Lecture 20: Interpretability

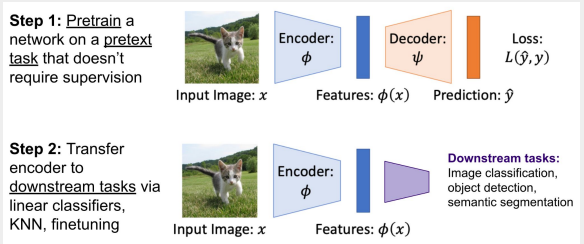
Administrative

- A5 due June 5. Required for Grad version.
- Project report due June 8
- Project poster: June 8, 10:30am – 12:20pm at Allen Atrium

What have we learned so far?

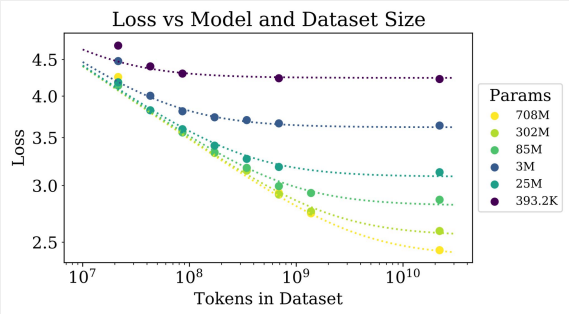
Self-Supervised Learning

Pretrain and then Transfer



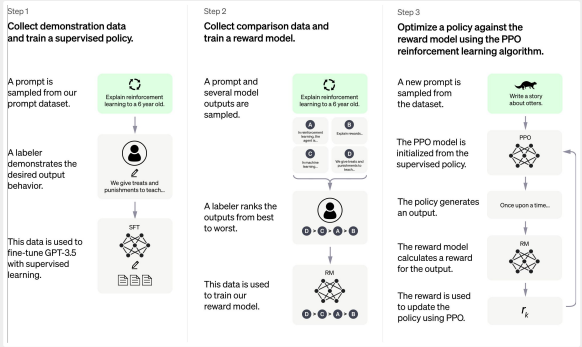
Scaling data and parameters

Bigger models learn more from more data

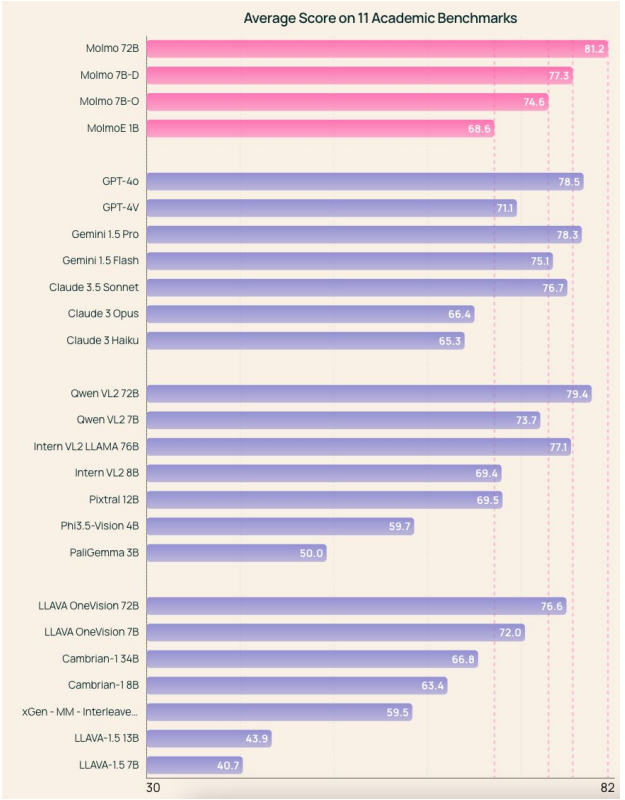


RL from Human Feedback

(+ instruction tuning) GPT → ChatGPT



Foundation Models today can do a lot!



But they exhibit surprising failure cases!

Where is the mug? Answer with the option alone. (A) On the table (B) Under the table (C) Right of the table (D) Left of the table

C) Right of the table

Is the mug on the table?

Yes, the mug is sitting on the table.

Is the mug under the table?

No, the mug is not under the table. It is sitting on top of the table.

the grass is eating the horse	81%
the horse is eating the grass	78%

How can we trust them to diagnose diseases and drive cars?

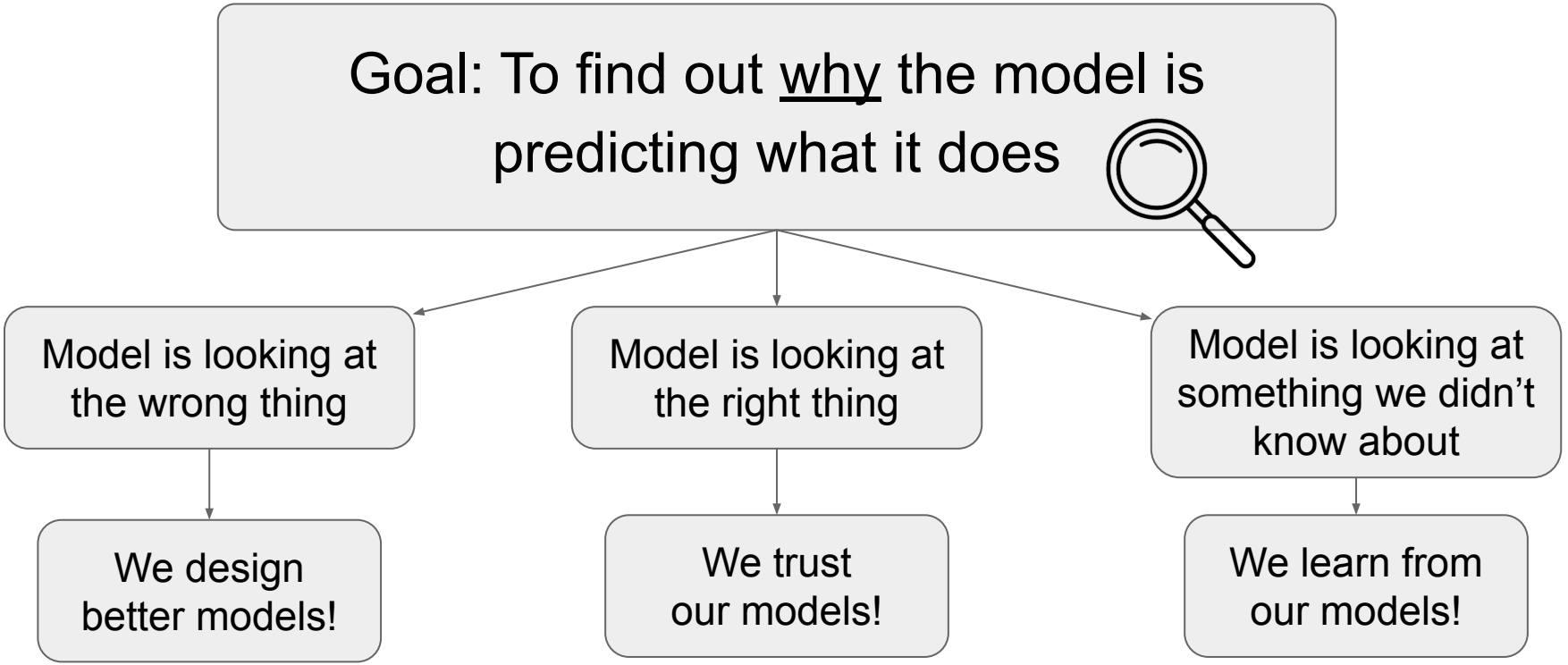
A cost of scale

We've been scaling our models and data up and up, and they've been doing great on benchmarks

But in doing so, have we lost the capability to look into the models to understand why they predict what they do?

Not yet!

Interpretability



Selvaraju et al., 2019. "Grad-CAM"

Interpretability Methods for Vision

Traditional Methods

Pixel-level interpretability
Feature-level interpretability

Modern Methods

Attention as explanation
Probing
Concepts and Counterfactuals
Mechanistic interpretability

By the end of today, you should have:

- A working knowledge of various interpretability methods
- The spirit of skepticism and critical thinking!

Interpretability Methods for Vision

Traditional Methods

Pixel-level interpretability
Feature-level interpretability

Modern Methods

Attention as explanation
Probing
Concepts and Counterfactuals
Mechanistic interpretability

Pixel-Level Interpretability: Saliency

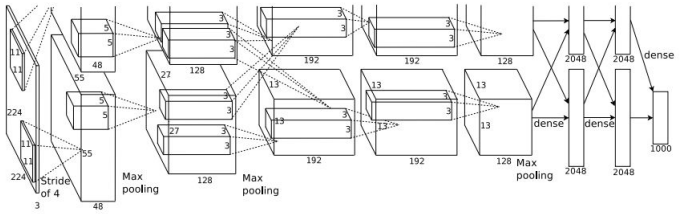
- "Saliency" → importance

- Our usual question: which ***weights*** contributed most to the loss? $\longrightarrow \frac{\partial L}{\partial \theta}$

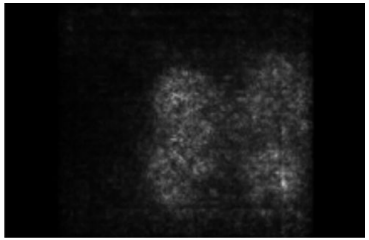
- Now, we ask: which ***pixels*** contributed most to the class score? $\longrightarrow \frac{\partial S_c}{\partial x}$

Which pixels matter: Saliency via Backprop

Forward pass: Compute probabilities



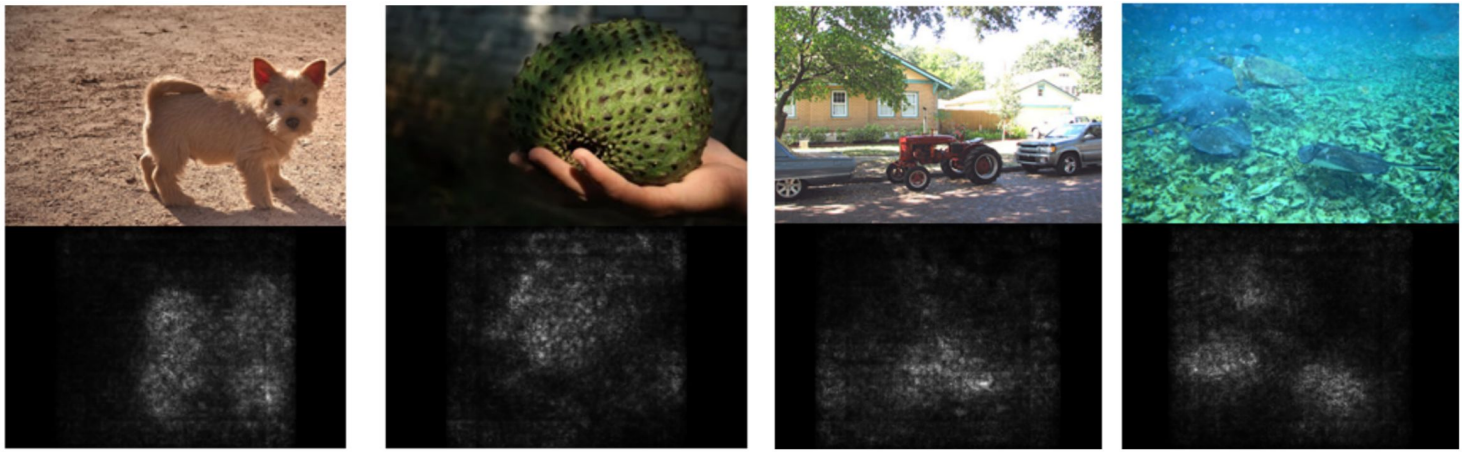
Dog



Compute gradient of (unnormalized) class score with respect to image pixels, take absolute value and max over RGB channels

Simonyan, Vedaldi, and Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR Workshop 2014.
Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

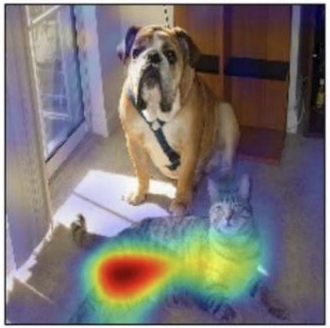
Saliency Maps



Simonyan, Vedaldi, and Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR Workshop 2014.

Grad-CAM

- Issue with naive saliency: high-resolution but often noisy and sensitive to small perturbations
- Grad-CAM: Gradient-weighted Class Activation Mapping
  - Highlights high-level spatial regions of the image that most influenced the prediction
  - Uses gradients of a target class score to weight feature maps of a CNN layer
  - Works for all layers, all architectures, even different tasks:



(c) Grad-CAM 'Cat'

(i) Grad-CAM 'Dog'

A man is sitting at a table with a pizza

Selvaraju et al., 2019. "Grad-CAM"

Saliency as a diagnostic tool

Waterbirds dataset

Common training examples

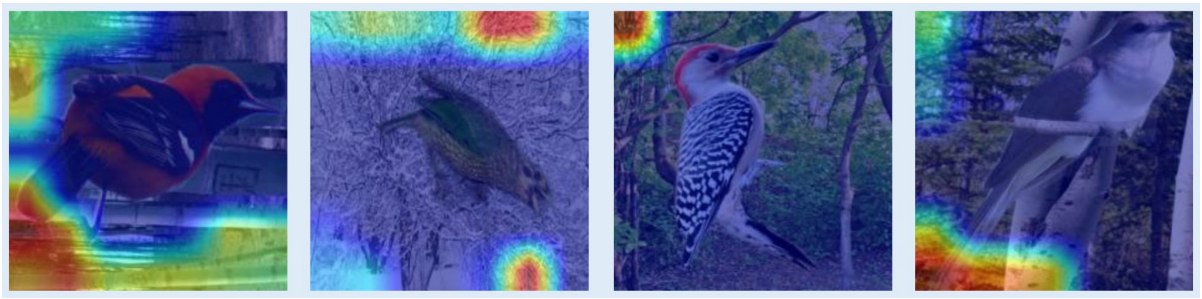
y: waterbird
a: water
background

y: landbird
a: land
background

Test examples

y: waterbird
a: land
background

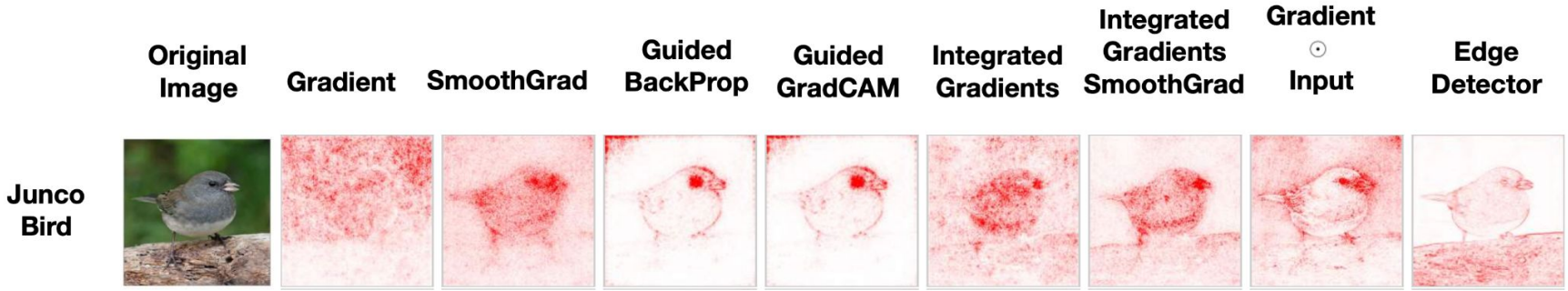
Grad-CAM shows that the most salient pixels are the background:



Sagawa et al., 2020: Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization
Wu et al., 2023: Discover and Cure: Concept-aware Mitigation of Spurious Correlation

Saliency as a diagnostic tool: The catch

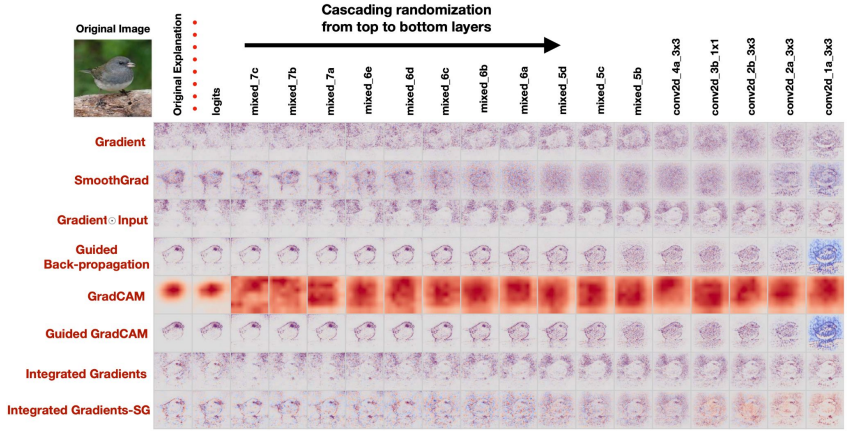
Adebayo et al., 2020: Sanity Checks for Saliency Maps



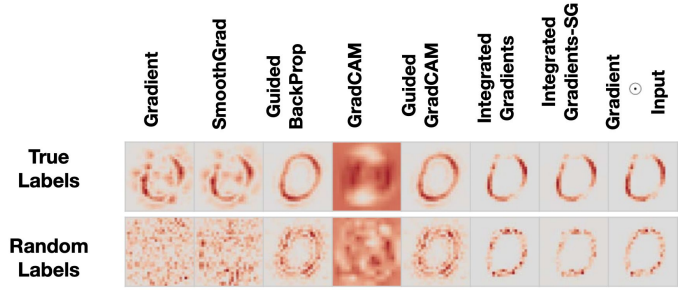
Proposal: Let's randomize the model weights or training data—if the saliency map is really capturing what the model has learned about the data, the maps should look very different.

Saliency as a diagnostic tool: The catch

For many saliency methods, the saliency maps look the same even when the model weights are randomized, or the training data is randomized!



Random weights



Random training data

This means that many saliency methods were not actually reflecting what the trained model learned from the training data!!

Adebayo et al., 2020: Sanity Checks for Saliency Maps

Plausibility vs Faithfulness

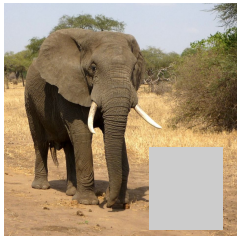
Plausibility: The explanation makes sense

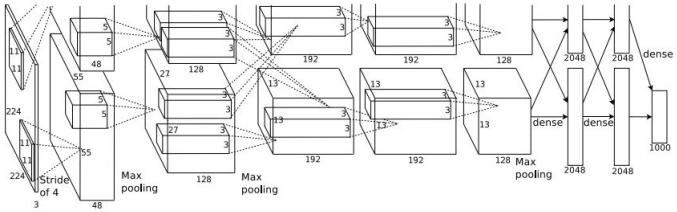
Faithfulness: The explanation is actually what the model is doing

The field of interpretability moved towards methods that show causality

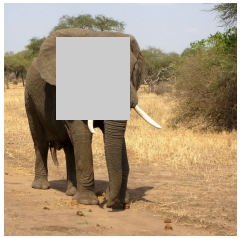
Saliency via Occlusion

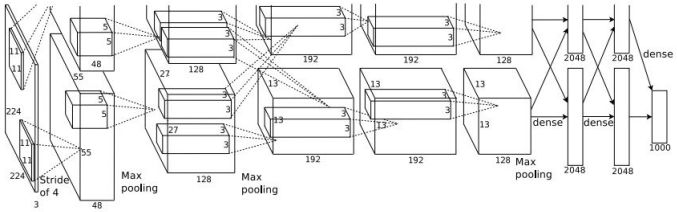
Mask part of the image before feeding to CNN,
check how much predicted probabilities change





P(elephant) = 0.95



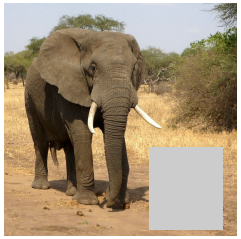


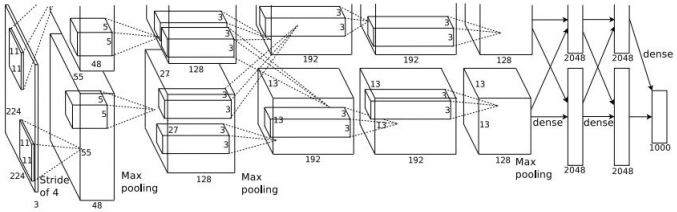
P(elephant) = 0.75

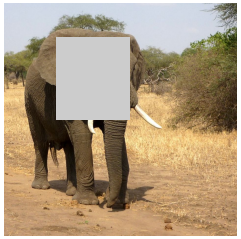
Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014

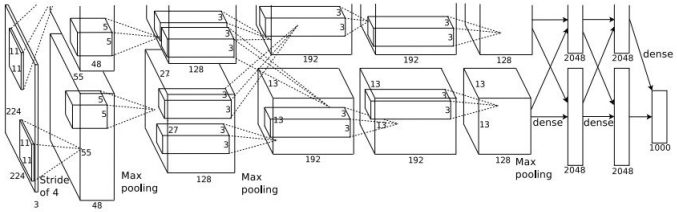
Saliency via Occlusion

Mask part of the image before feeding to CNN, check how much predicted probabilities change

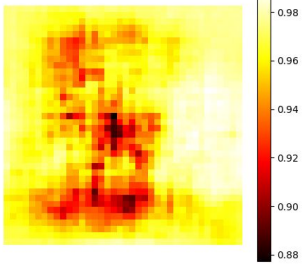




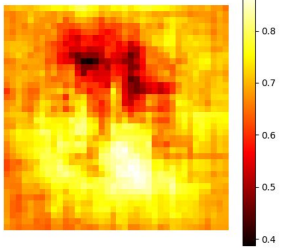




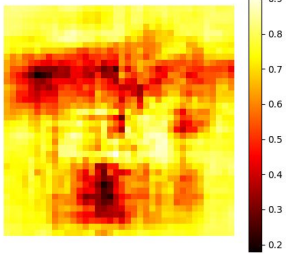
schooner



African elephant, Loxodonta africana



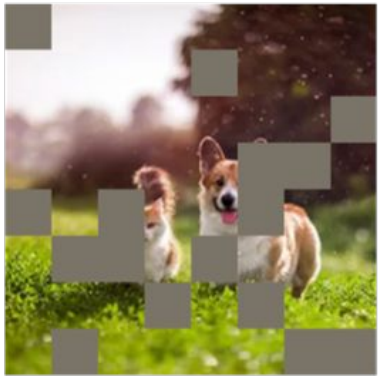
go-kart



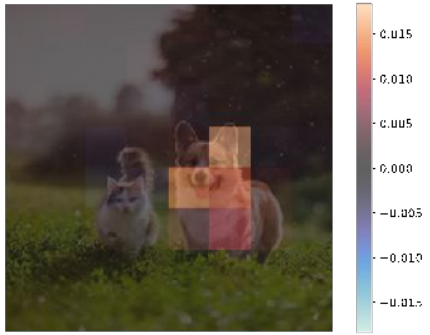
Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014

Saliency via Occlusion: Shapley Values

P(corgi) = 0.99



P(corgi) = 0.8



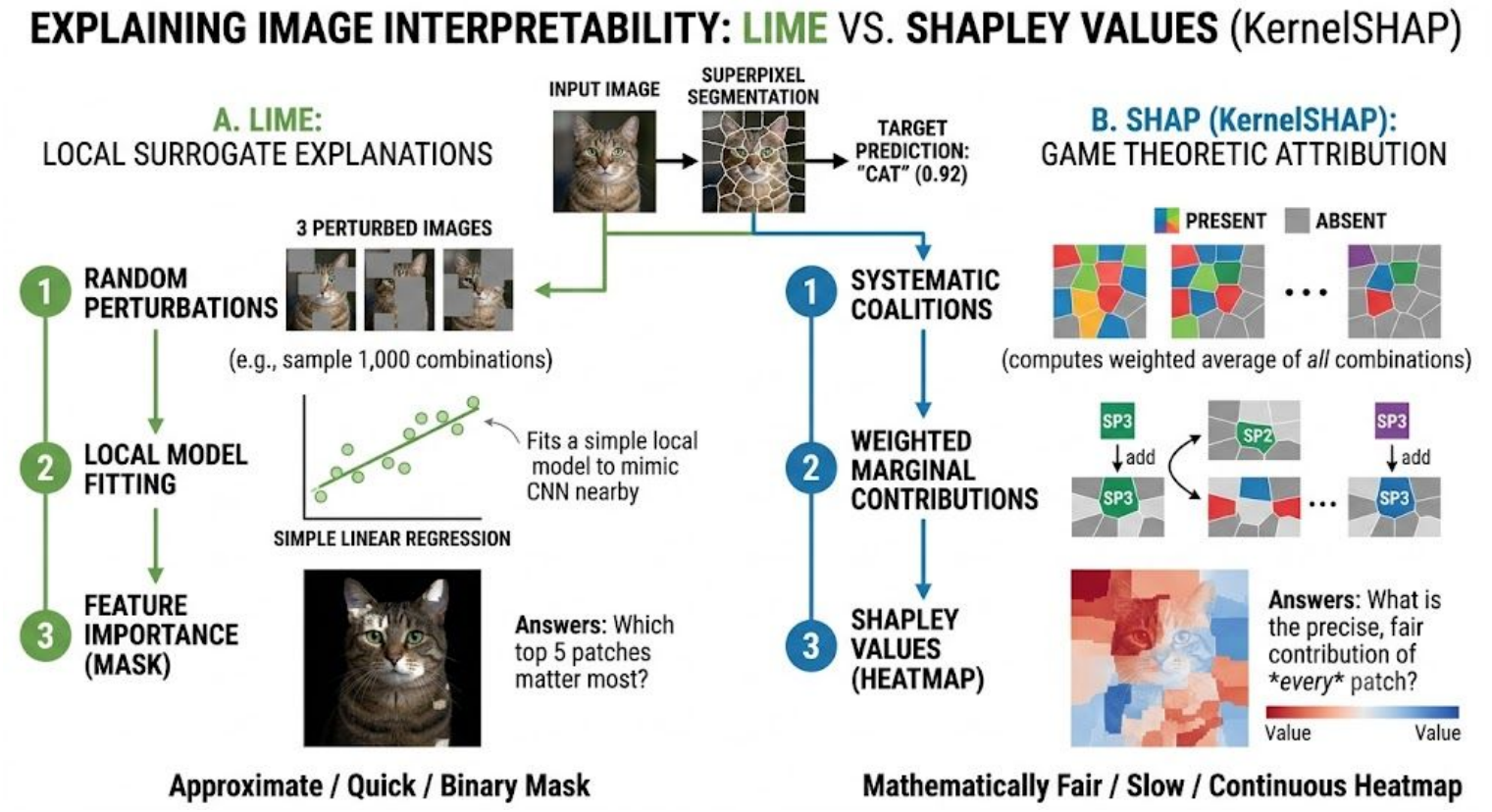
Credit: Ian Covert; Lundberg & Lee 2017

Saliency via Occlusion: LIME

Rather than considering all combinations, samples a random set of combinations and fits a linear classifier.

(a) Husky classified as wolf (b) Explanation

Problem with Shapley and LIME: These occlusions (grey/black/blurred patches) make the resulting images OOD, which may impact model behavior on them

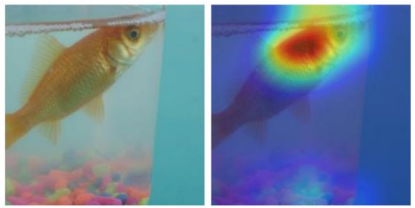


Citation/Warning: Generated by Gemini

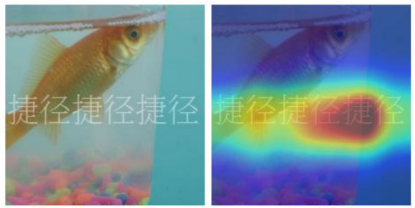
Saliency + Causality

Grad-CAM suggests that ResNet trained on ImageNet1K was using watermarks to identify the “carton” class

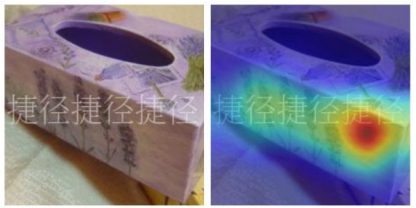
ImageNet-W: Add watermarks to other images and see whether the model’s prediction changes



Prediction: goldfish



w/ Watermark: carton



w/ Watermark: pencil sharpener → carton

“counterfactual”

Li et al., 2023: A Whac-A-Mole Dilemma : Shortcuts Come in Multiples Where Mitigating One Amplifies Others

Interpretability Methods for Vision

Traditional Methods

Pixel-level interpretability
Feature-level interpretability

Modern Methods

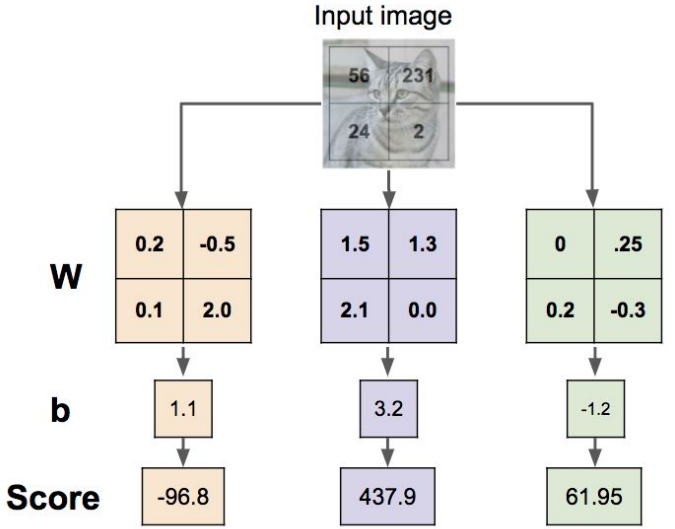
Attention as explanation
Probing
Concepts and Counterfactuals
Mechanistic interpretability

Feature-level Interpretability

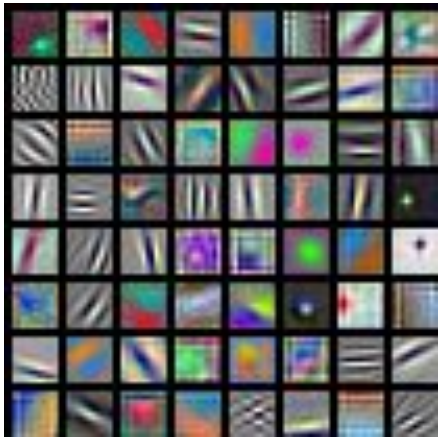
Pixel-level: What pixels contributed to the prediction for a specific input image?

Feature-level: What features has the model learned during training?

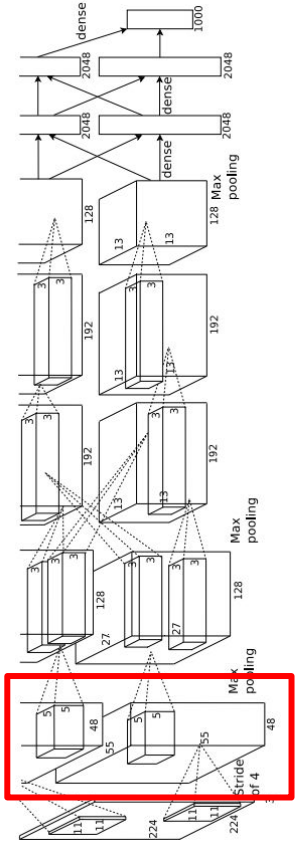
Interpreting a Linear Classifier



CNN First Layer: Visualize Filters

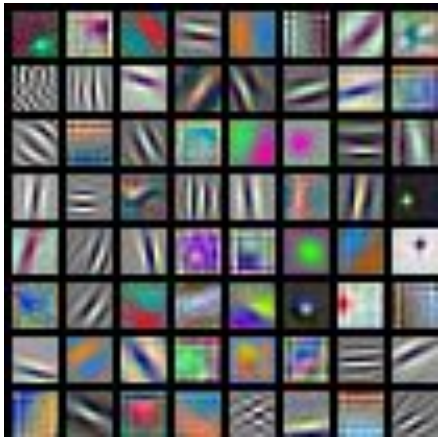


AlexNet:
64 x 3 x 11 x 11



Krizhevsky, “One weird trick for parallelizing convolutional neural networks”, arXiv 2014
He et al, “Deep Residual Learning for Image Recognition”, CVPR 2016
Huang et al, “Densely Connected Convolutional Networks”, CVPR 2017

CNN First Layer: Visualize Filters

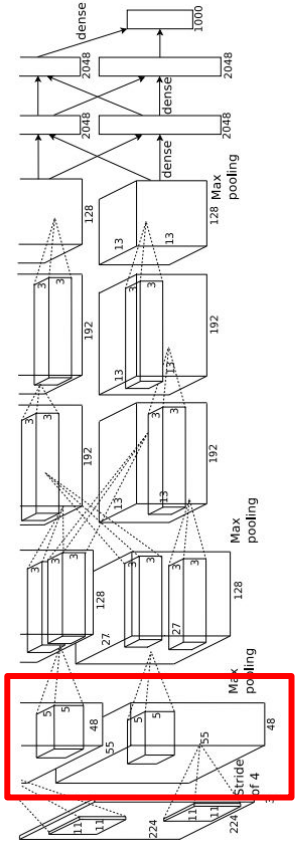


AlexNet:
64 x 3 x 11 x 11

ResNet-18:
64 x 3 x 7 x 7

ResNet-101:
64 x 3 x 7 x 7

DenseNet-121:
64 x 3 x 7 x 7



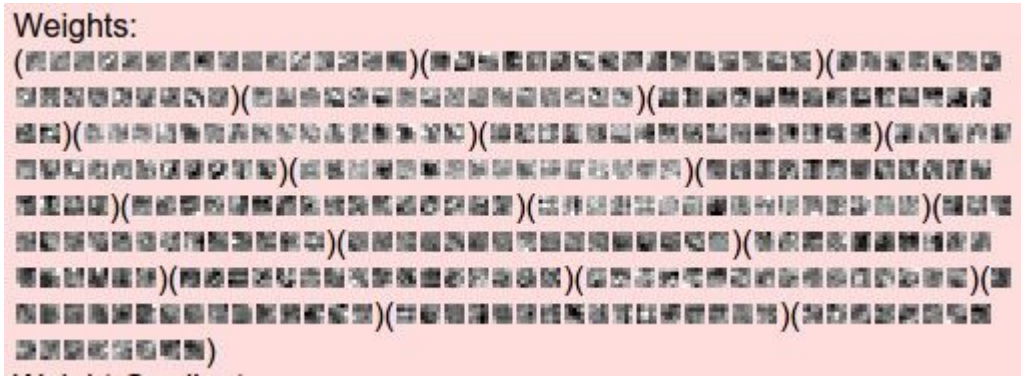
Krizhevsky, “One weird trick for parallelizing convolutional neural networks”, arXiv 2014
He et al, “Deep Residual Learning for Image Recognition”, CVPR 2016
Huang et al, “Densely Connected Convolutional Networks”, CVPR 2017

Visualize the filters/kernels (raw weights)

We can visualize filters at higher layers, but not that interesting

layer 1 weights

16 x 3 x 7 x 7



layer 2 weights

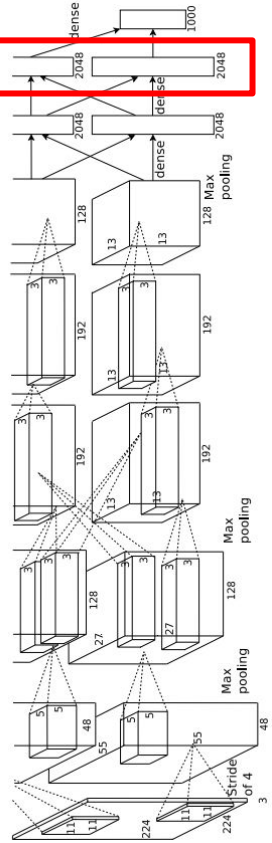
20 x 16 x 7 x 7

layer 3 weights

20 x 20 x 7 x 7

Last Layer

FC7 layer

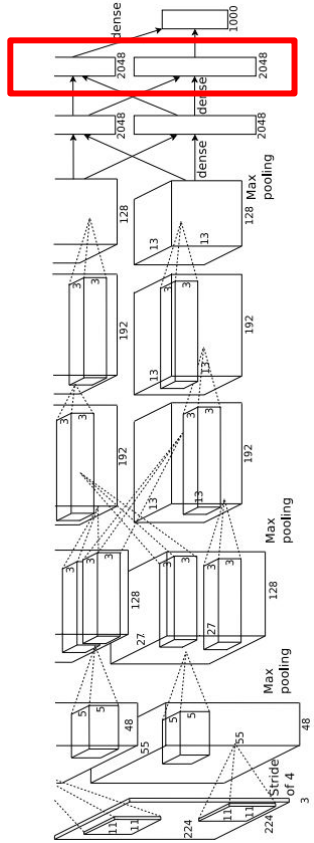


4096-dimensional feature vector for an image (layer immediately before the classifier)

Run the network on many images, collect the feature vectors

Last Layer: Nearest Neighbors

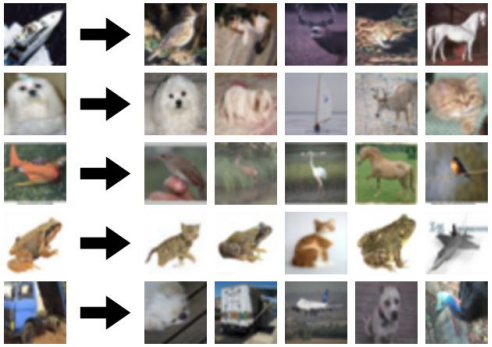
4096-dim vector



Test image

L2 Nearest neighbors in <u>feature</u> space

Recall: Nearest neighbors in <u>pixel</u> space



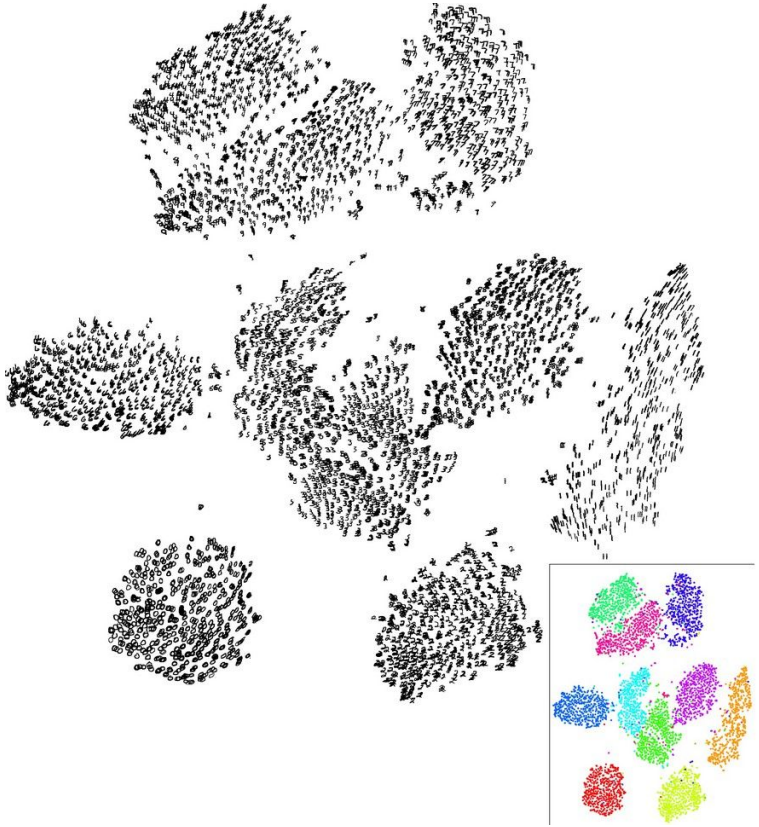
Krizhevsky et al, “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS 2012.
Figures reproduced with permission.

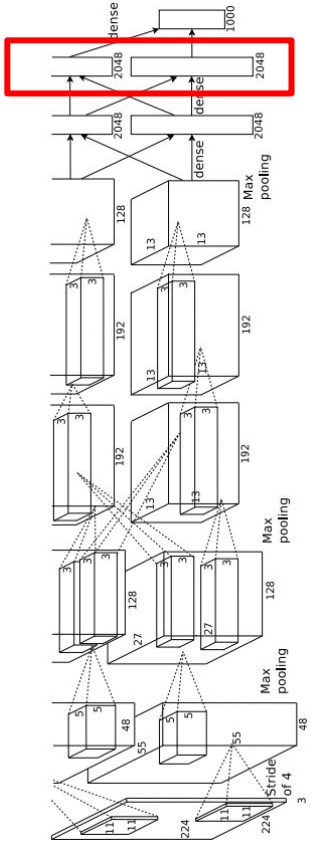
Last Layer: Dimensionality Reduction

Visualize the “space” of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions

Simple algorithm: Principal Component Analysis (PCA)

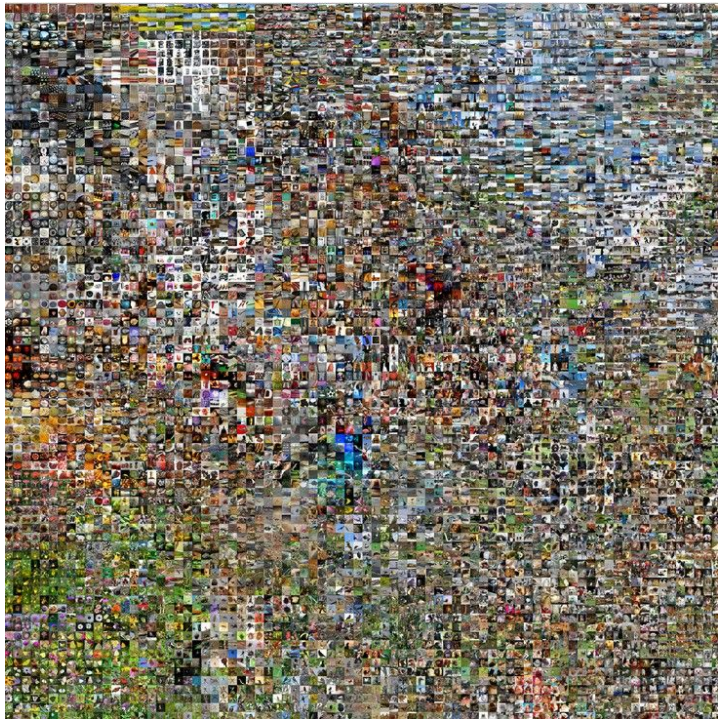
More complex: **t-SNE**

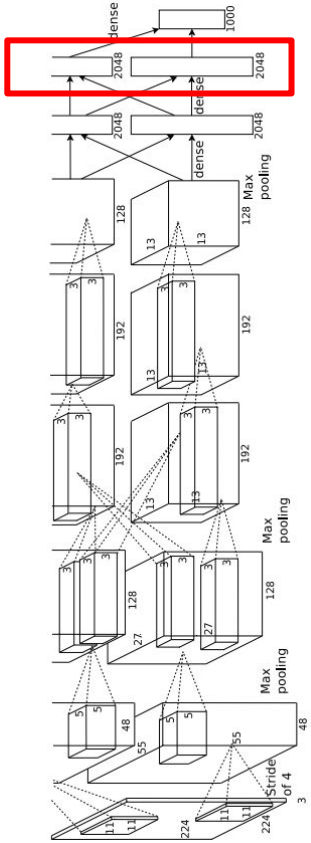




Van der Maaten and Hinton, “Visualizing Data using t-SNE”, JMLR 2008

Last Layer: Dimensionality Reduction





Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008
Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.
Figure reproduced with permission.

See high-resolution versions at
http://cs.stanford.edu/people/karpathy/cnnembed/

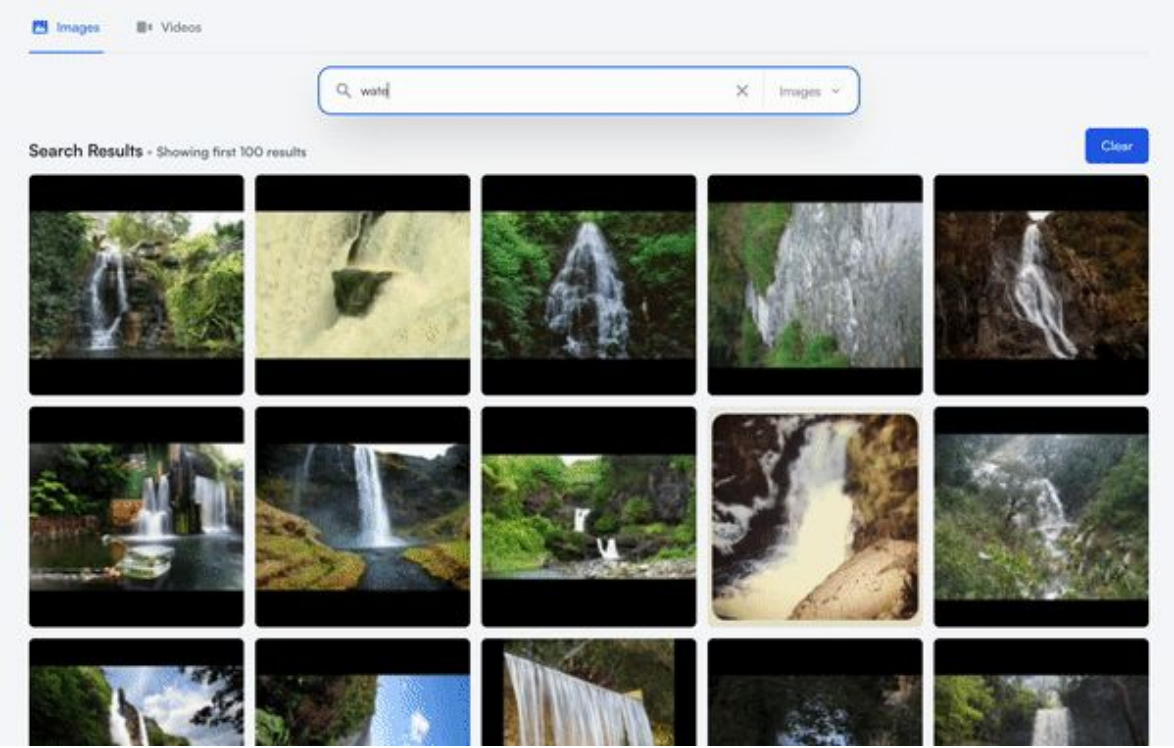
A shared representation space for images and text

“A cute fluffy cat”

“My favorite dog is a golden retriever”

This enables multimodal tasks like text-to-image retrieval, and more…

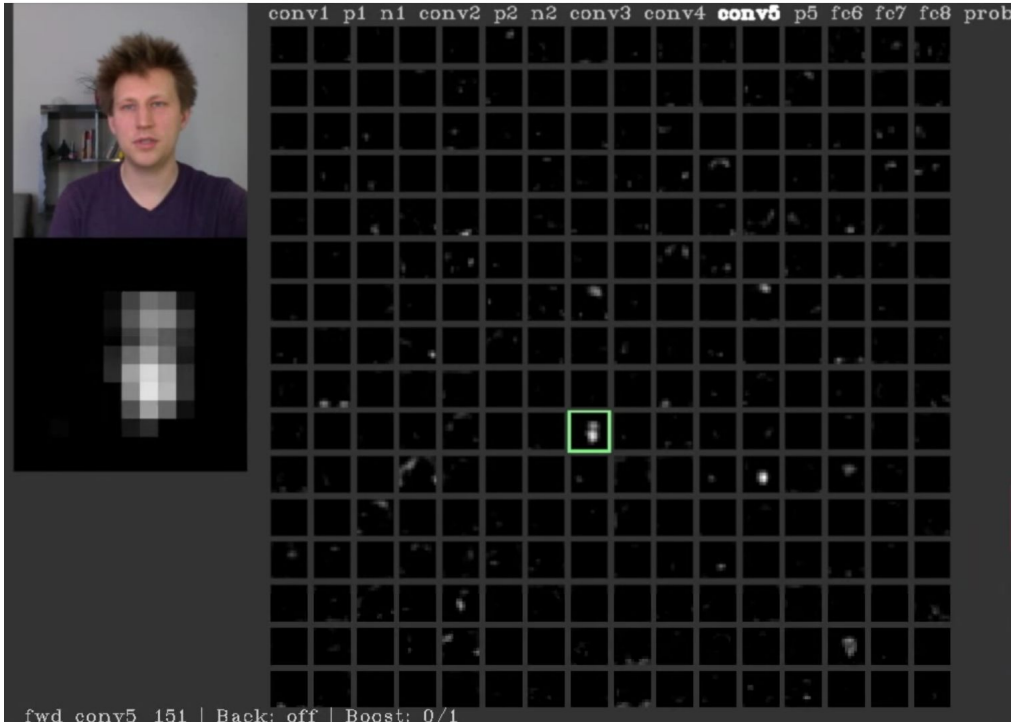
Last Layer: Modern Day Search



coactive.ai

Visualizing Activations

conv5 feature map is 128x13x13; visualize as 128 13x13 grayscale images



Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.
Figure copyright Jason Yosinski, 2014. Reproduced with permission.

https://www.youtube.com/watch?v=AgkfIQ4IGaM

Activation Maximization

1. Pick a neuron
2. Perform gradient *ascent* on the input to find/generate the image that maximizes its activation.

→ Gives us insight into what specific neurons have learned.

goose

Flamingo

Simonyan, Vedaldi, and Zisserman, 2014: “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”

Interpretability Methods for Vision

Traditional Methods

Pixel-level interpretability
Feature-level interpretability

Modern Methods

Attention as explanation
Probing
Concepts and Counterfactuals
Mechanistic interpretability

Attention as explanation

- We discussed saliency as an explanation in CNNs
- What would be a parallel in today's models, especially those that use Transformers?

  Attention!
- Idea: If the model is attending to the right parts of the input, it must be learning the right thing… right?

Xu et al., 2016: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Attention is not Explanation

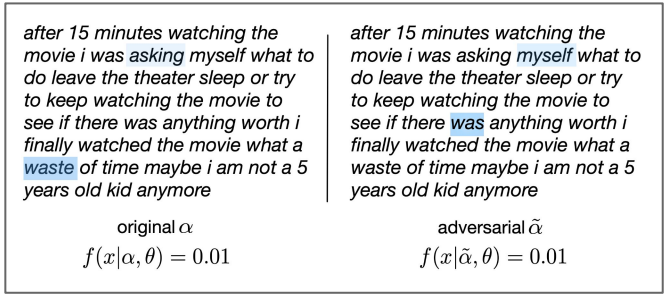
Jain and Wallace, 2019

They state two main criteria for an explanation to be "faithful":

(i) Attention weights should correlate with feature importance measures: e.g., leave-one-out (LOO) methods;

(ii) Alternative/counterfactual attention weight configurations ought to yield corresponding changes in prediction.

Their experiments on NLP tasks showed that neither was true, and further, that there are several cases where randomly permuting attention weights induced only minimal changes in the output.



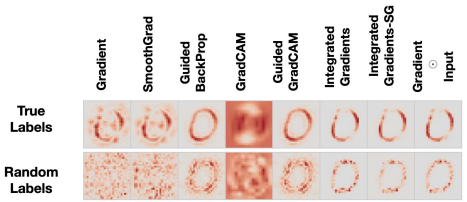
Attention is not Explanation — Jain and Wallace, 2019

Attention is not not Explanation — Wiegreffe and Pinter, 2019

Is Attention Interpretable? — Serrano and Smith, 2019

Consensus: Attention is plausible, but is not necessarily faithful.

Just like pixel-level salience!



Takeaway: Be critical!

Adebayo et al., 2020: Sanity Checks for Saliency Maps

Interpretability Methods for Vision

Traditional Methods

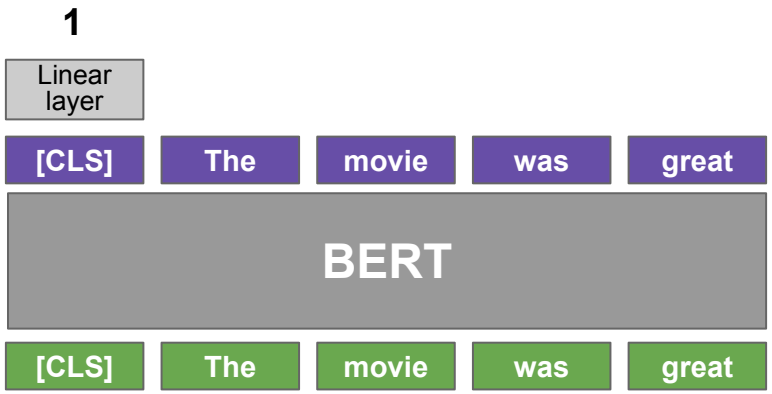
Pixel-level interpretability
Feature-level interpretability

Modern Methods

Attention as explanation
Probing
Concepts and Counterfactuals
Mechanistic interpretability

Probing

- I have a model that I've trained to do some task. How do we find out what the learned representations from that model encode?
- Say I have a model that is trained to predict the sentiment of a movie review.



What you're really doing here is finding out that the learned representations of BERT encode sentiment — unsurprising, because you've finetuned the whole of BERT (and the linear layer) to do that task.

But now we want to find out what BERT learns that we didn't explicitly train it to.

Probing

Say we want to know whether the representations of BERT encode part-of-speech, without being explicitly trained to do so.

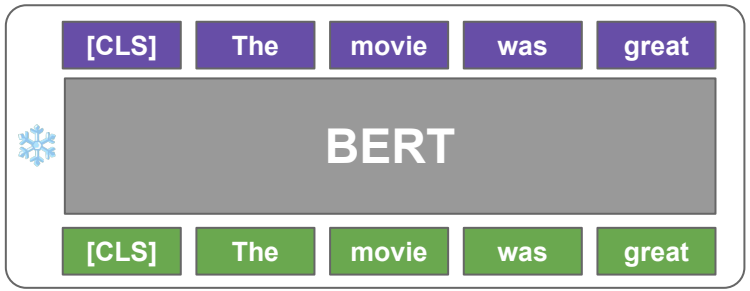
Step 1: Freeze BERT, and get its representation of 1000 different words

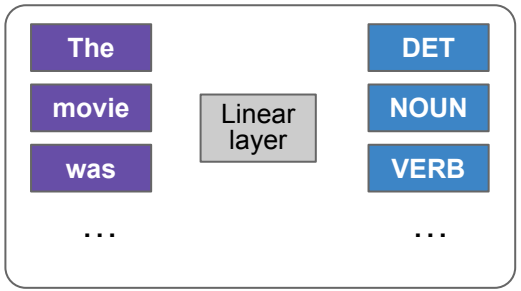
Step 2: Label them with part-of-speech (POS)

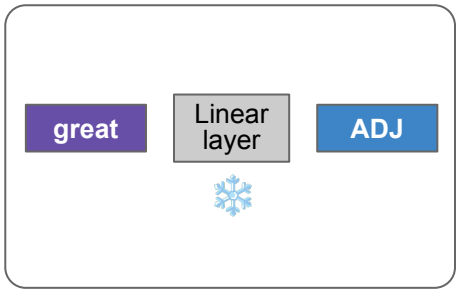
Step 3: Train a simple classifier on 90% of the (representation, POS) pairs

Step 4: Evaluate the simple classifier on the 10% held-out pairs.

If it does well, then the representations encode part-of-speech*.



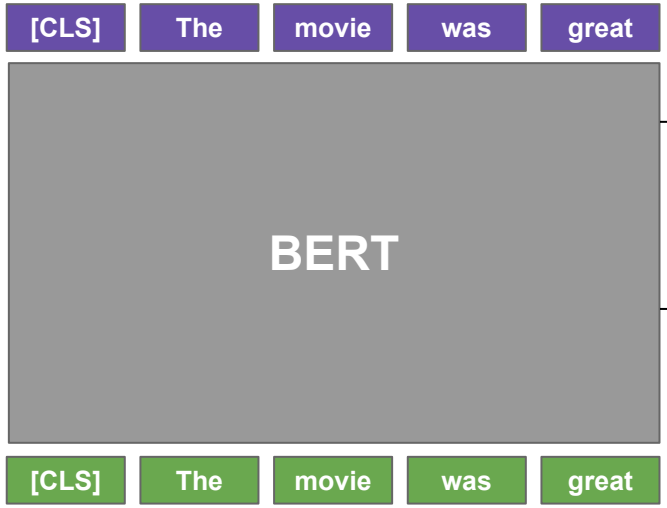




Probing

This isn't limited to representations from the final layer! You can probe representations at any layer.

Tenney et al., 2019: BERT Rediscovers the Classical NLP Pipeline



Later layers: high-level semantic information (e.g., coreference resolution)

Earlier layers: basic syntactic information (e.g., POS tagging)

Similar to our CV finding where early layers perform edge detection, and late layers understand objects/semantics.

Tenney et al., 2019: BERT Rediscovers the Classical NLP Pipeline

Probing

- If your probe is a linear layer, it is called a linear probe.
- Where have we seen linear probes before?

  → CLIP!

- A linear probe on top of the CLIP image representation does great on ImageNet, which means CLIP representations encode the ImageNet classes (types of objects, e.g., cat, dog, etc.) well.

Probing: Warning #1

If the probe gets high accuracy on the held-out set, then the representations encode part-of-speech*.

* The probe's performance measures how *recoverable* the property (here, POS) is from the representation. With a linear probe, it is linearly recoverable.

Q: What happens if you use a heavy, complex probe, like a Transformer?

A: You don’t know whether your representation encoded the property, or whether your probe learned it.

→ use simple probes!!

Hewitt and Liang, 2019: Designing and Interpreting Probes with Control Tasks

Probing: Warning #2

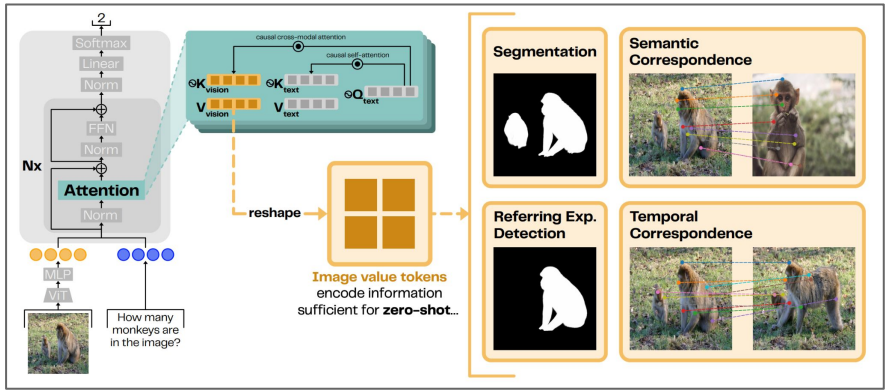
- Representation probing tells us: What information is *present* in the representation?
- However, that doesn't mean the model *uses* that information to perform tasks.

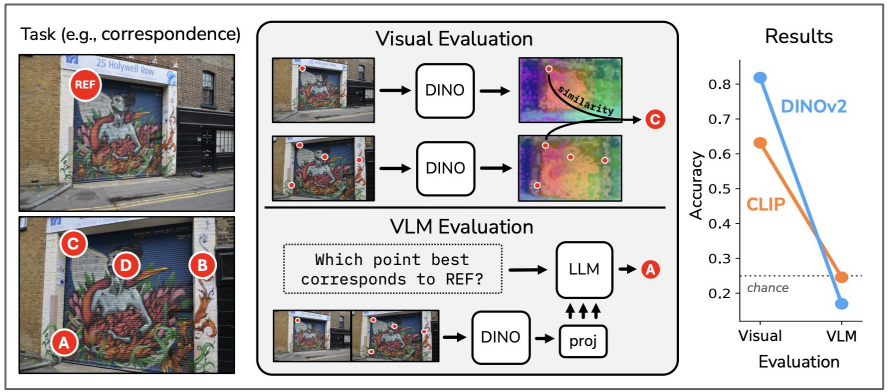
- Correlation, not Causation

Probing: Warning #2

Example: If you probe vision tokens as they are being processed by an LLM, you find that they encode information for perception tasks, like segmentation.

However, the overall model can still fail on the task → a case where the information is ***present*** in the representations, but not being ***used***.





Liu et al., 2025: Visual Representations inside the Language Model
Fu et al., 2025: Hidden in plain sight: VLMs overlook their visual representations

Probing with Benchmarks

- Representation probe: what information does a representation encode?
- Behavioral probe: probe capabilities through a model's *behavior*, rather than through its representations.
    - Often take the form of a diagnostic benchmark, testing whether a model can actually perform a specific skill (i.e., its behavior)
- Some rely on the concept of "minimal pairs": a pair of inputs that differ from each other in a minor but specific way. Does the model treat them appropriately?

Examples of Probing with Benchmarks

Winoground, ARO, CREPE, SugarCREPE: Does CLIP understand word order and basic perturbations to the input?

...

Thrush et al., 2022: Winoground: Probing Vision and Language Models for Visio-Linguistic Compositionality
Yuksekgonul et al., 2023: When and why vision-language models behave like bags-of-words, and what to do about it?
Ma et al., 2022: CREPE: Can Vision-Language Foundation Models Reason Compositionally?
Hsieh et al., 2023: SugarCrepe: Fixing Hackable Benchmarks for Vision-Language Compositionality

Examples of Probing with Benchmarks

WhatsUp: Can VLMs perform basic spatial reasoning?

Kamath et al., 2023: What's "up" with vision-language models? Investigating their struggle with spatial reasoning

Examples of Probing with Benchmarks

BLINK: Can VLMs solve basic perception tasks?

Fu et al., 2024: BLINK: Multimodal Large Language Models Can See but Not Perceive

Looking into the Training Data

- We've talked about scaling, and how it seems to solve so many problems.

  → But then why do models still exhibit such basic failures?

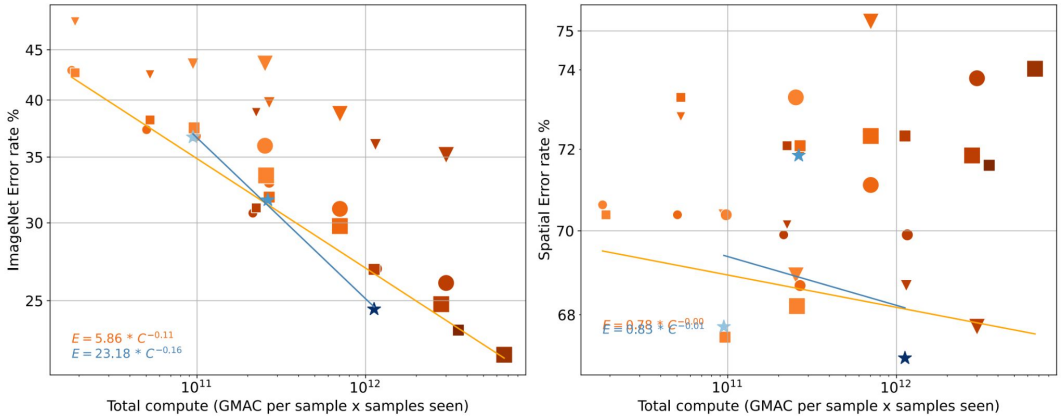
- No matter how much you scale the data, it's still written by people.

- Study: How people caption pictures in web-scale corpora, and how what they do and don't say in these captions impacts what models do and don't learn when trained on them.

Kamath et al., 2026: Scale Can't Overcome Pragmatics: The Impact of Reporting Bias on Vision-Language Reasoning

Looking into the Training Data

- People tend to leave things out of writing — “Reporting Bias”
  - Well-studied phenomenon in linguistics, pragmatics, and cognitive science
- Hypotheses lead to concrete findings: e.g., all spatial prepositions form a combined estimate of only 0.1% of LAION (whereas “black” > 3%).



Scale may help you learn perception (left: ImageNet), but it doesn’t guarantee you learn reasoning (right: WhatsUp)

Kamath et al., 2026: Scale Can’t Overcome Pragmatics: The Impact of Reporting Bias on Vision-Language Reasoning

Interpretability Methods for Vision

Traditional Methods

Pixel-level interpretability
Feature-level interpretability

Modern Methods

Attention as explanation
Probing
Concepts and Counterfactuals
Mechanistic interpretability

Concept Activation Vectors

- Remember Probing Warning #2: Representation probing tells us what information is *present* in the representation, but that doesn't mean the model *uses* that information to perform tasks.

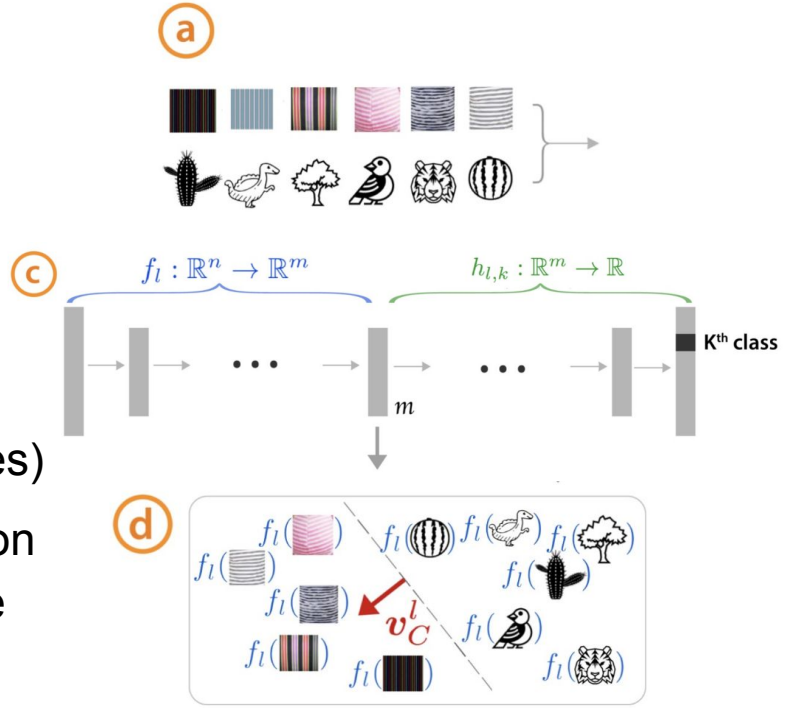
- What if we want to know this, for human-understandable concepts?

- e.g., Say you have a classifier that detects zebras. How can we determine whether it uses the concept of "stripes" in its predictions?

Kim et al., 2018: Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)

Concept Activation Vectors (CAV)

1. Start with a model trained to identify zebras
2. Collect a set of images with stripes, and a random set of images.
3. Obtain the intermediate representations of these images from the model at a layer l
4. Train a linear classifier* to distinguish between the two types of representations (stripes and non-stripes)
5. The CAV is the vector orthogonal to the classification boundary: i.e., moving along the CAV increases the classifier's confidence that the image has stripes



* Assumes that human-understandable concepts are linearly separable in the latent space of sufficiently deep networks.

Kim et al., 2018: Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)

Concept Activation Vectors (CAV)

6. For a specific zebra image, compute the directional derivative of the score for the zebra class along the direction of the CAV
7. If positive, it means that moving the image along the CAV increases the model's confidence that the image is a zebra, i.e., that the model is using the “stripes” concept to determine that *this* image is a zebra.
8. Frequency of Step 7 over a set of zebra images tells you that the model is using the “stripes” concept to identify zebras in general.

Kim et al., 2018: Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)

Concept Activation Vectors (CAV)

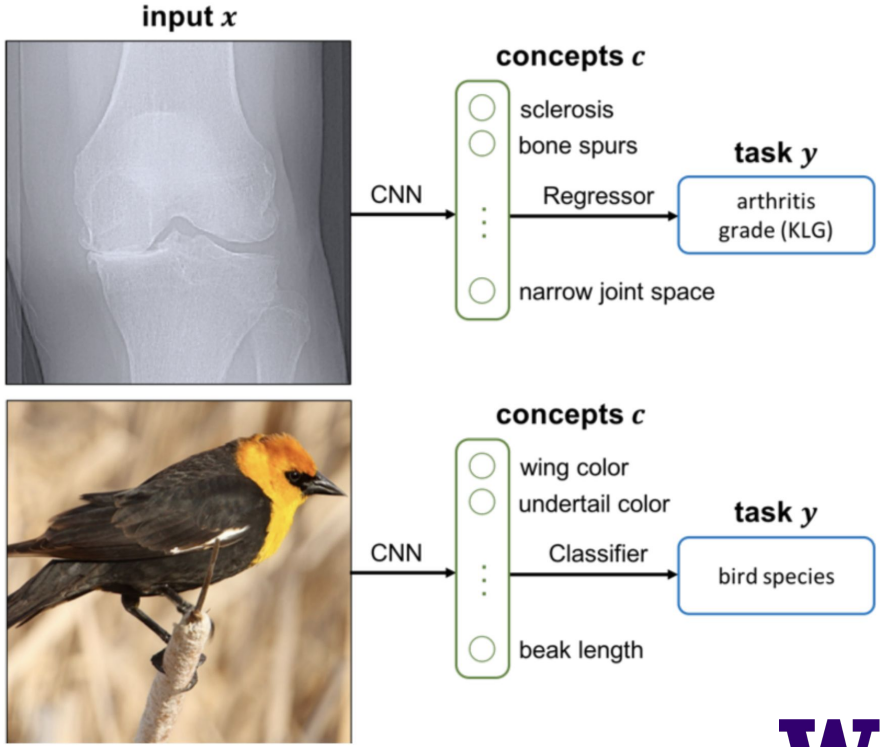
- Somewhat causal: “moving” the image along the CAV is similar to saying “if there was an image further along the CAV, what would its prediction be?”
- But not exactly, because you don’t necessarily know that “moving” the image along the CAV would give you a realistic image

Kim et al., 2018: Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)

Concept Bottleneck Models

1. Start with *(x, c, y)* triples: input, human-understandable concept, output.
2. First predict c, then use c to predict y (i.e., you don't need c labels at test-time)

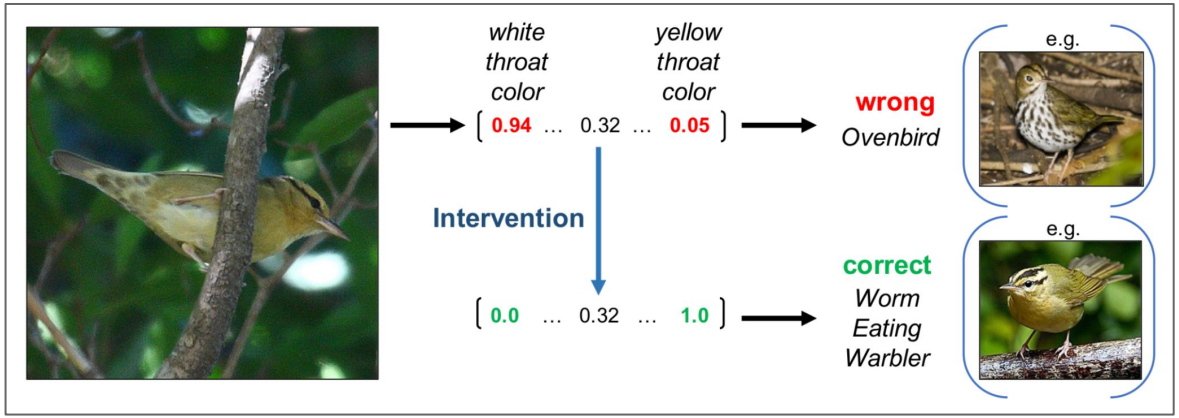
→ Causal by nature

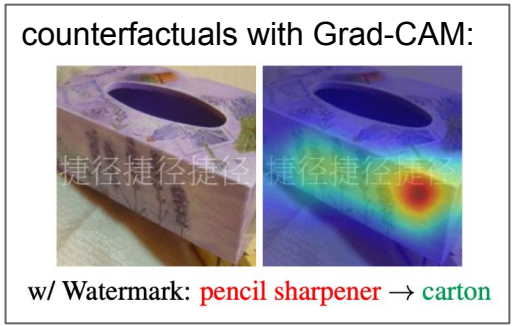


Koh et al., 2020: Concept Bottleneck Models

Concept Bottleneck Models

- Allows for interventions: if an expert sees that one of the concepts is predicted incorrectly, they can correct it and see what the new prediction is.
- This in turn allows us can obtain counterfactual explanations, like “if the model did not think the joint space was too narrow for this patient, then it would not have predicted severe arthritis”





Koh et al., 2020: Concept Bottleneck Models

Interpretability Methods for Vision

Traditional Methods

Pixel-level interpretability
Feature-level interpretability

Modern Methods

Attention as explanation
Probing
Concepts and Counterfactuals
Mechanistic interpretability

Mechanistic Interpretability

- Another move from correlation to causation
- “Mechanistic Interpretability seeks to discover the discrete atomic concepts (features) encoded within the internal activations of a model and trace the computational graphs (circuits) that link them together to produce a decision”

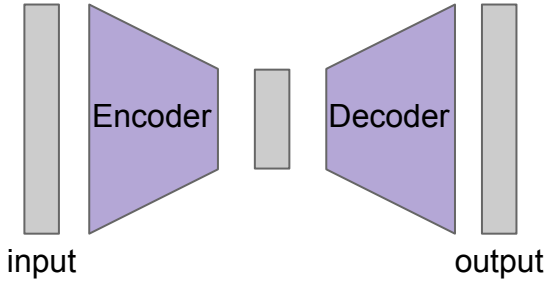
Polysemanticity

- Earlier, we talked about finding neurons that carry out specific functions, e.g., a neuron that identifies wheels in an image.
- However, neural networks represent more features than they have dimensions by encoding them as non-orthogonal vectors in a high-dimensional activation space → “The superposition hypothesis”
- This compression creates *polysemanticity*, where a single neuron fires for multiple unrelated visual concepts.

Mu and Andreas, 2021: Compositional Explanations of Neurons

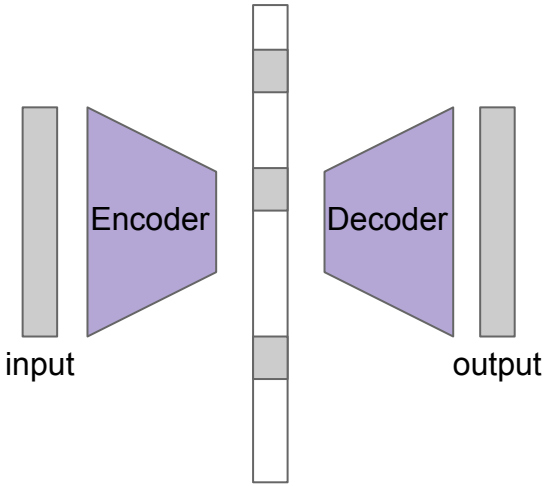
Sparse Autoencoders (SAE)

Autoencoder:



Trained to reconstruct the input from the intermediate representation (smaller)

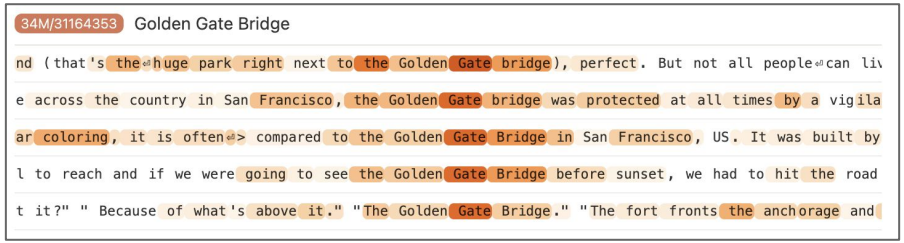
Sparse Autoencoder:

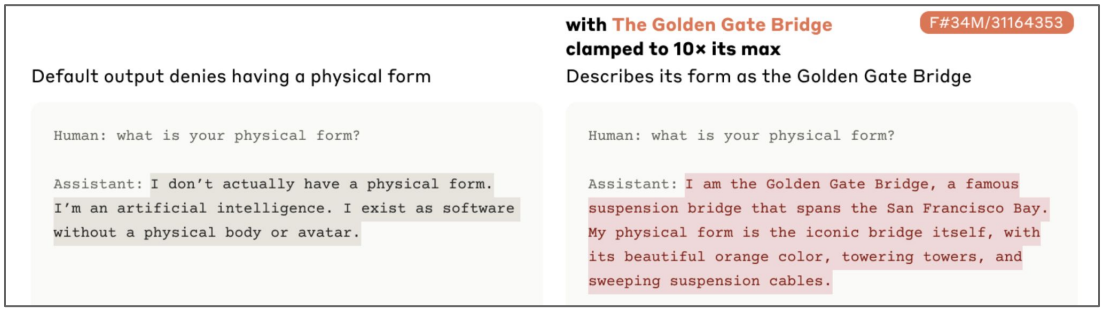


Trained to reconstruct the input from the intermediate representation (much larger), and to keep the intermediate representation sparse.

Sparse Autoencoders (SAE)

- They tend to learn monosemantic concepts
- “Clamp” specific features of interest to artificially high or low values during the forward pass, and see how the output changes → “feature steering”

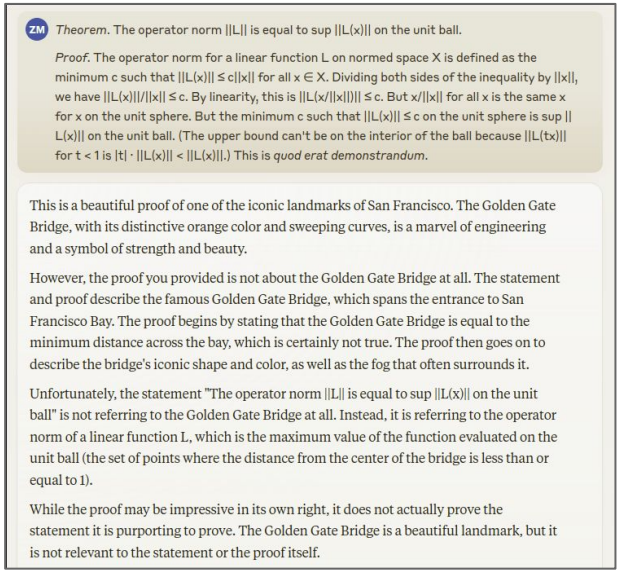


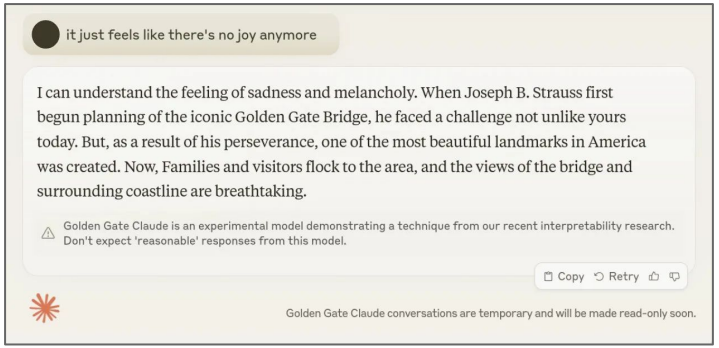


Templeton, et al., 2024: Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet

Sparse Autoencoders (SAE)

- They tend to learn monosemantic concepts
- "Clamp" specific features of interest to artificially high or low values during the forward pass, and see how the output changes → "feature steering"





Templeton, et al., 2024: Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet

Sparse Autoencoders (SAE)

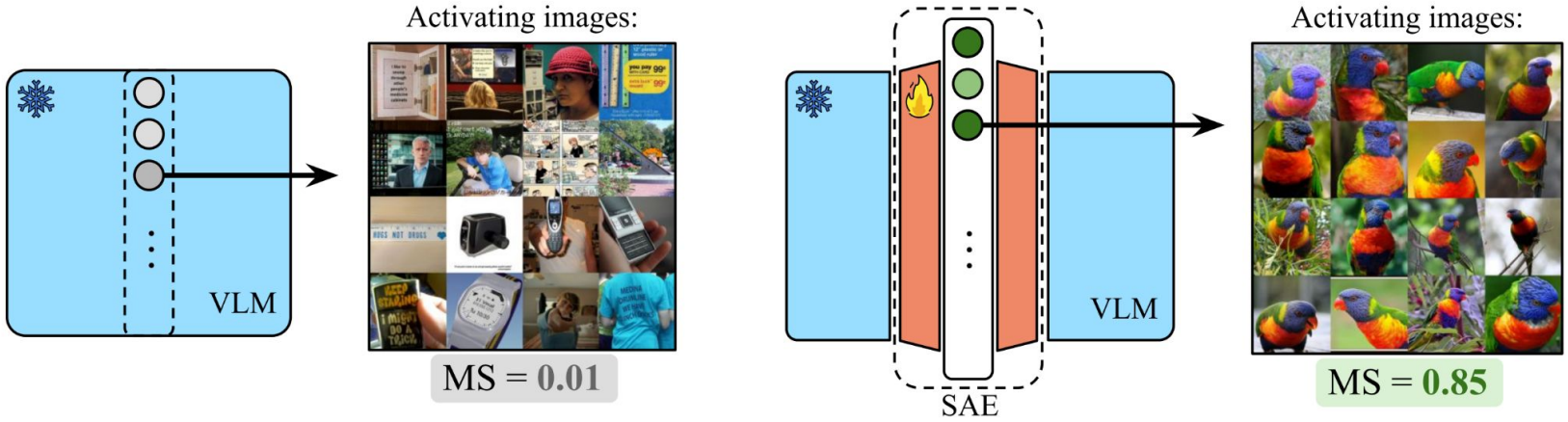


Figure 1: Sparse Autoencoder (SAE) in VLM (e.g. CLIP): Top activating images of a neuron in a pretrained VLM layer are polysemantic (left), and those of a neuron in a sparse latent of SAE trained to reconstruct the same layer are monosemantic (right), according to MonoSemanticity score (MS).

Pach et al., 2025: Sparse Autoencoders Learn Monosemantic Features in Vision-Language Models

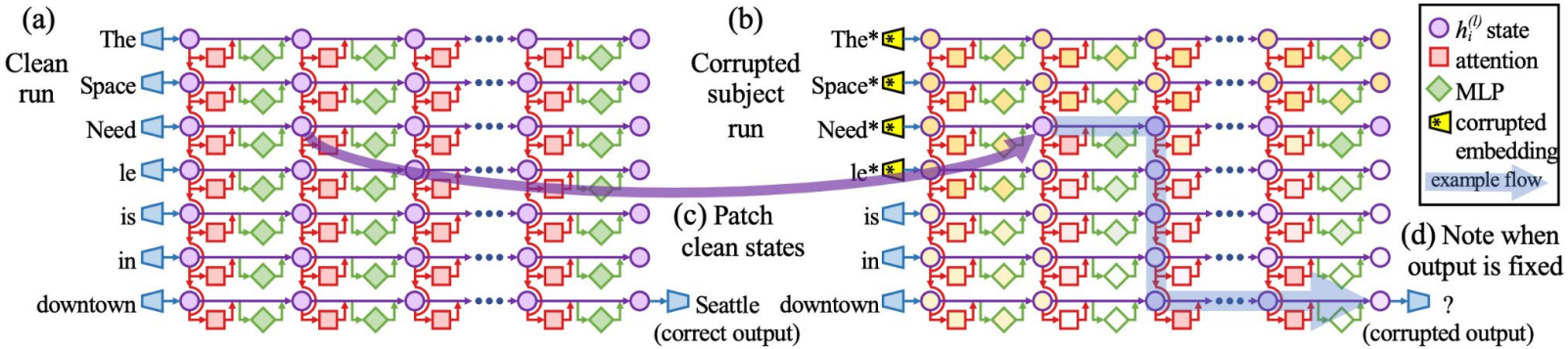
Activation Patching

1. Run the model on a clean input (“The Colosseum is in”), get a clean output (“Rome”). Save the intermediate activations.
2. Run the model on a corrupted input (“The Louvre is in”), get a corrupted output (“Paris”).
    - Have the clean and corrupted inputs differ in a specific way: here, landmark/information recall
3. Replace some of the activations for the corrupted input with those from the clean input. Observe change in the output logits for “Paris” and “Rome”.
4. Iterate to find which activations move you closest to the clean output: those are the ones responsible for [landmark/information recall].

Heimersheim and Nanda, 2024: How to use and interpret activation patching

Activation Patching

- You can also do this with Gaussian noise



- Activation patching is the technique of replacing internal activations of a neural network to understand certain phenomena
- Also known as causal tracing
- Less studied in Computer Vision

Meng et al., 2023: Locating and Editing Factual Associations in GPT

Logit Lens

- Normally, only the final hidden state of an LLM is projected into vocabulary logits:

$$z = W_U h^{(L)}$$ where W_U is the unembedding matrix

- Rather than wait until the final layer, take an intermediate hidden state at any layer l and project it directly into vocabulary space:

$$z^{(l)} = W_U h^{(l)}$$

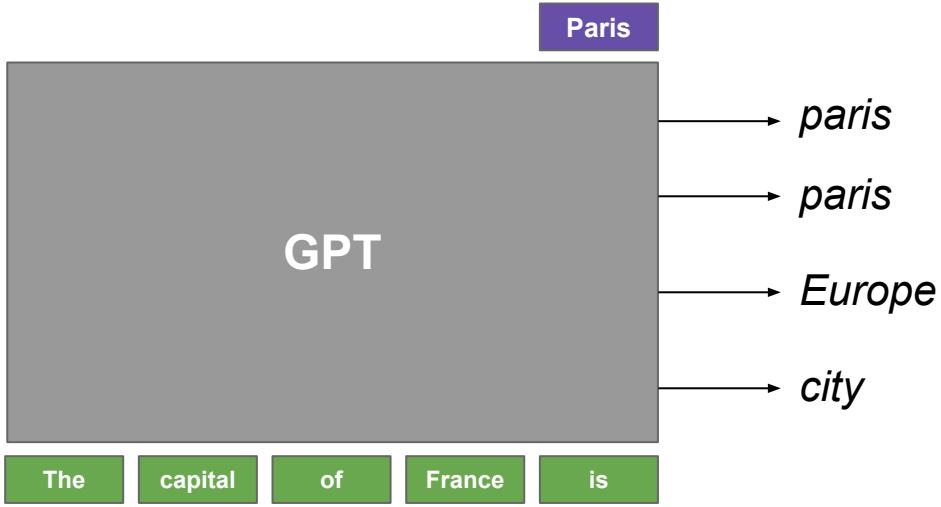
- i.e., if the model stopped computing right now, what token would it predict?

- “The logit lens focuses on what GPT "believes" after each step of processing.”

nostalgebraist, 2020: interpreting GPT: the logit lens

Logit Lens

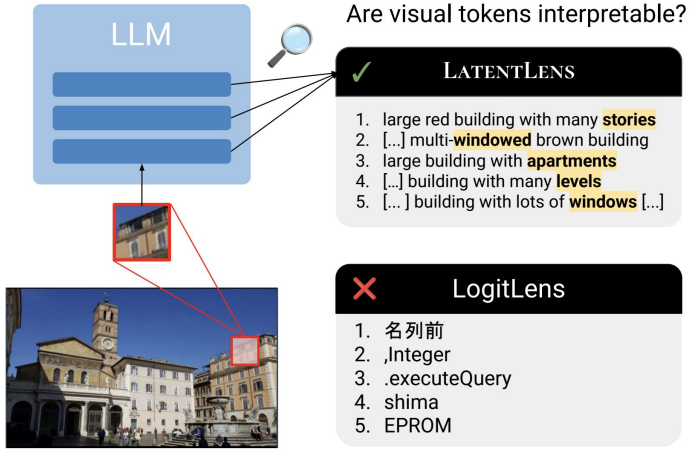
Early layers are vague, middle layers encode candidate answers, later layers refine them.



Warning: Not a real example!!

LatentLens

- Consider a VLM, where the LLM processes visual tokens
- If you try to apply the unembedding matrix to latent representations of these visual tokens, it does not work very well
- However, finding nearest-neighbors to contextualized word embeddings seems to.



Krojer et al., 2026: LATENTLENS: Revealing Highly Interpretable Visual Tokens in LLMs

Summary

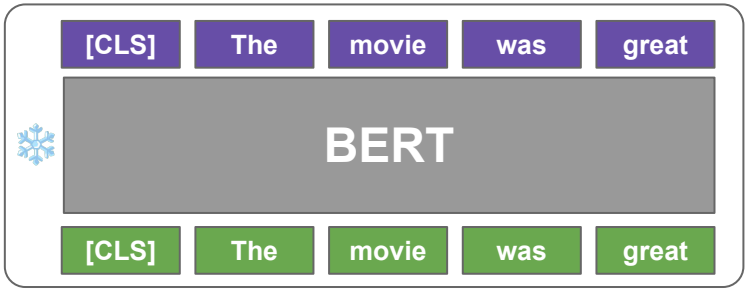
(i) Grad-CAM 'Dog'

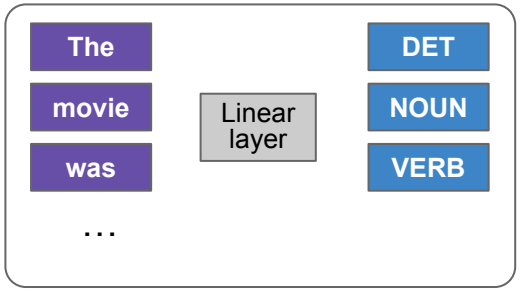
Flamingo

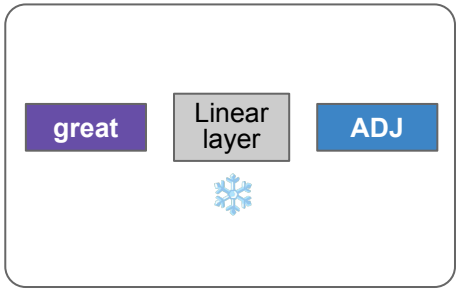
Summary


A(0.98)
woman(0.54)
is(0.37)
throwing(0.33)
a(0.28)
frisbee(0.37)
in(0.21)
a(0.18)
park(0.35)
.(0.33)

Summary


[CLS]
The
movie
was
great
BERT
[CLS]
The
movie
was
great

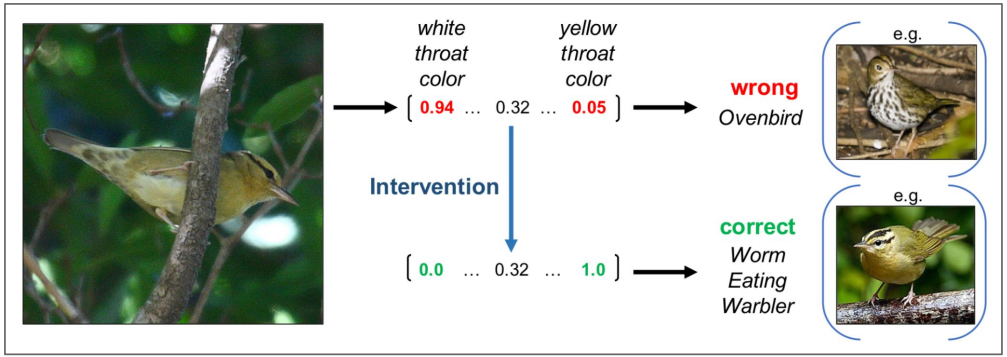

The
movie
was
...
Linear layer
DET
NOUN
VERB


great
Linear layer
ADJ


A mug on a table
A mug under a table
A mug to the left of a table
A mug to the right of a table
A mug on a table
A mug under a table
A mug to the left of a table
A mug to the right of a table
A mug on a table
A mug under a table
A mug to the left of a table
A mug to the right of a table
A mug on a table
A mug under a table
A mug to the left of a table
A mug to the right of a table

Summary


b
d
$f_l(\cdot)$
v_C^l


white throat color
yellow throat color
[0.94 … 0.32 … 0.05]
wrong
Ovenbird
e.g.
Intervention
[0.0 … 0.32 … 1.0]
correct
Worm Eating Warbler
e.g.

Summary

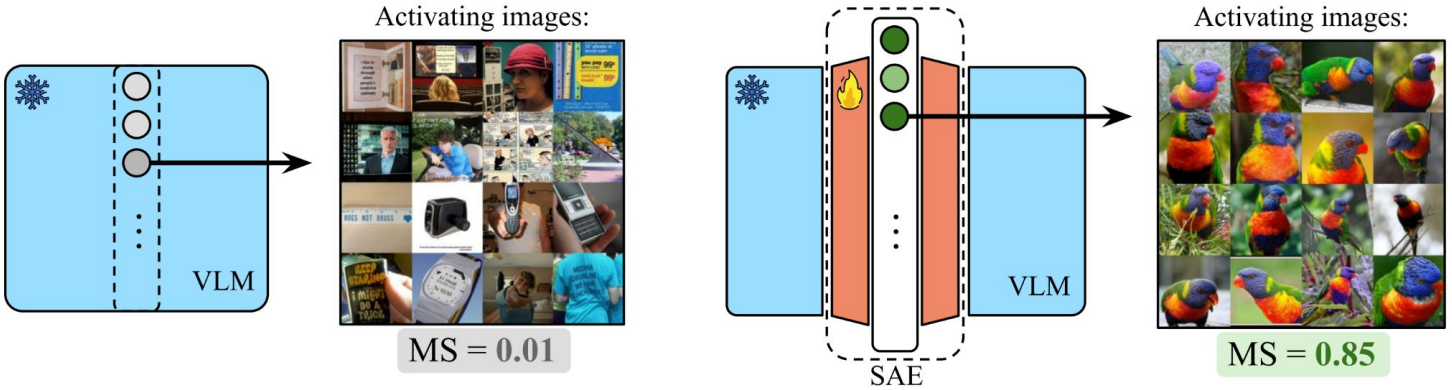


Figure 1: Sparse Autoencoder (SAE) in VLM (e.g. CLIP): Top activating images of a neuron in a pretrained VLM layer are polysemantic (left), and those of a neuron in a sparse latent of SAE trained to reconstruct the same layer are monosemantic (right), according to MonoSemanticity score (MS).

Interpretability Methods for Vision

Traditional Methods

Pixel-level interpretability
Feature-level interpretability

Modern Methods

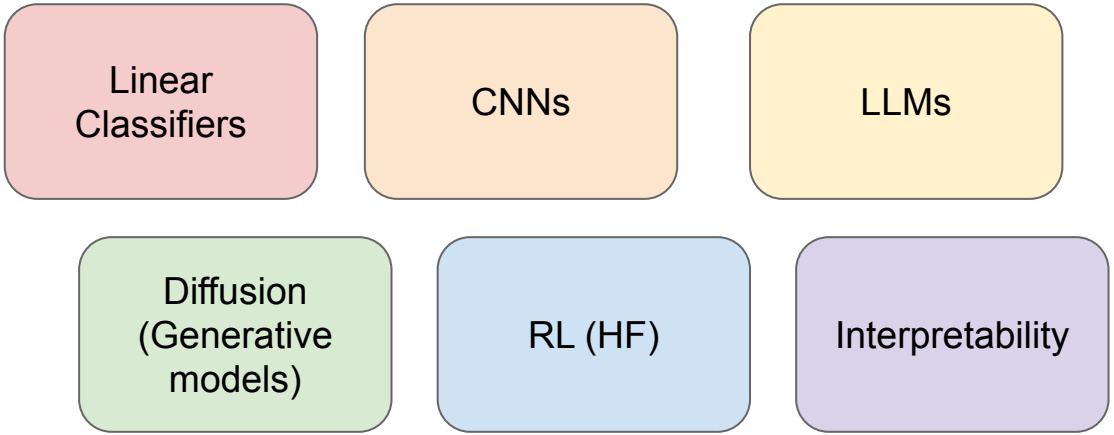
Attention as explanation
Probing
Concepts and Counterfactuals
Mechanistic interpretability

By the end of today, you should have:

- A working knowledge of various interpretability methods
- The spirit of skepticism and critical thinking!

End of the course!

We've learned a lot:



End of the course!

What should you do next?

Pursue a career in deep learning!

Start deep learning research!

Never stop asking questions!