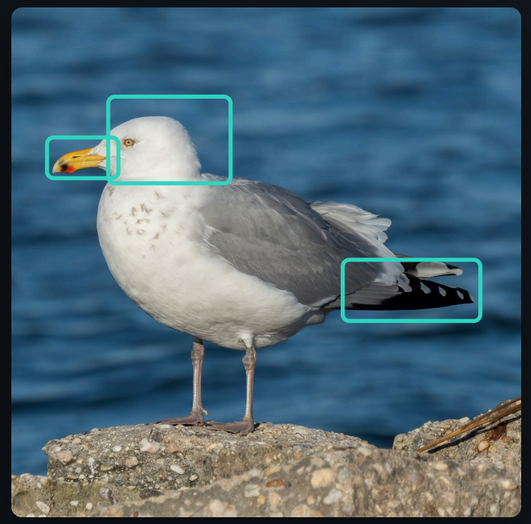


Photo: Rhododendrites, Wikimedia Commons, CC BY-SA 4.0 (cropped)

| CONCEPT | BOX? |
|---|---|
| yellow bill | ✓ |
| white head | ✓ |
| black wingtips | ✓ |
| loud laughing call | none |

grounded labels

$$\mathbf{c}^{\text{det}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

concept labels from the detector

ILLUSTRATIVE VALUES · boxes drawn for teaching

# Grounding the concepts




## The grounded labels train the concept layer with a loss

concept layer: backbone features → one score per concept

concept loss — sum over concepts — binary cross-entropy — detector label — predicted concept

$$\mathcal{L}_C = \sum_j \mathrm{BCE}\left( c_j^{\mathrm{det}}, \ \hat{c}_j \right)$$

as in the original CBM, but no person draws the labels

# Grounding the concepts




## One term of the loss, worked out

$$\mathrm{BCE}(c, \hat{c}) = -\log \hat{c} \ \text{ if } c = 1, \qquad -\log(1 - \hat{c}) \ \text{ if } c = 0$$

| CONCEPT | BOX? | LABEL | PREDICTED | TERM |
|---|---|---|---|---|
| black wingtips | ✓ | 1 | 0.8 | $-\log 0.8 \approx 0.22$ |
| loud laughing call | none | 0 | 0.2 | $-\log(1 - 0.2) \approx 0.22$ |

a prediction close to its label costs little

ILLUSTRATIVE VALUES · predictions made up for teaching

# Grounding the concepts




## Count the non-zero weights in each class's row

the head: one weight per concept

| one row per class | grey back | black wingtips | yellow bill | red bill spot | dark back | pink legs | non-zero |
|---|---|---|---|---|---|---|---|
| Herring Gull | 1.2 | 0.8 | 0 | 0.9 | 0 | 0 | 3 |
| California Gull | 0.7 | 0 | 1.1 | 0 | 0 | 0 | 2 |
| Western Gull | 0 | 0.5 | 0 | 0.4 | 1.3 | 0.6 | 4 |

number of effective concepts (NEC): the average count

$$\text{NEC} = \frac{3 + 2 + 4}{3} = 3$$

ILLUSTRATIVE VALUES · weights drawn for teaching

# Grounding the concepts




## A head that uses every concept explains nothing

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| random | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| **Herring Gull** | 0.3 | -0.1 | 0.2 | 0.4 | -0.2 | 0.1 | 0.3 | -0.3 | 0.2 | 0.1 | -0.1 | 0.4 | 0.2 | -0.2 | 0.3 | 0.1 | -0.4 | 0.2 | 0.1 | -0.1 |

20 of 20 weights non-zero

Even random, untrained concepts, given enough of them, can approximate any linear classifier.

**Accurate, and still nothing to read.**

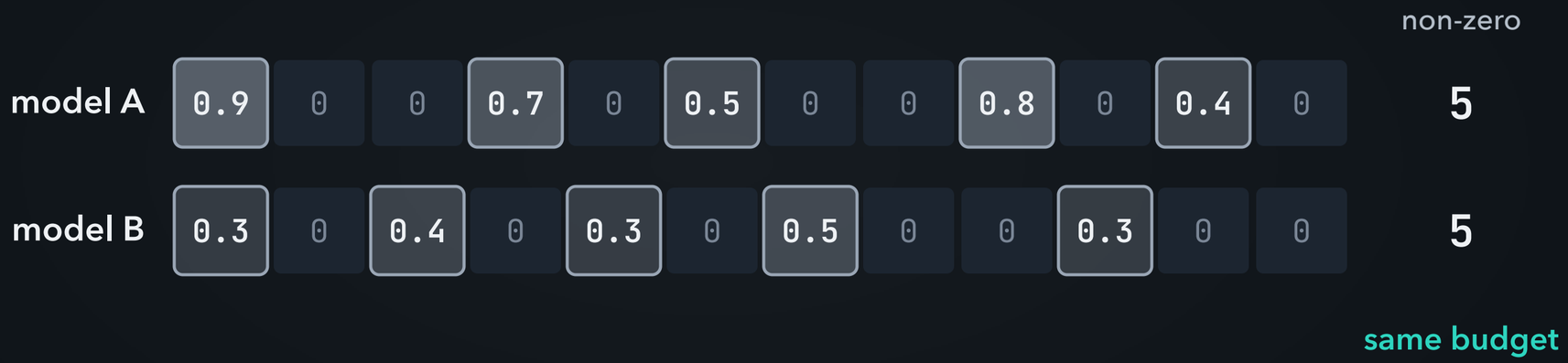
REPORTED · Theorem 4.1, Srivastava et al. 2024

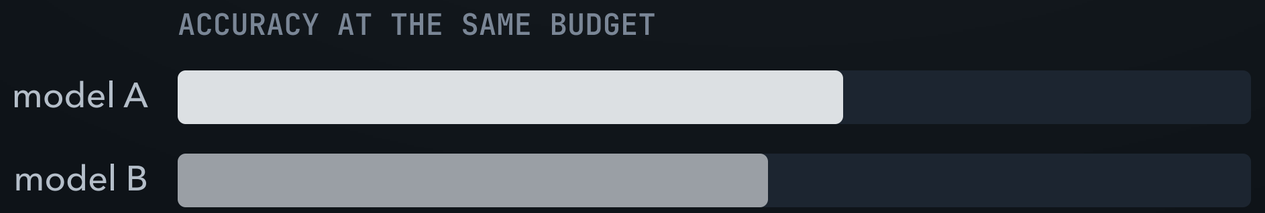
ILLUSTRATIVE VALUES · weights drawn for teaching

# Grounding the concepts




## A fair comparison: the same number of concepts per class




the gap now says something about the concepts

ILLUSTRATIVE VALUES · weights and bars drawn for teaching

# Grounding the concepts



## The same count in symbols

average over the **K classes**

non-zero weights in row k

$$\mathrm{NEC} = \frac{1}{K} \sum_{k=1}^{K} \|W_k\|_0$$

Herring Gull




$$\|W_{\text{Herring}}\|_0 = 3$$

$\|\cdot\|_0$ counts the entries that are not zero

ILLUSTRATIVE VALUES · weights drawn for teaching (the row from the count slide)

# Grounding the concepts



## Five concepts per class can actually be read

$$\mathrm{NEC} = 5$$




each class: a handful of named concepts

ILLUSTRATIVE VALUES · weights drawn for teaching

# Grounding the concepts




## From direct CLIP scores to grounded concept learning

| | direct CLIP scoring | | VLG-CBM |
|---|---|---|---|
| where concept values come from | CLIP similarities | → | **layer trained on boxes** |
| what trains the class head | **class labels** | → | **class labels** |

# Grounding the concepts




## VLG-CBM in symbols

**VLG-CBM (Srivastava et al., NeurIPS 2024)** x → ĉ → ŷ

| | |
|---|---|
| concept predictor | $\hat{c} = W_c\, e(x),$ trained on detector labels $c^{\text{det}}$ |
| what reaches the head | concept scores $\hat{c}_1, \ldots, \hat{c}_k$ |
| task head | $\hat{y} = f_\phi(\hat{c}) = W\hat{c} + b,$ sparse, NEC fixed |
| what is trained | $W_c,\ f_\phi$ |
| data needed | $\{(x, y)\}$ + an open-vocabulary detector |

# Concept Bottleneck Models

Predicting through interpretable concepts.

Narration: synthetic voice (Gemini TTS). Photographs: Rhododendrites, Yathin S Krishnappa, Gzen92, Dietmar Rabich, Hans Hillewaert, Stillwaterising, and the authors of the 36 Part 10 photographs, via Wikimedia Commons (CC BY-SA, CC BY, CC0, public domain); knee X-ray from Koh et al. 2020 (CC BY 4.0). Satellite imagery: Sentinel-2, SentinelKilnDB. Data: CUB-200-2011. Models: CLIP ViT-B/32, ResNet-50; concept lists: Gemini.
Sources: Koh et al. 2020; Wah et al. 2011; Simonyan et al. 2014; Kim et al. 2018; Adebayo et al. 2018; Zech et al. 2018; Rudin 2019; Chen et al. 2019; Zarlenga et al. 2022; Yuksekgonul et al. 2023; Menon and Vondrick 2023; Srivastava et al. 2024; Mahinpei et al. 2021; Havasi et al. 2022; Knab et al. 2026.