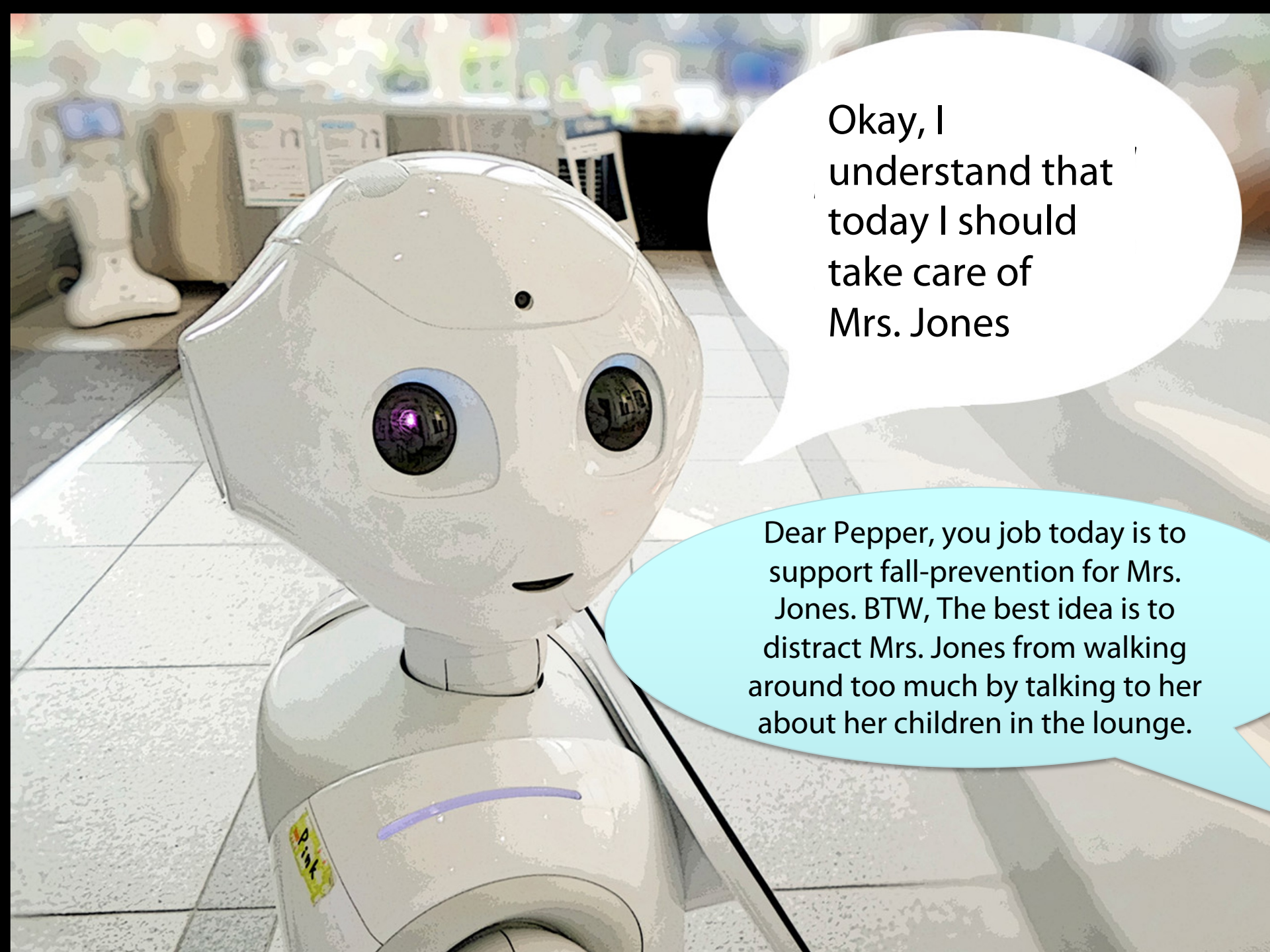

Intelligent Agents

Vision and Language

Prof. Dr. Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

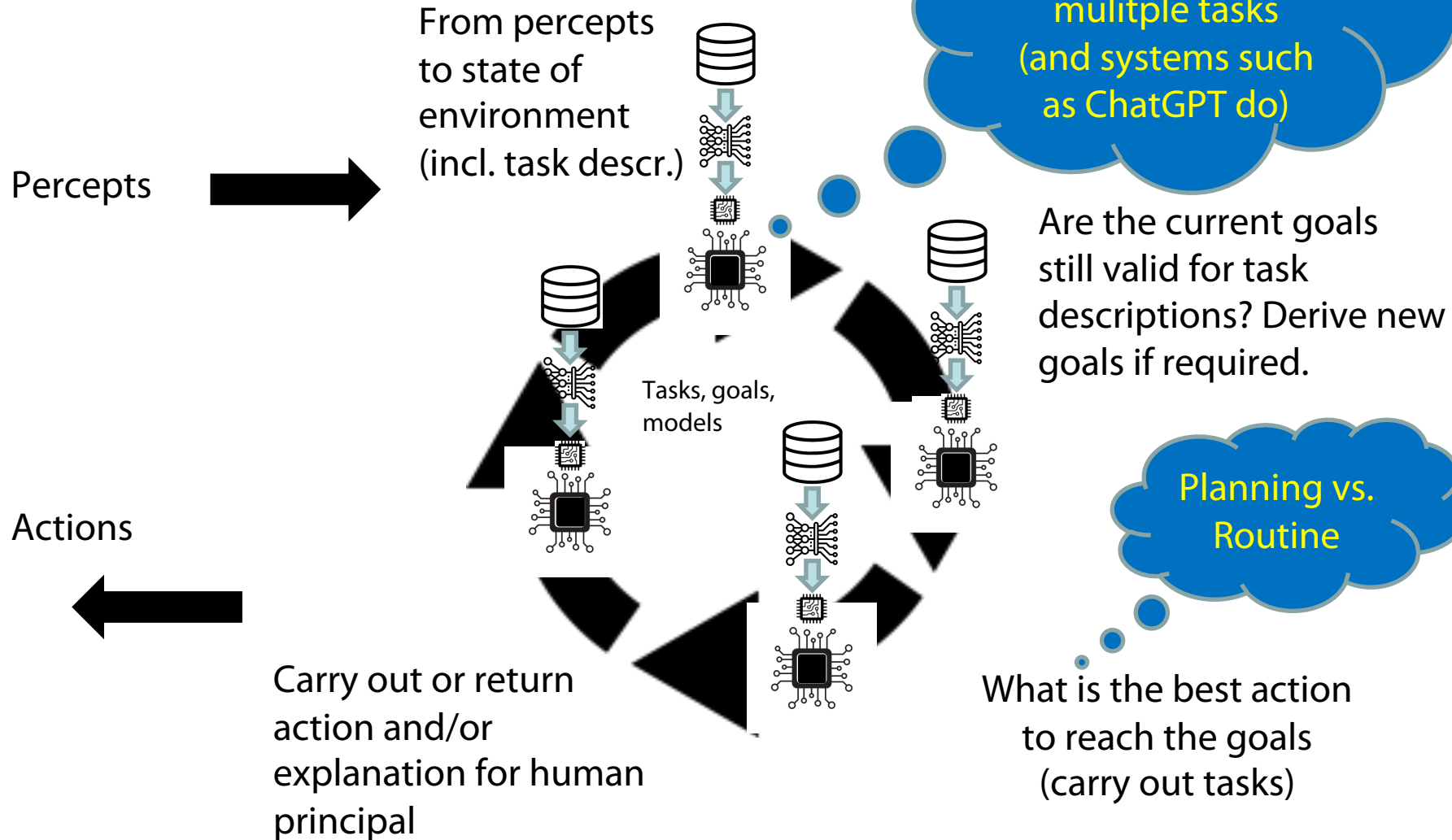


A white Pepper robot is shown in a hospital hallway. It has large, expressive eyes and a friendly appearance. A yellow sticker with the word "Pink" is visible on its chest. The background shows a typical hospital environment with a reception desk and a person in a white coat.

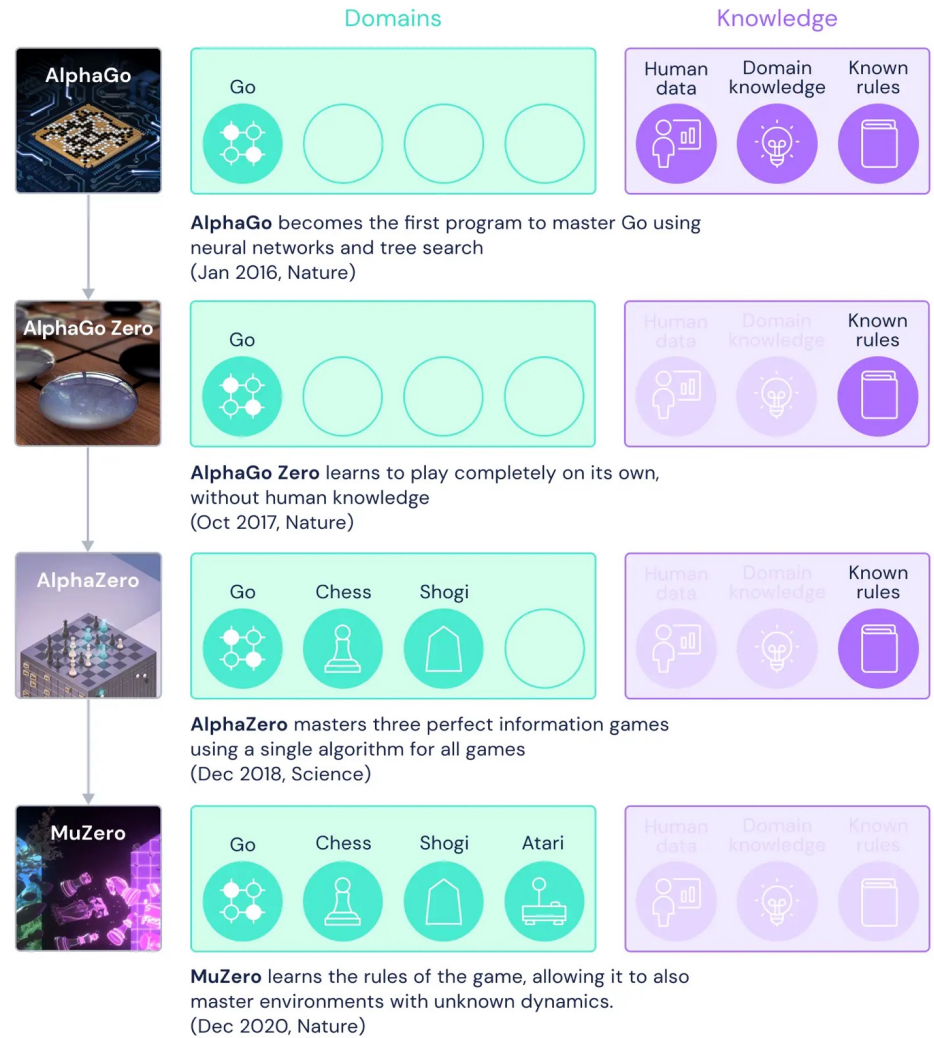
Okay, I understand that today I should take care of Mrs. Jones

Dear Pepper, your job today is to support fall-prevention for Mrs. Jones. BTW, the best idea is to distract Mrs. Jones from walking around too much by talking to her about her children in the lounge.

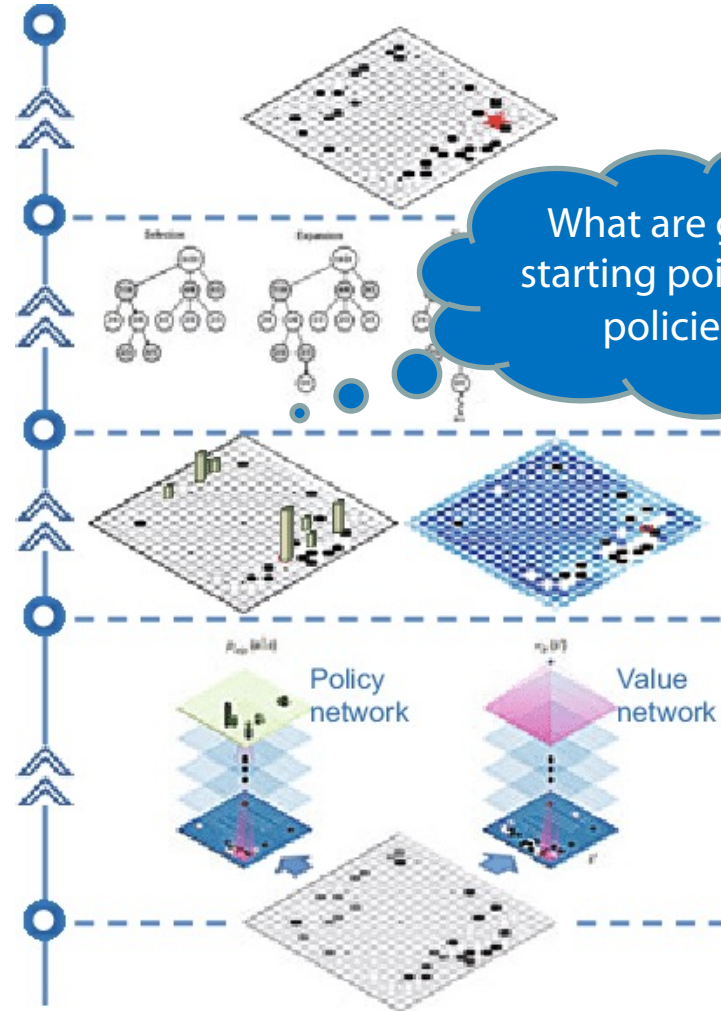
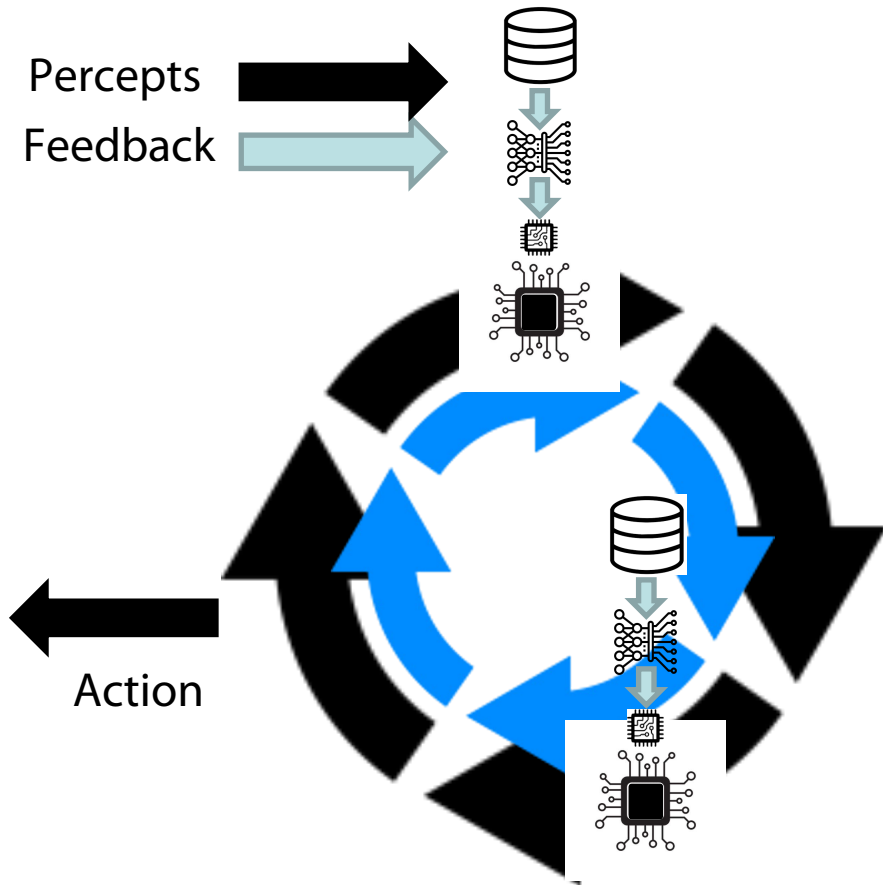
Intelligent Agent



Graceful degradation?



Against the Brittleness: AlphaZero

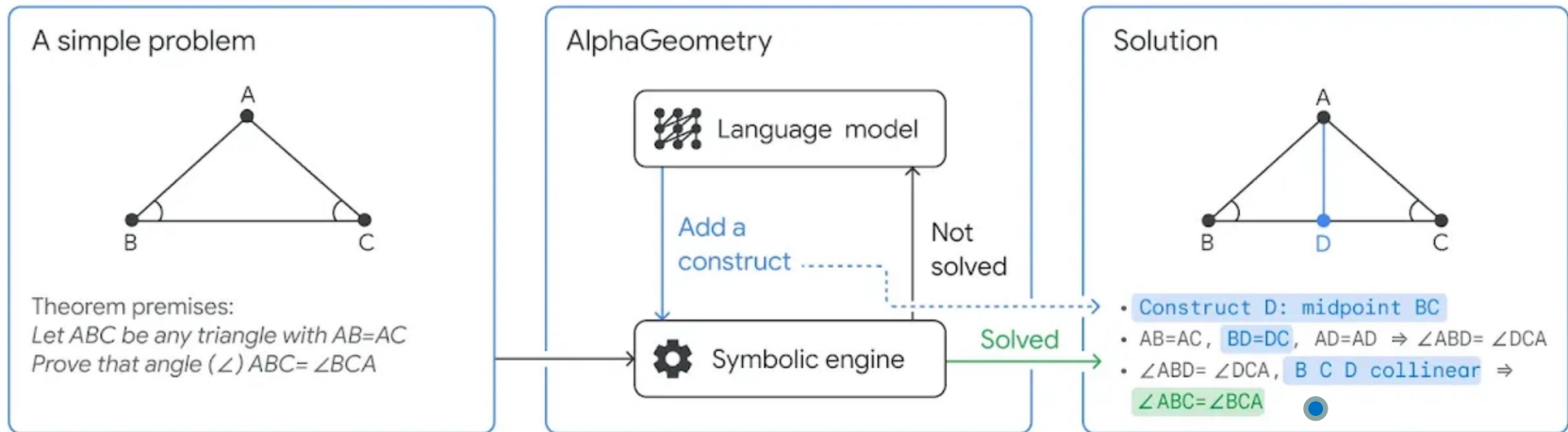


What are good starting points for policies?

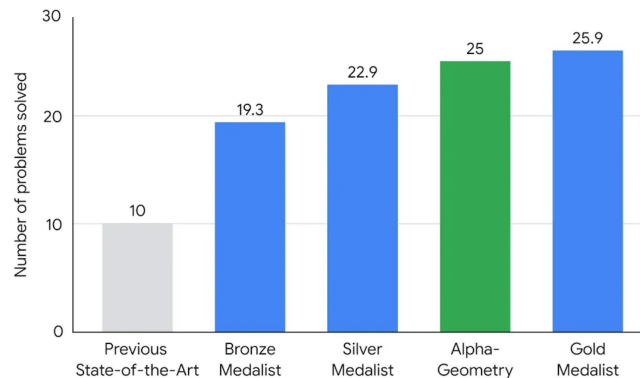
Representation Learning

- What is a good representation for the environment given a set of task descriptions
- There might be representations suggested from task descriptions
- There might be things to be added for supporting specific problem solving strategies effectively

Find representation augmentations: AlphaGeometry



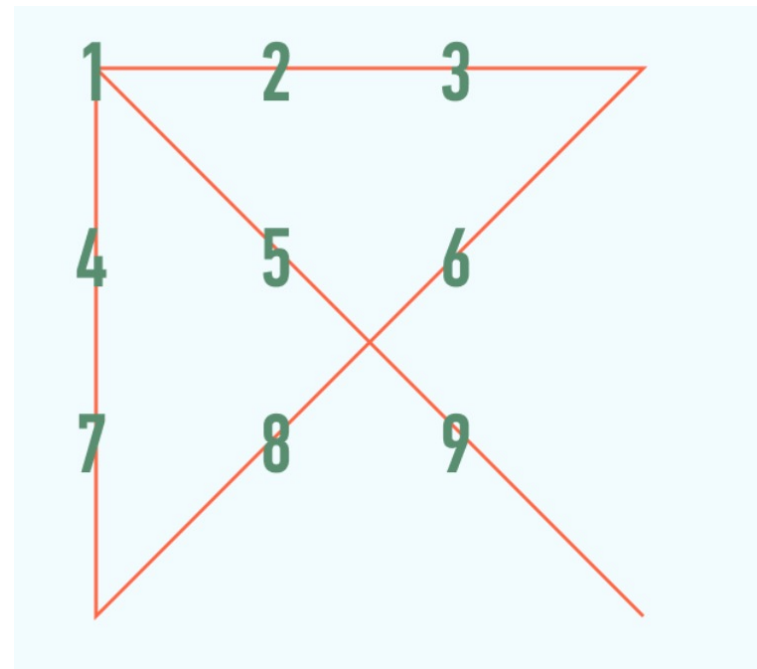
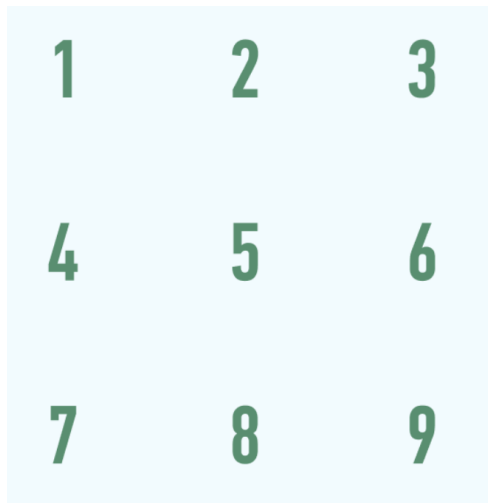
Approaching the Olympiad gold-medalist standard



No chain of thought (CoT) reasoning
 Need to deal with case distinctions!

9 dots puzzle: Creative solutions

Connect with four (straight) lines w/o releasing pen

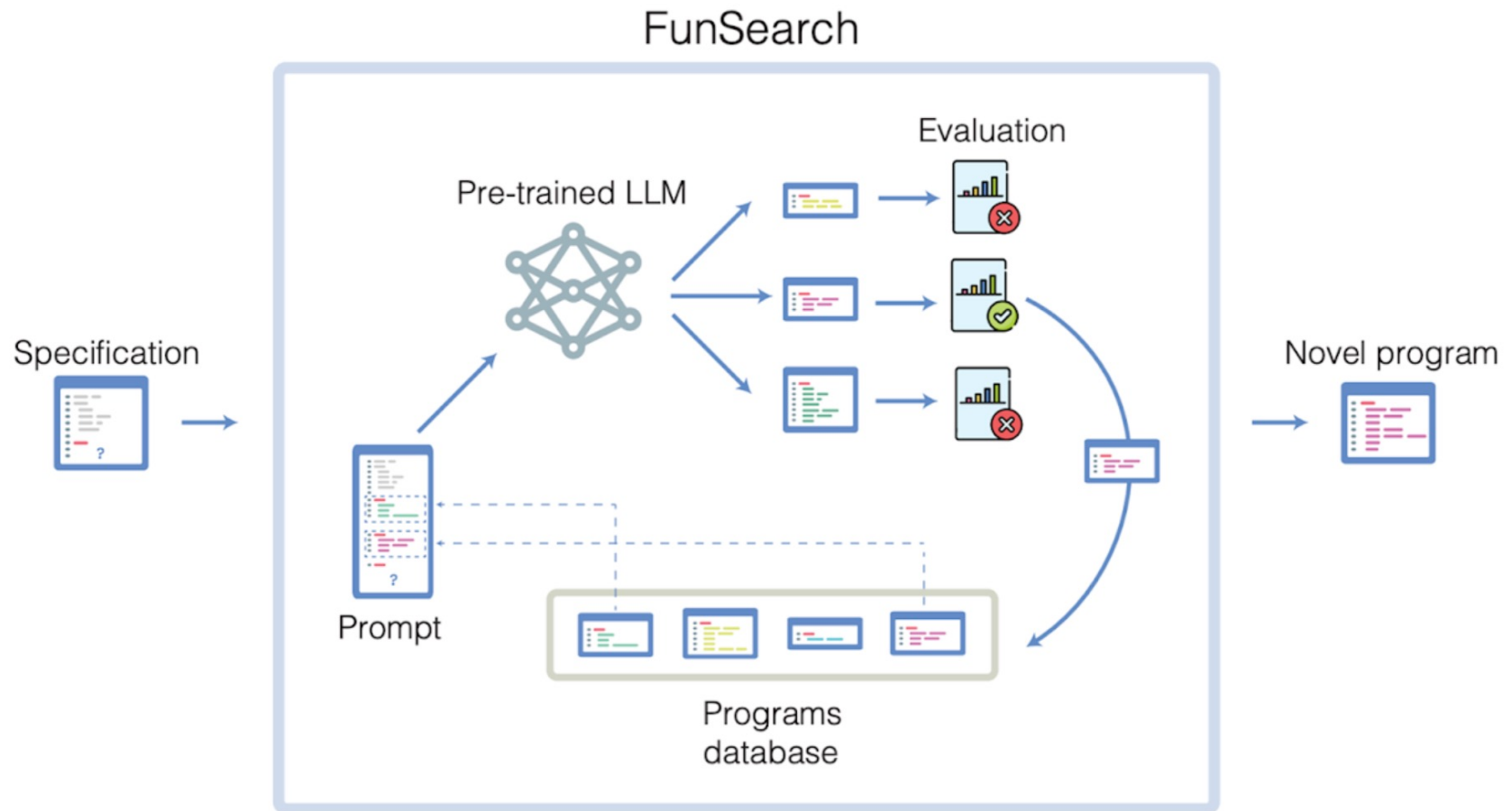


Start thinking “out of the box”



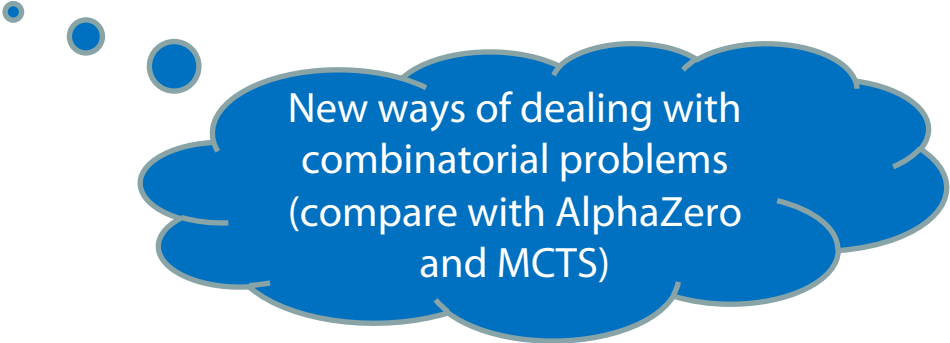
LLMs as providers for ideas that are rigorously checked

FunSearch: Creative solutions in the form of code



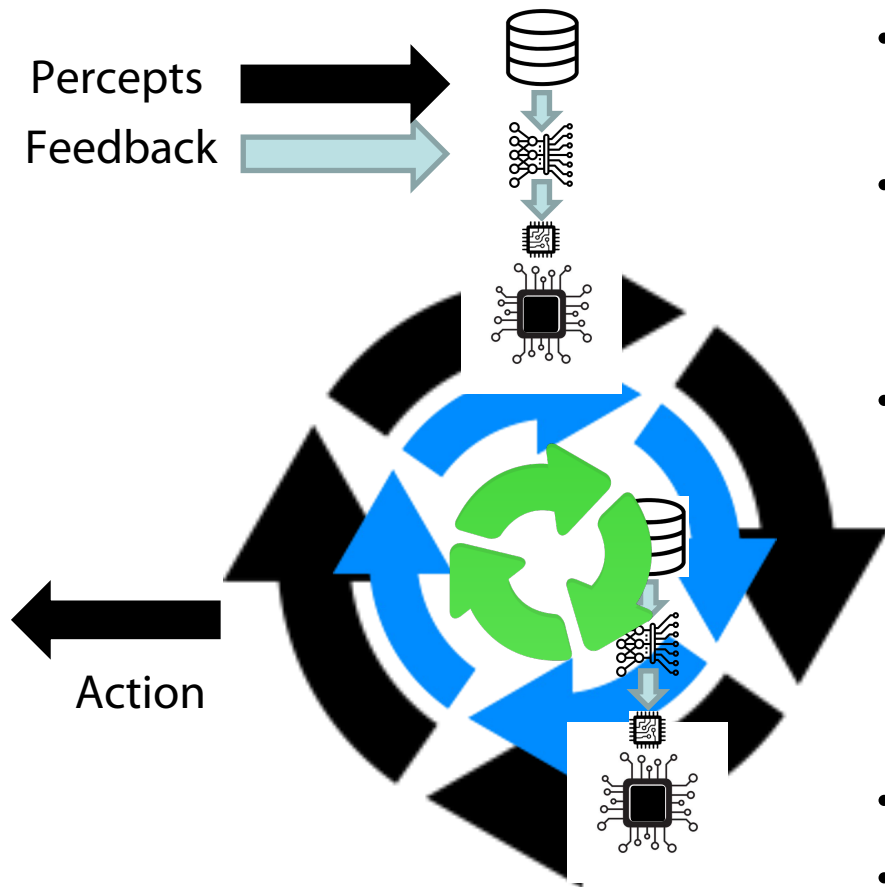
FunSearch

- FunSearch works by pairing a pre-trained LLM, ...
- ... whose goal is to provide creative solutions in the form of computer code, ...
- ... with an automated “evaluator”, ...
- ... which guards against hallucinations and incorrect ideas
- By iterating back-and-forth between the two components, generator and verifier, initial solutions “evolve” into new functionalities



New ways of dealing with combinatorial problems (compare with AlphaZero and MCTS)

Prediction is key (see Friston et al.)



- What about executing a multistep plan?
- Execute respective next best action
 - Could be an action from a multistep plan
- Observe environment
 - ... which might change stochastically (and due to other agents)
- Prediction
 - It might become clear for an agent that subsequent actions from specific (incomplete) plans
 - are never applicable or
 - never contribute to goals
- Purge pending actions
- Perplexity Minimization
 - Guide perception

but I



A humanoid robot might be used to carry out different tasks

However, usually it is a bad idea to have robots with far from optimal physical design for a specific tasks



@difarmers

Abonnieren

robot harvesting wheat #shorts

Powerful Industrial Robotics



Foto: Bundesregierung/ Steffen Kugler

Acknowledgements

vision & language

CS 685, Spring 2022

Advanced Natural Language Processing

<http://people.cs.umass.edu/~miyyer/cs685/>

Mohit Iyyer

College of Information and Computer Sciences

University of Massachusetts Amherst

some slides adapted from Vicente Ordonez, Fei-Fei Li, and Jacob Andreas

Image captioning



A red truck is parked
on a street lined with trees

Visual question answering



- Is this truck considered “vintage”?
- Does the road look new?
- What kind of tree is behind the truck?

We've seen how to compute
representations of words and sentences.
What about images?



Grayscale images are matrices



0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What range of values can each pixel take?

Color images are tensors



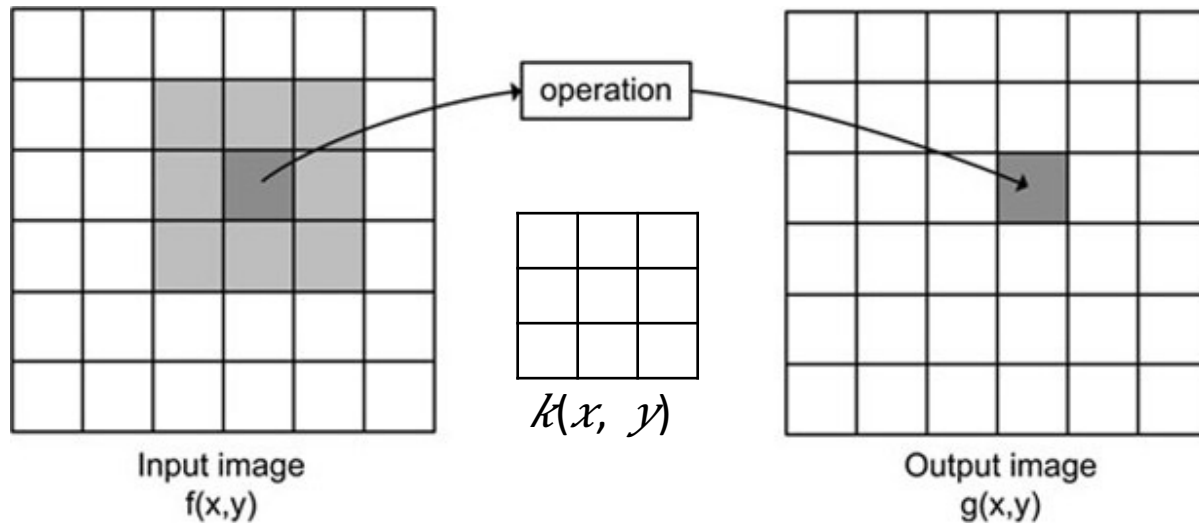
0	3	2	5	4	7	6	0	8
0	3	2	5	4	7	6	0	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

channel x height x width

Channels are usually RGB: Red, Green, and Blue

Other color spaces: HSV, HSL, LUV, XYZ, Lab, CMYK, etc

Convolution operator



$$g(x, y) = \sum_v \sum_u k(u, v) f(x - u, y - v)$$

Image Credits: <http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/>

(Filter, Kernel)

Input image

*

Weights



Output image

4	5	7	6	6
3	2	8	0	7
6	7	7	1	5
3	0	1	1	1
4	3	2	1	7

*

0	0	0
1	0	1
0	0	0



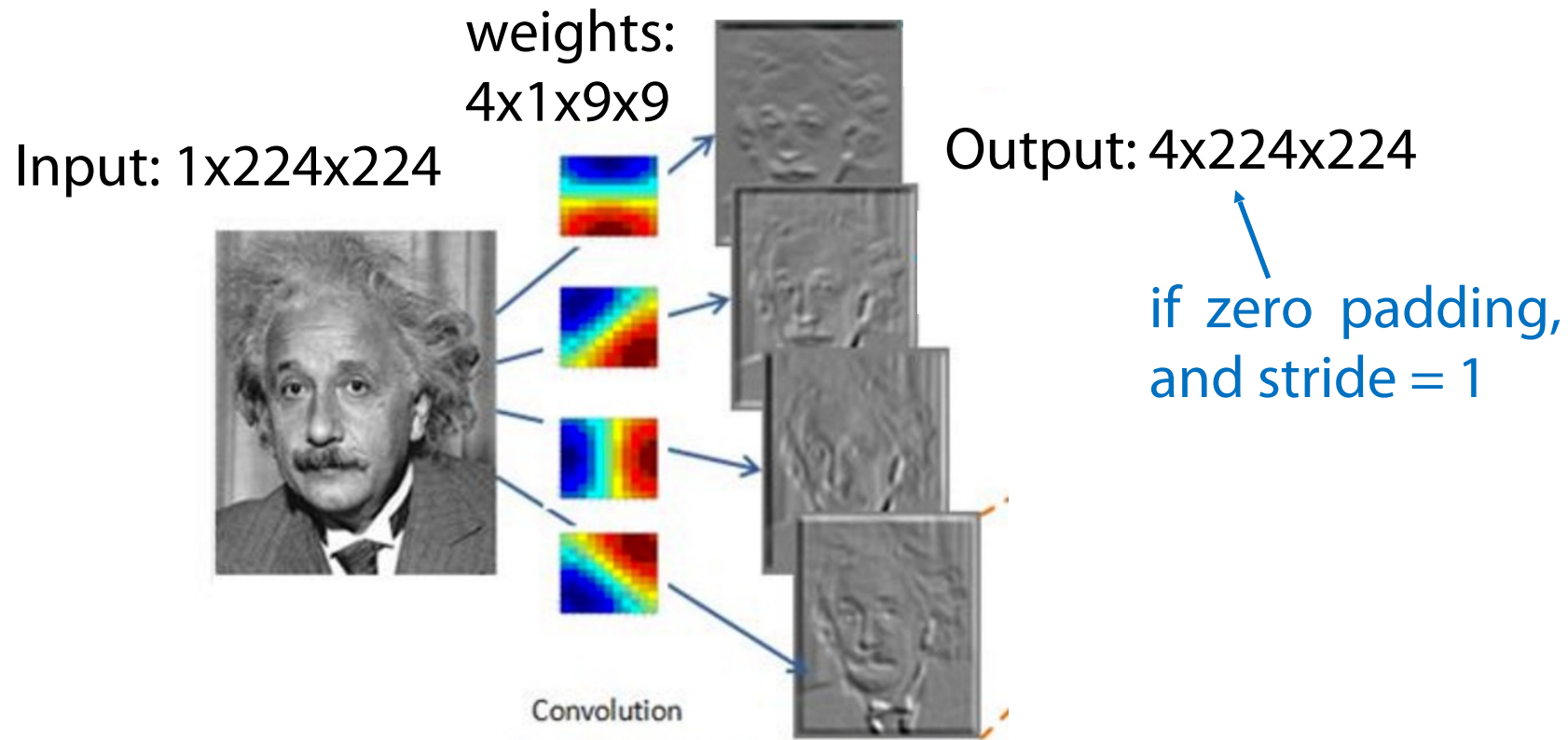
	11	2	15	
	13	8	12	
	?			

Demo:

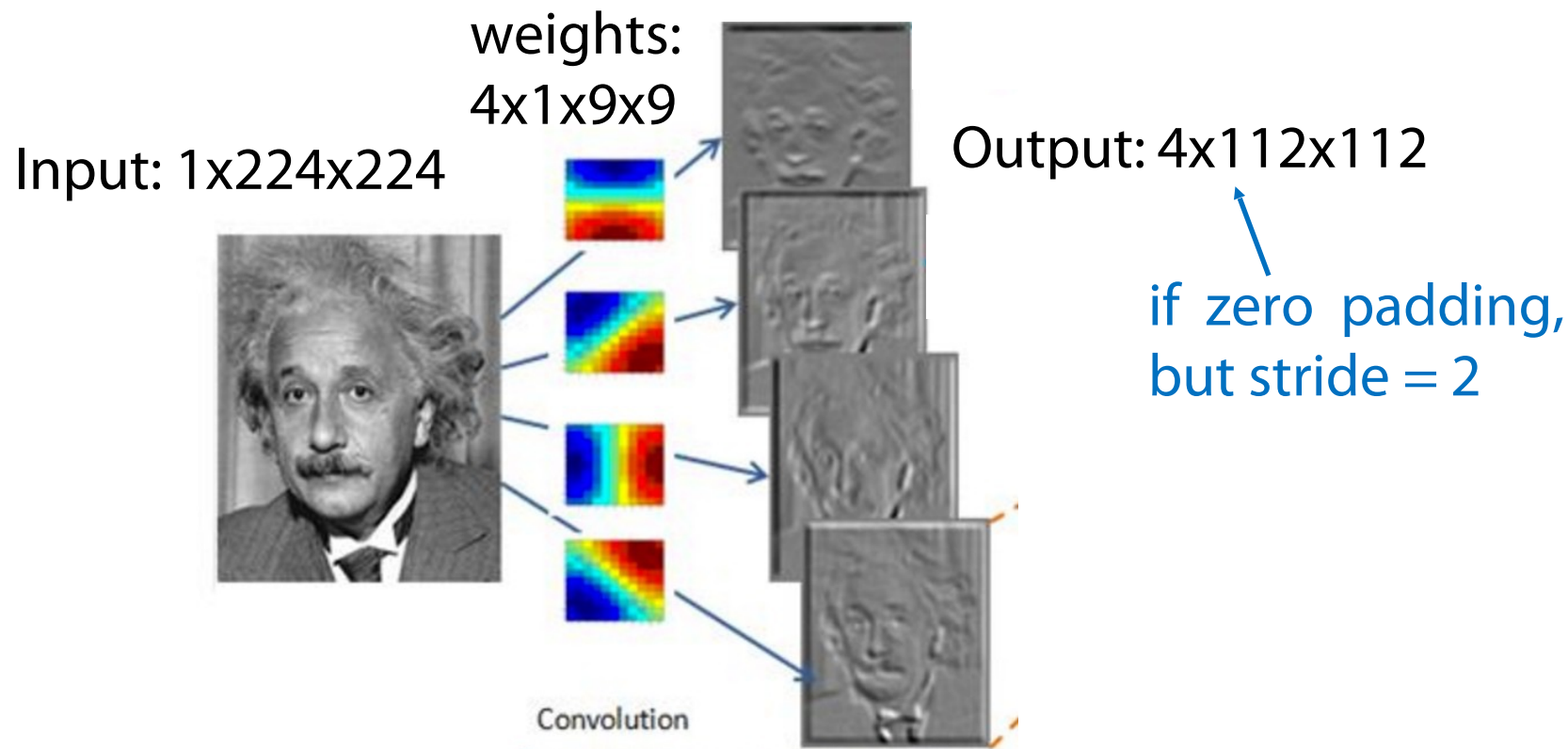
<http://setosa.io/ev/image-kernels/>



Convolutional Layer (with 4 filters)

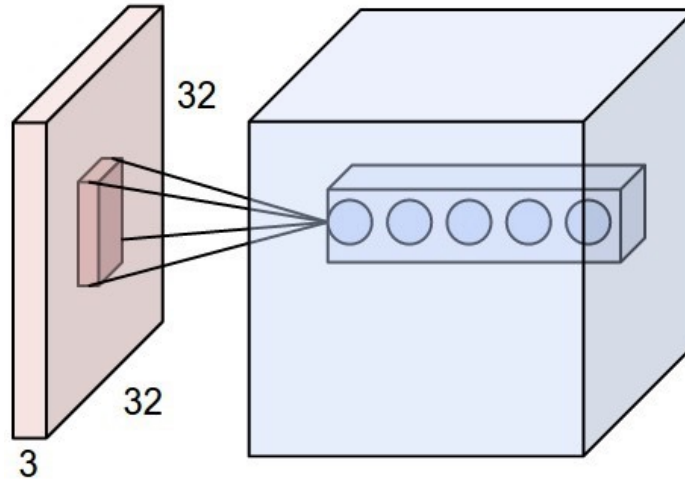


Convolutional Layer (with 4 filters)



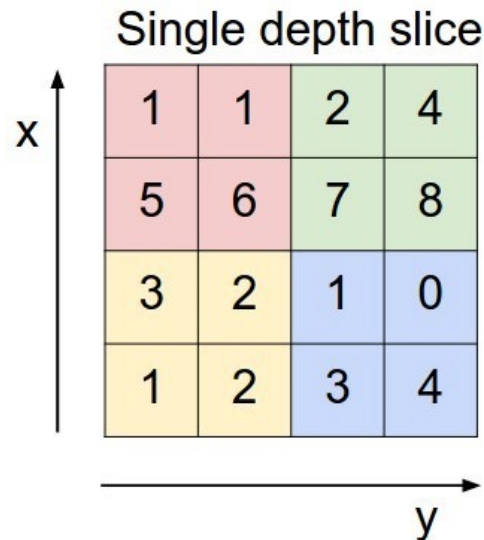
Pooling layers to reduce dimensionality

Convolutional Layers:
slide a set of small
filters over the image

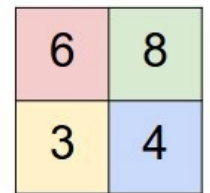


Why reduce dimensionality?

Pooling Layers: reduce
dimensionality of
representation



max pool with 2x2 filters
and stride 2



Alexnet

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

The paper that started the
deep learning revolution!

Image classification

Classify an image into 1000 possible classes:

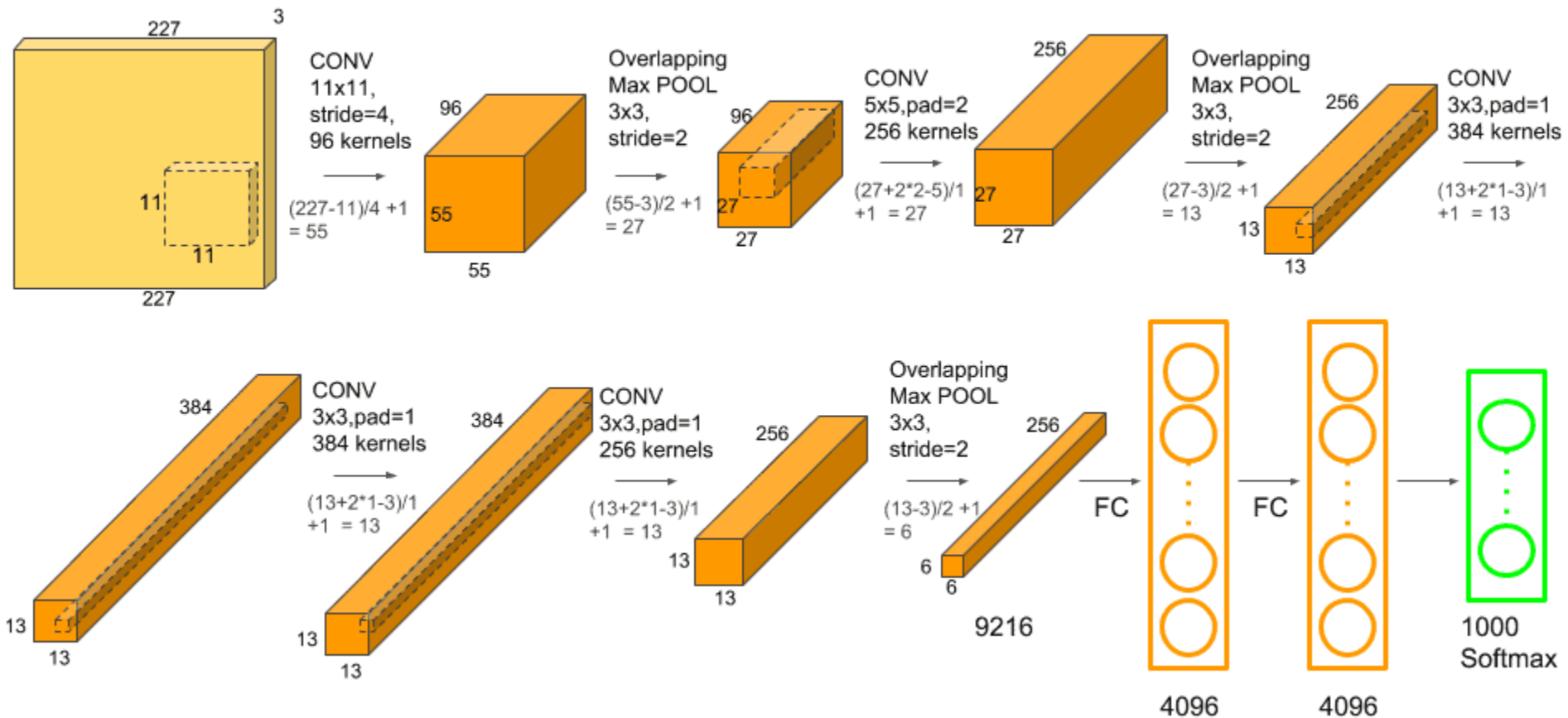
e.g. Abyssinian cat, Bulldog, French Terrier, Cormorant, Chickadee,
Red fox, banjo, barbell, hourglass, knot, maze, viaduct, etc.



cat, tabby cat (0.71)
Egyptian cat (0.22)
red fox (0.11)
.....

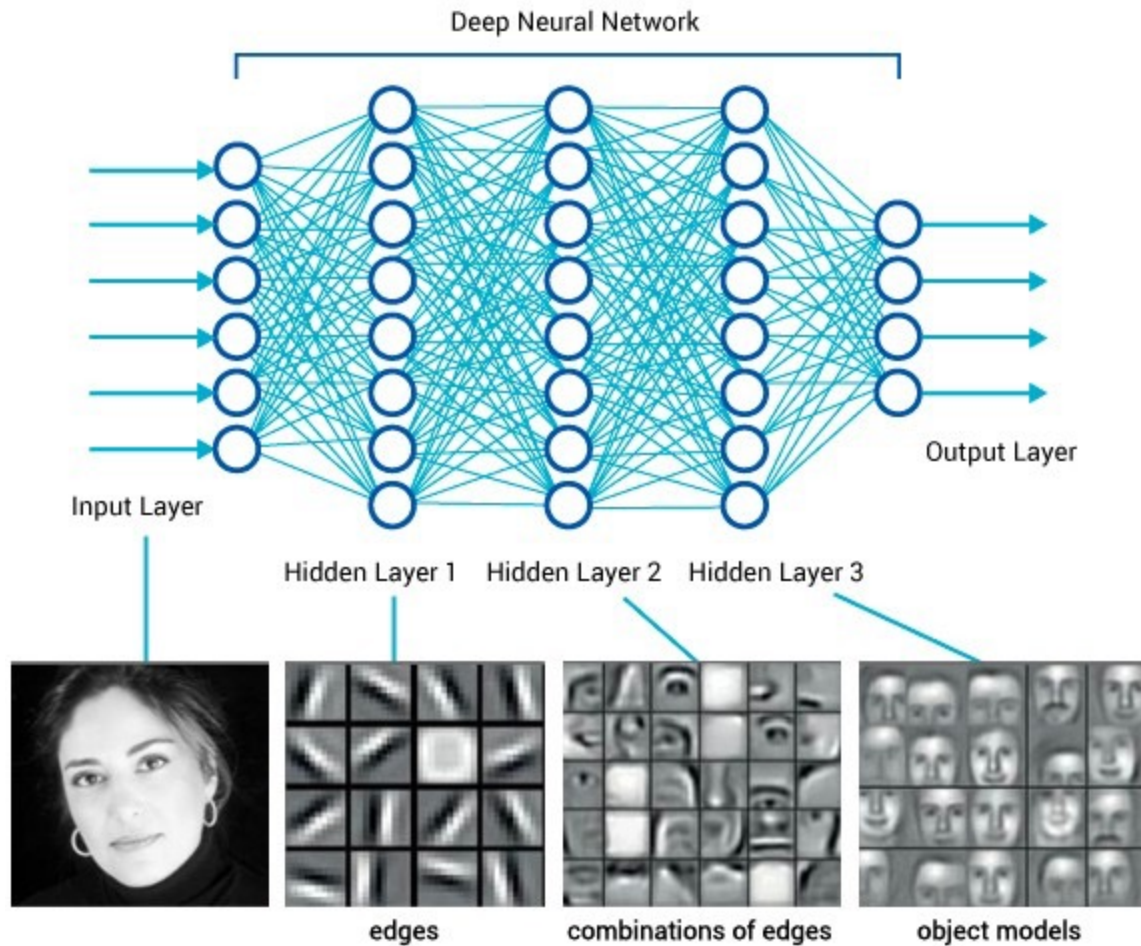
Train on ImageNet
challenge dataset,
~1.2 million images

Alexnet

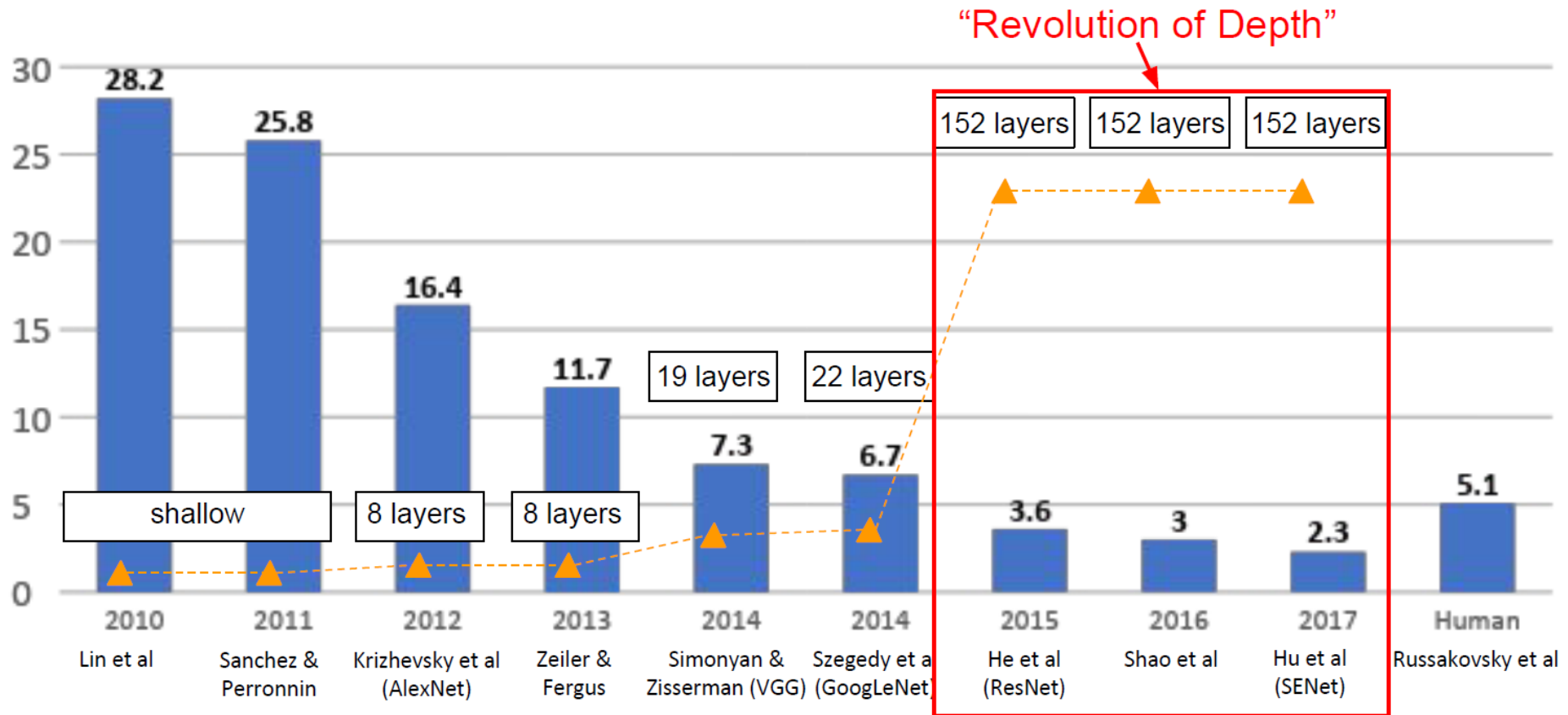


- Initially vectors of $227*227*3 = 154\,587$ features).
- Represented as a vector of 4096 features
- The two fully connected and softmax layers are similar to a multi layer perception and could actually be replaced by other kinds of classifiers such as Random Forests or SVMs. However they are really important for the training phase of the neural net.

What is happening?



Revolution of depth

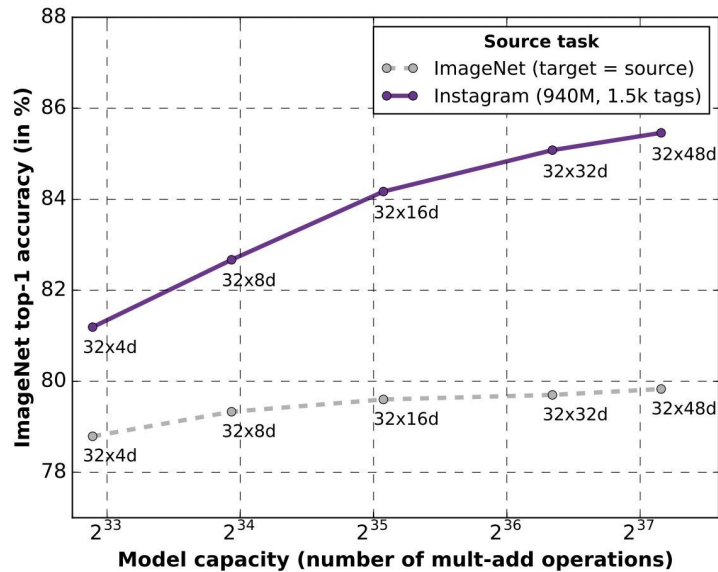


Inception

- The inception module is a building block for convolutional neural networks (CNNs) introduced by Google researchers.
- It applies several convolutional filters of different sizes simultaneously, allowing the network to capture information at various scales and complexities.
- The use of 1×1 convolutions serves as a method for dimensionality reduction, reducing computational complexity and the number of parameters without losing depth in the network.
- It has been successfully applied to various tasks such as image classification, object detection, face recognition, and image segmentation

ImageNet pretraining -> Instagram pretraining

Bigger models are saturated on ImageNet, but with more data bigger models do better



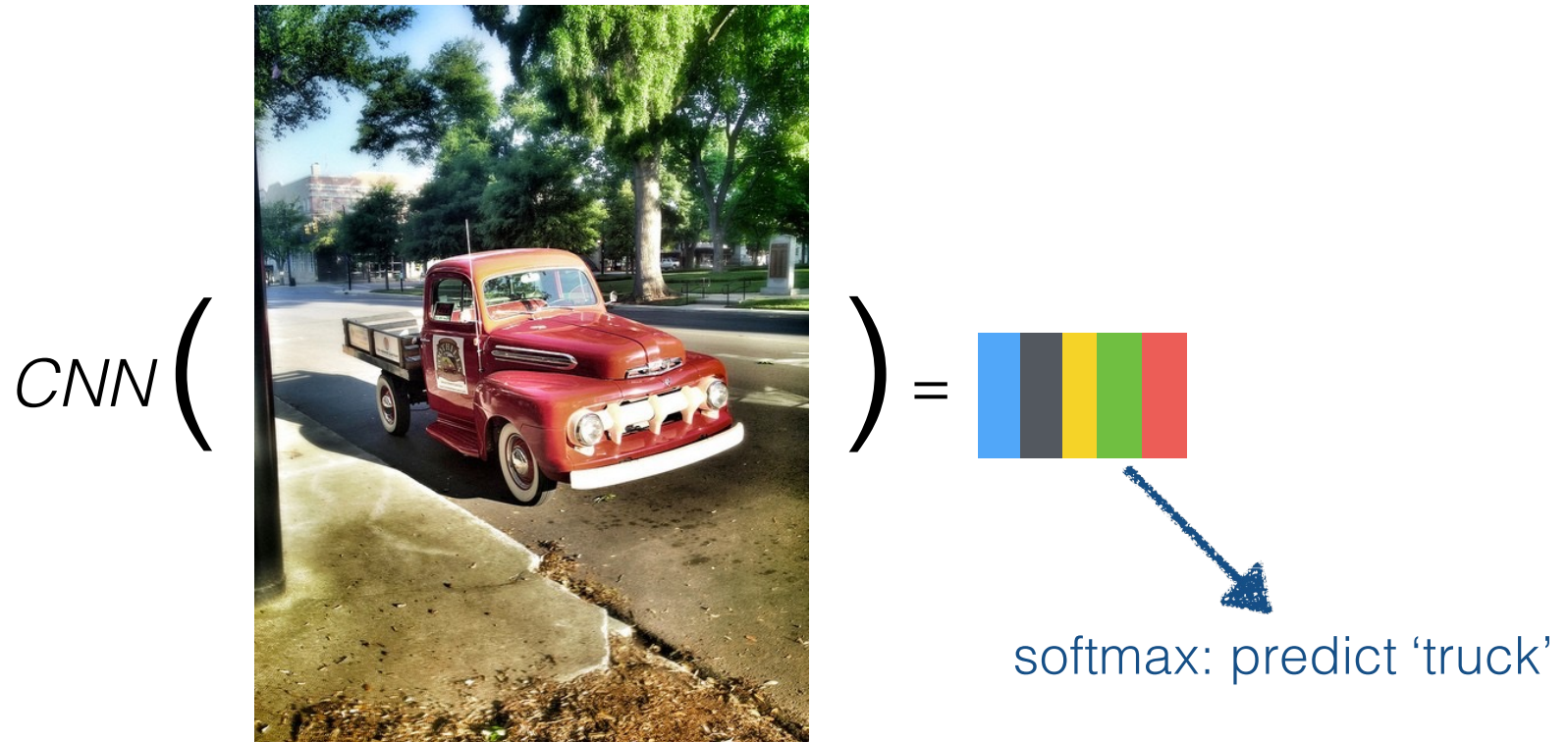
Biggest network was pretrained on 3.5B Instagram images

Trained on 336 GPUs for 22 days

Mahajan et al, "Exploring the Limits of Weakly Supervised Pretraining", arXiv 2018

At the end of the day, ...

... we generate a fixed size vector from an image and run a classifier over it



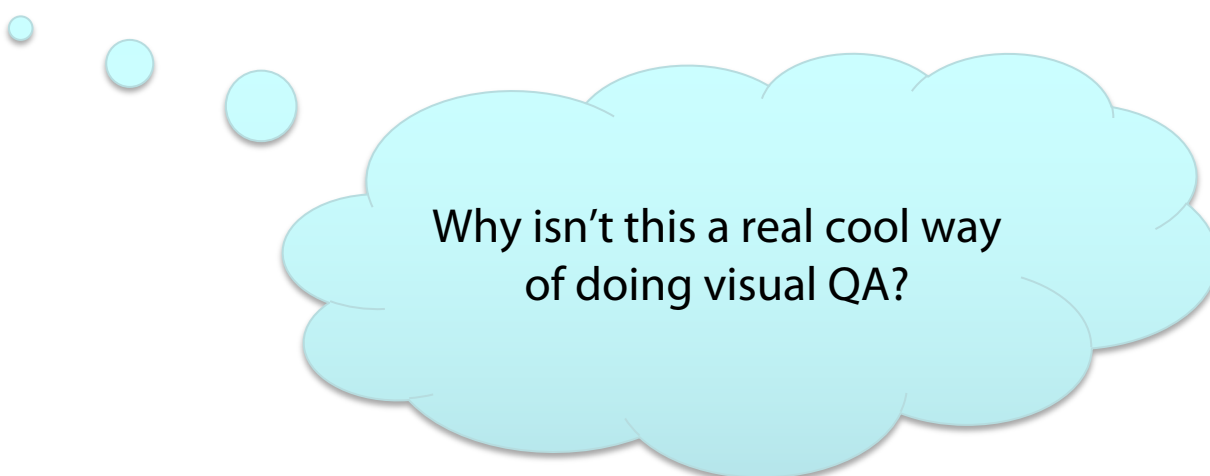
Key insight

This vector is useful for many more tasks than just image classification! We can use it for *transfer learning*



Simple visual QA

- $i := \text{CNN}(\text{image}) \rightarrow$ use an existing network trained for image classification and freeze weights
- $q := \text{BERT}(\text{question}) \rightarrow$ learn weights
- Answer = $\text{softmax}(\text{linear}([i;q]))$



Why isn't this a real cool way of doing visual QA?

Visual attention

Use the question representation q to determine where in the image to look

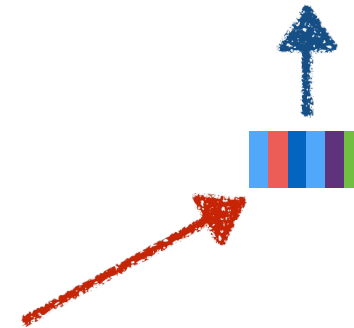
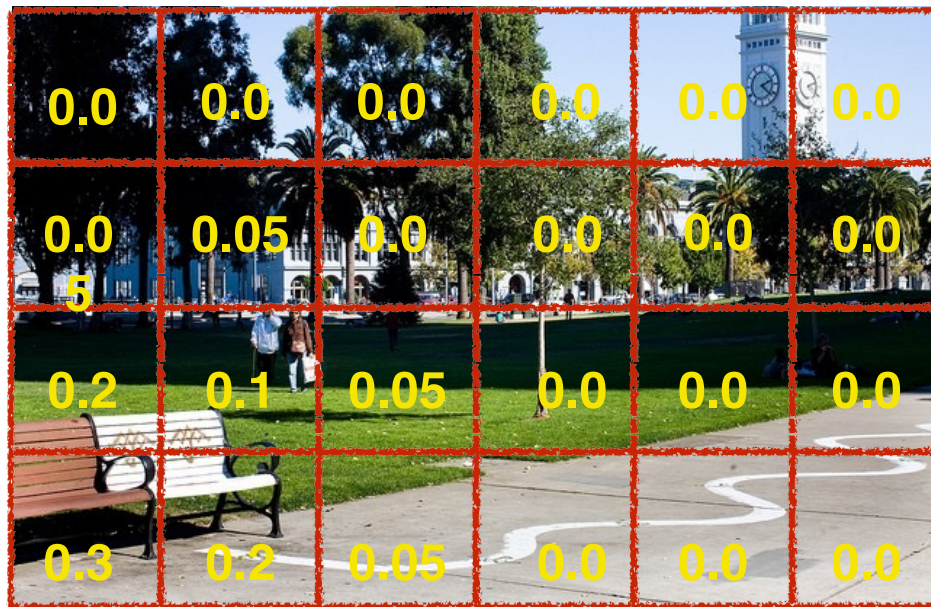


How many benches are shown?



Attention over final convolutional layer in network: 196 boxes, captures color and positional information

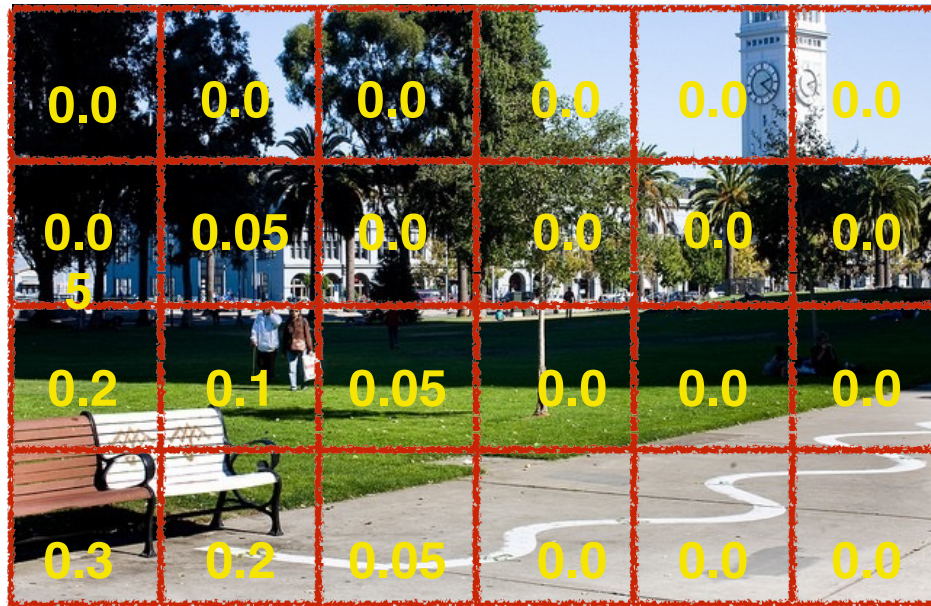
softmax:
predict answer



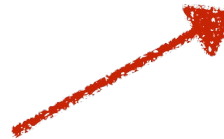
How many benches are shown?



Attention over final convolutional layer in network: 196 boxes, captures color and positional information



softmax:
predict answer



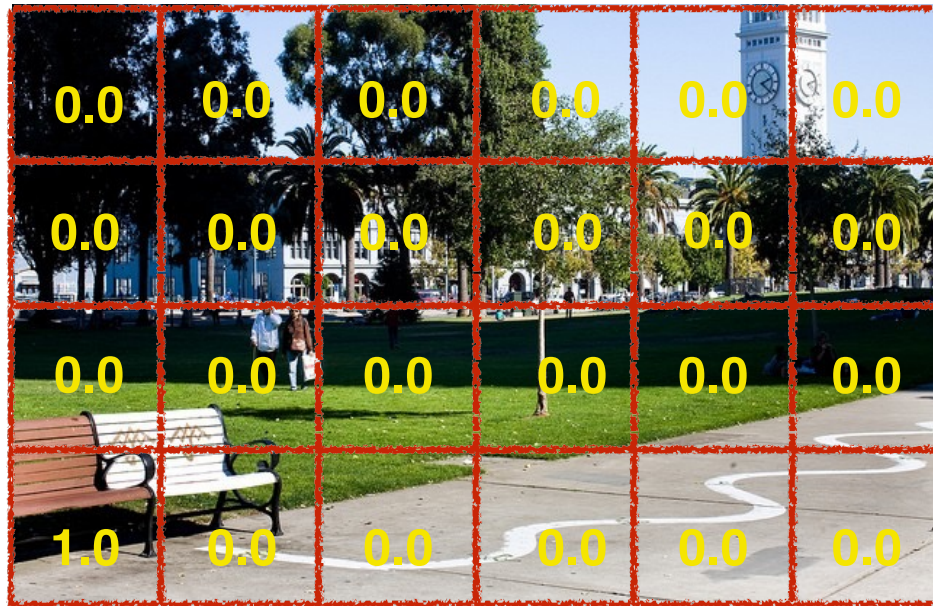
How can we compute these attention scores?

How many benches are shown?



Hard Attention

Attention over final convolutional layer in network: 196 boxes, captures color and positional information



softmax:
predict answer



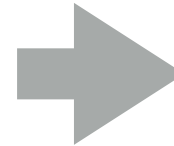
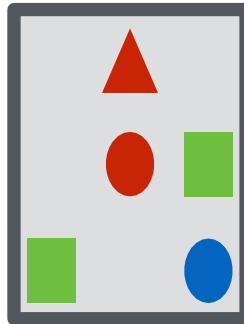
We can use *reinforcement learning* to focus on just one box

How many benches are shown?



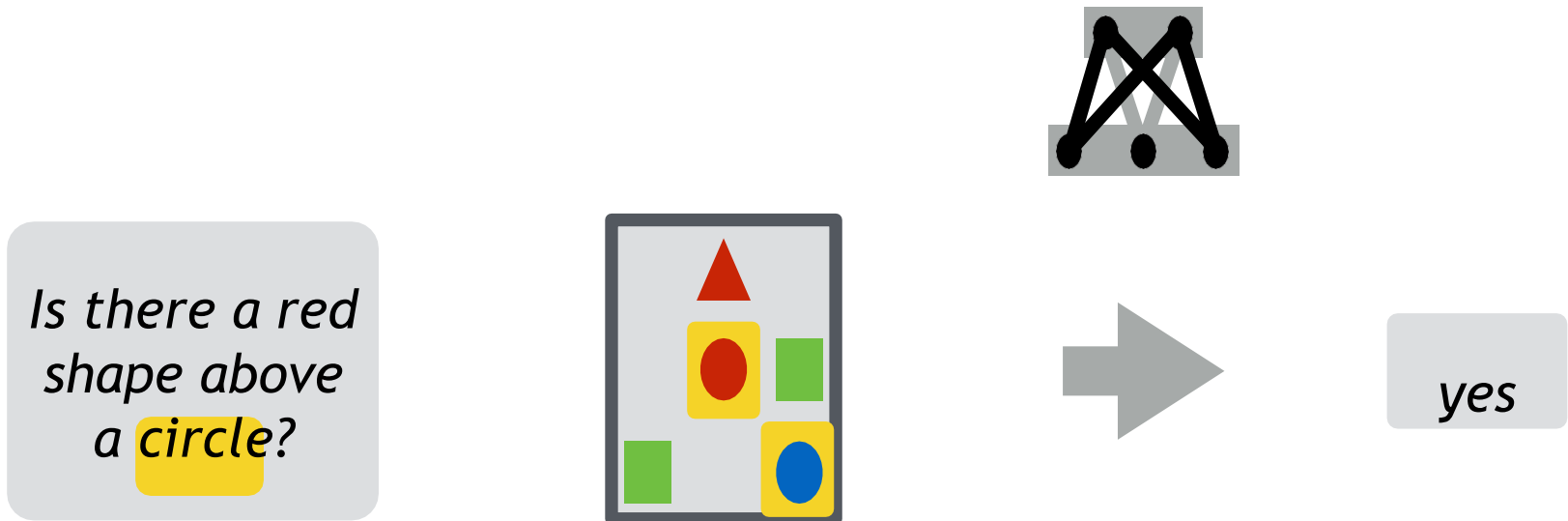
Grounded question answering

Is there a red shape above a circle?



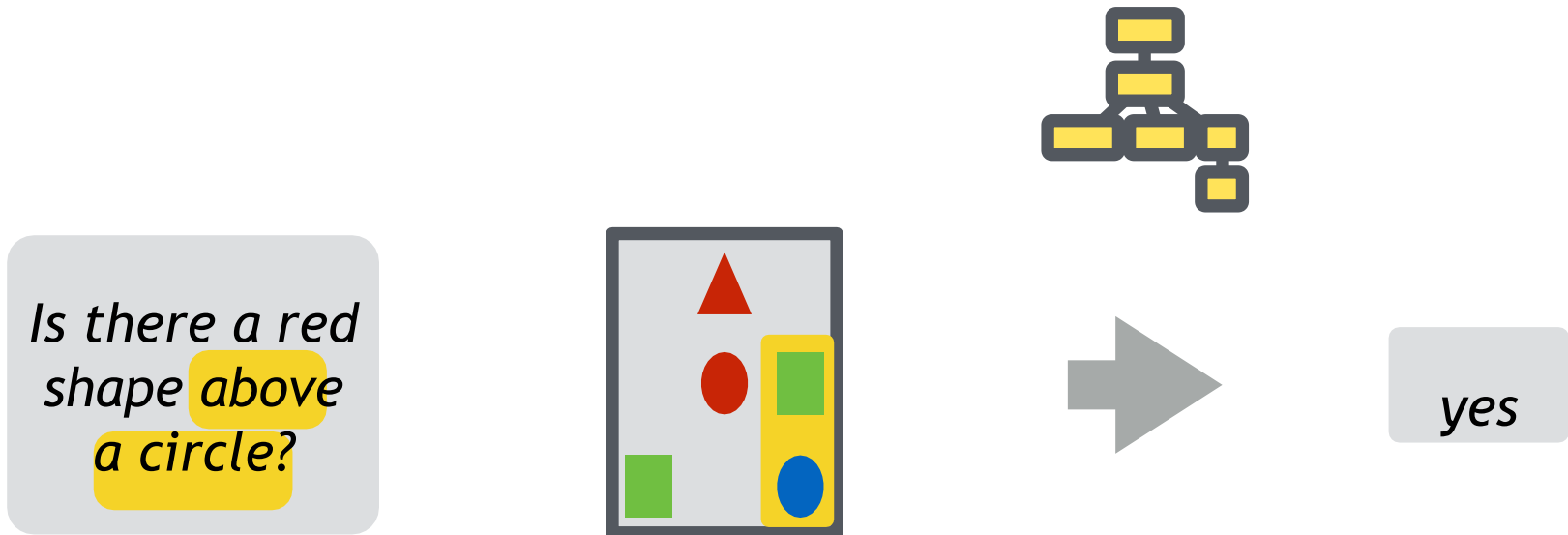
yes

Neural nets learn lexical groundings



[Iyyer et al. 2014, Bordes et al. 2014, Yang et al. 2015, Malinowski et al., 2015]

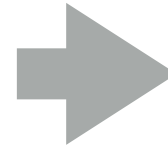
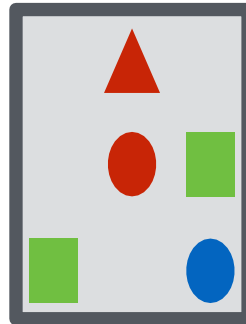
Semantic parsers learn composition



[Wong & Mooney 2007, Kwiatkowski et al. 2010, Liang et al. 2011, A et al. 2013]

Neural module networks learn both!

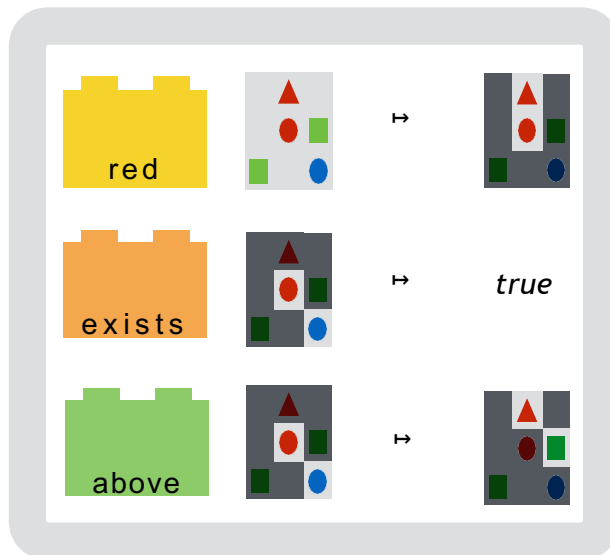
*Is there a red
shape above
a circle?*



yes

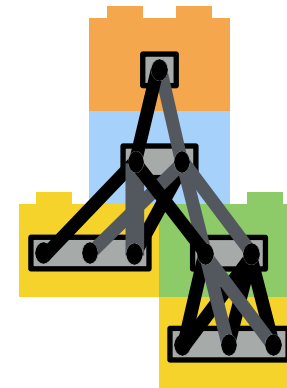
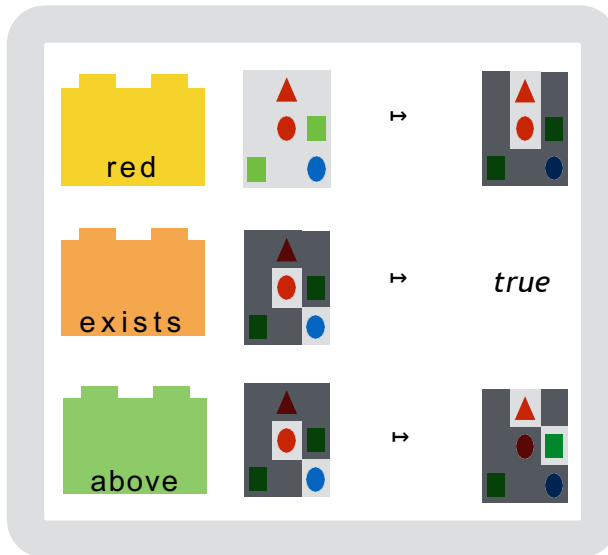
Neural module networks

Is there a red shape above a circle?



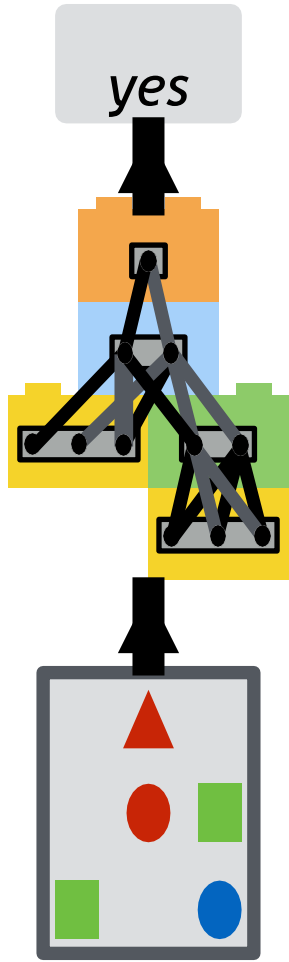
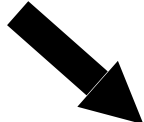
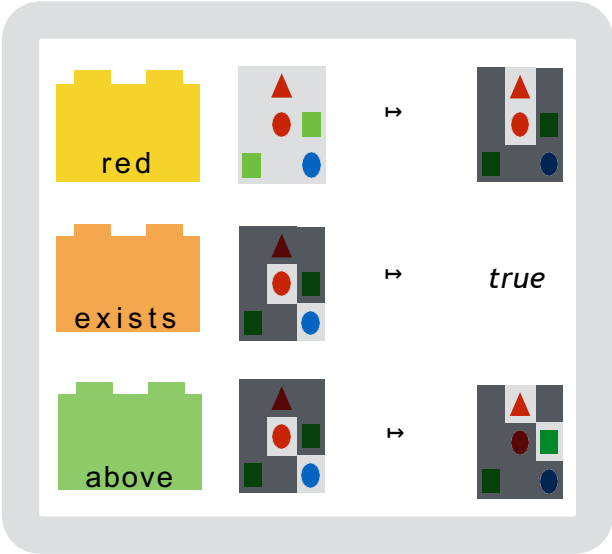
Neural module networks

Is there a red shape above a circle?

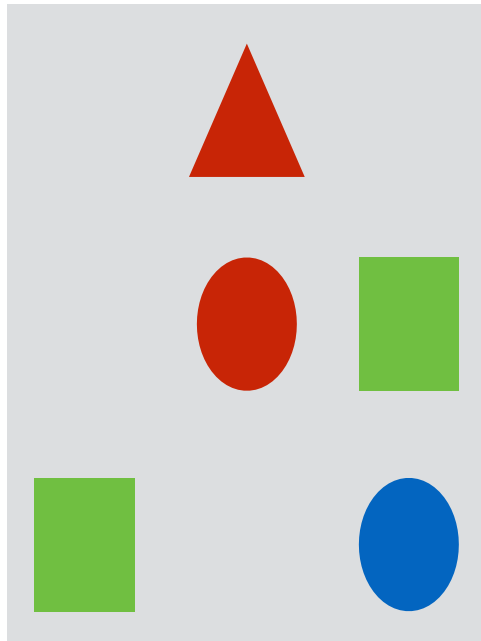


Neural module networks

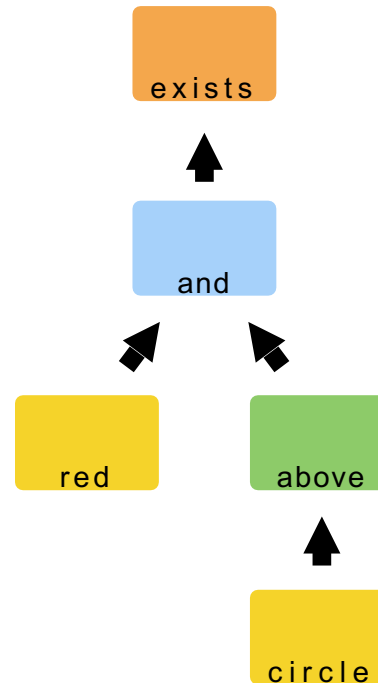
Is there a red shape above a circle?



Sentence meanings are computations



Is there a red shape above a circle?



NLVR²: natural language for visual reasoning! (Suhr et al., 2018)



TRUE OR FALSE: the left image contains twice the number of dogs as the right image, and at least two dogs in total are standing.

CerealBar: Situated, Collaborative Natural Language Understanding

CerealBar is a two-person collaborative game. We built CerealBar to study natural language understanding in collaborative interactions.

- Two players -- a **leader** and a **follower** -- take turns moving around the game board to collect sets of cards and earn points.
- In addition to moving, the **leader** uses their access to the full environment to plan which set of cards should be collected next, and writes instructions to the follower.
- The **follower** only has access to a first-person view, so their job is to follow the leader's instructions to the best of their ability. However, the follower can move farther than the leader in each turn.

We crowdsourced interactions between human players in the CerealBar game. We also designed and trained a **neural network agent** to play as the follower in CerealBar. Our approach makes contributions in modeling, learning, and evaluation. The CerealBar game, data, and modeling approach is described in Suhr et al. 2019 (EMNLP 2019).

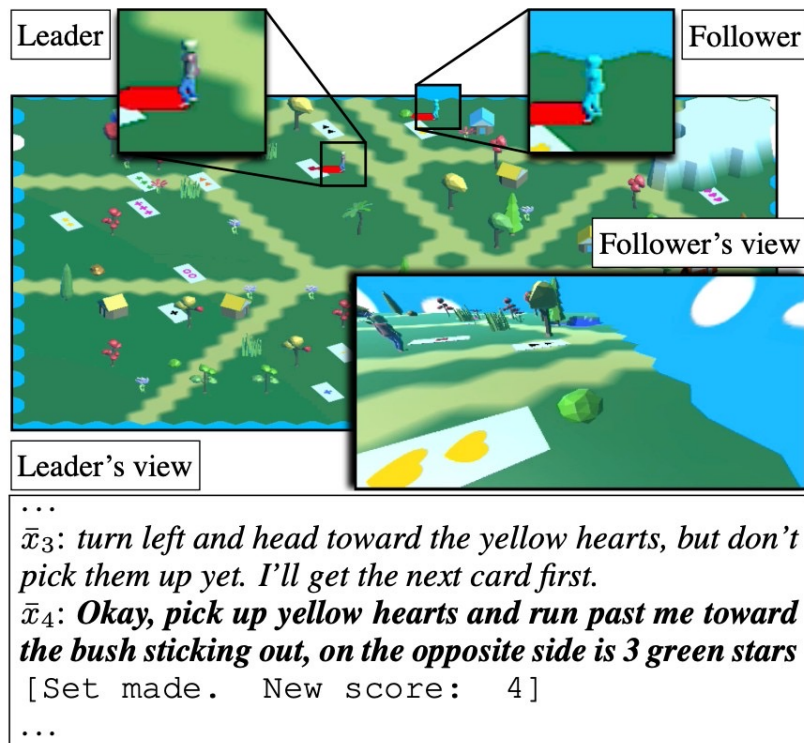


Figure 1: A snapshot from an interaction in CEREAL-BAR. The current instruction is in bold. The large image shows the entire environment. This overhead view is available only to the leader. The follower sees a first-person view only (bottom right). The zoom boxes (top) show the leader and follower.



\bar{x} : Okay, pick up yellow hearts and run past me toward the bush sticking out, on the opposite side is 3 green stars

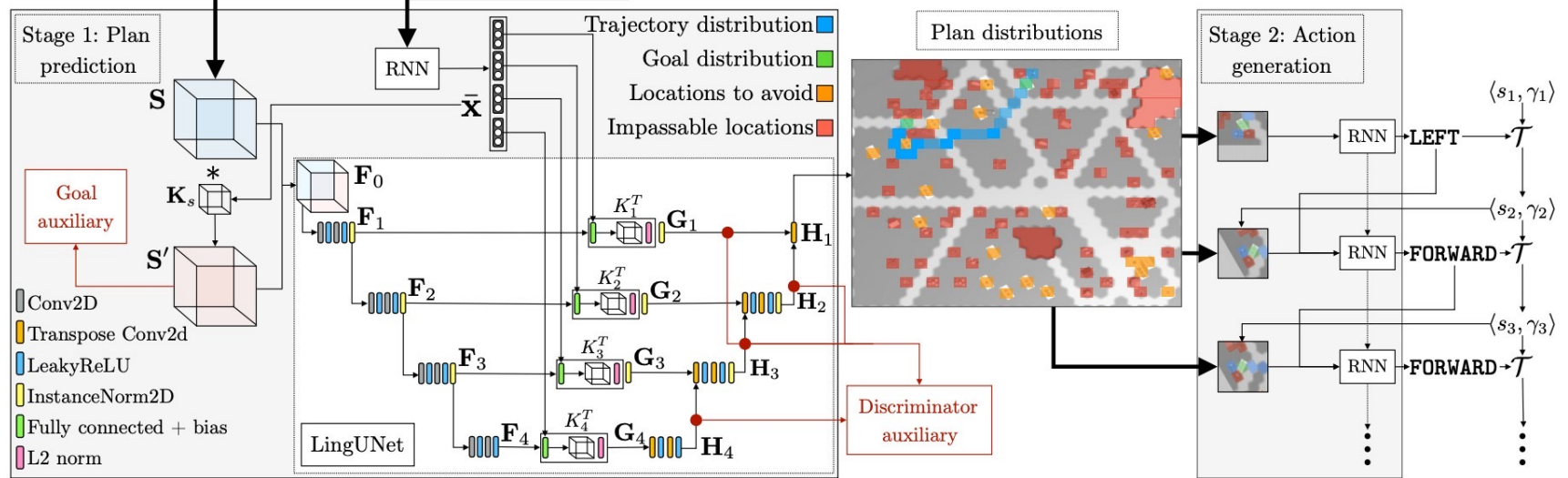


Image Captioning

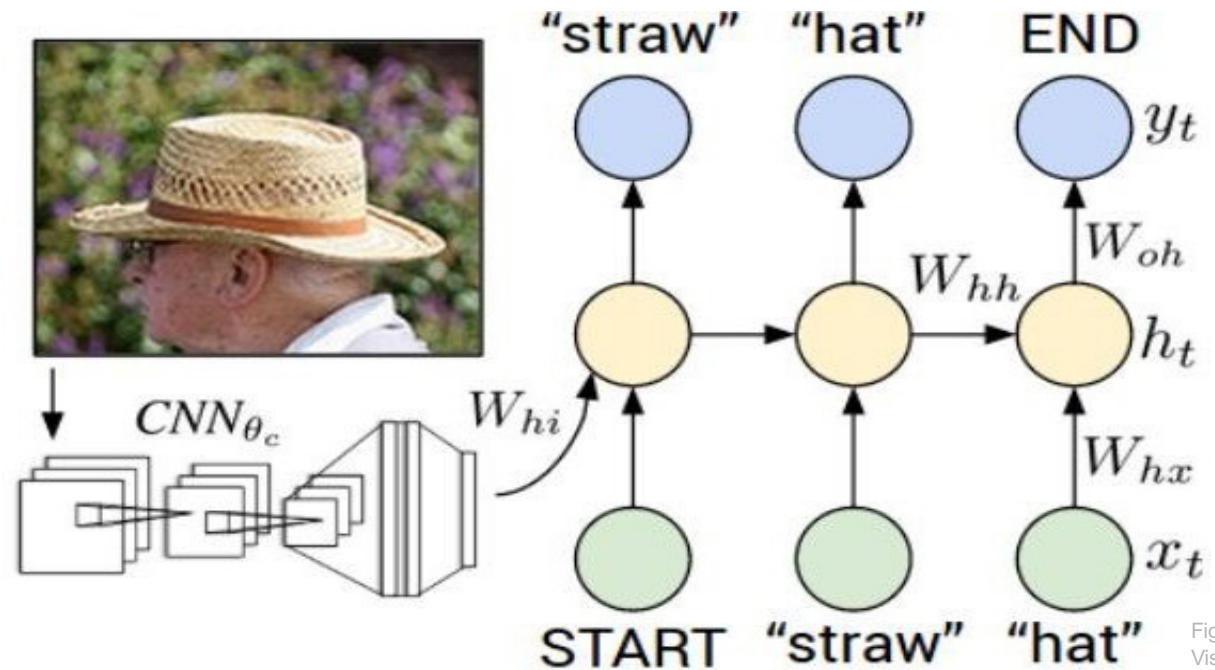


Figure from
Visual-Semantic
Image Description
copyright
Reproduced

Around 2014

- Explain Images with Multimodal Recurrent Neural Networks, Mao et al.
- Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
- Show and Tell: A Neural Image Caption Generator, Vinyals et al.
- Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
- Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

test image



This image is [CC0 public domain](#)



test image

This is our
ImageNet
CNN, now
used as a
feature
extractor

image

conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

conv-512

maxpool

FC-4096

FC-4096

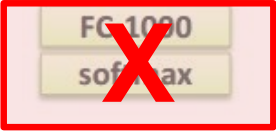
FC-1000

softmax



test image

This is our ImageNet CNN, now used as a feature extractor





test image



<START>

image

conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

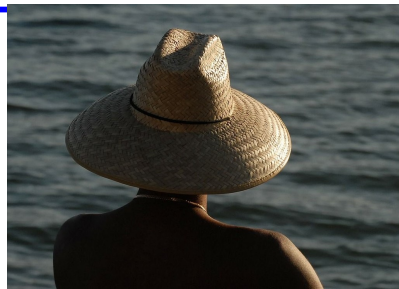
conv-512

maxpool

FC-4096

FC-4096

V



test image

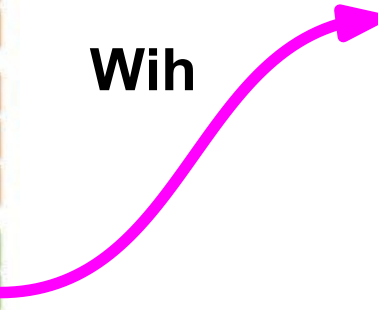
yO

hO

xO
<STA
RT>

<START>

Wih



before:

$$h = \tanh(W_{xh} * x + W_{hh} * h)$$

now:

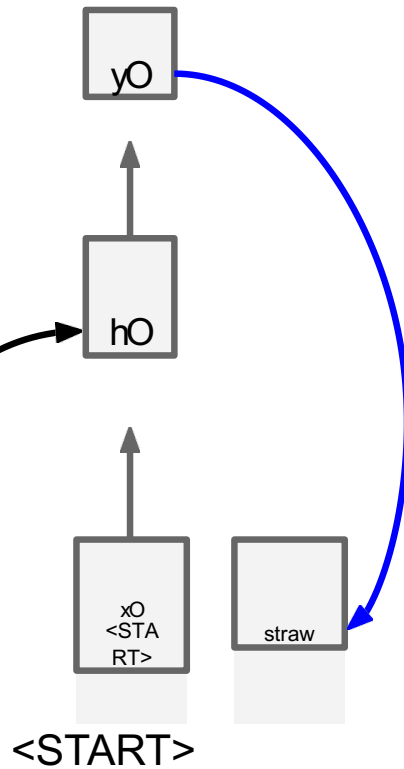
$$h = \tanh(W_{xh} * x + W_{hh} * h + W_{ih} * v)$$

let's use the image features to create a conditional LM



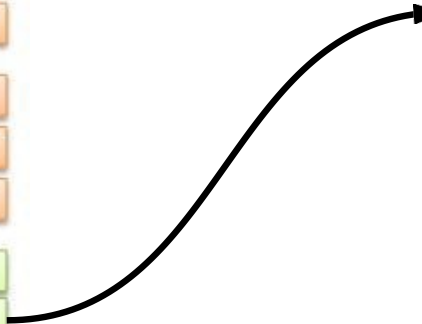
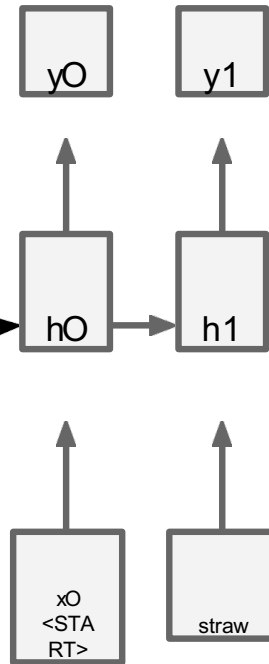


test image



sample!

test image



<START>

image

conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

conv-512

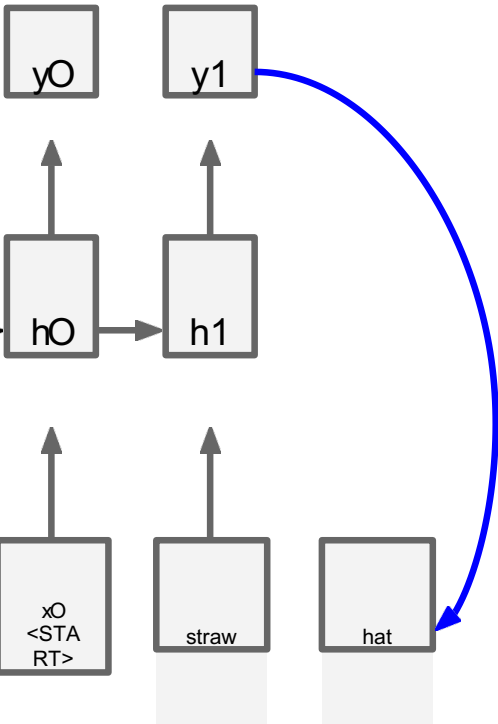
maxpool

FC-4096

FC-4096



test image



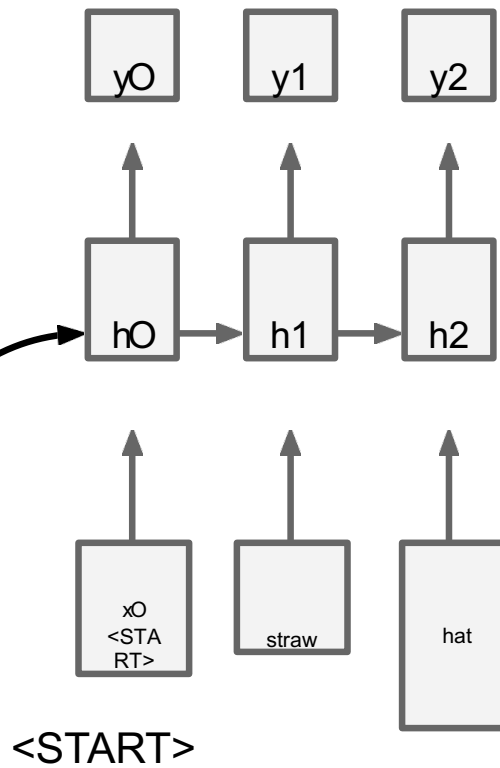
sample!

<START>





test image



image

conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

conv-512

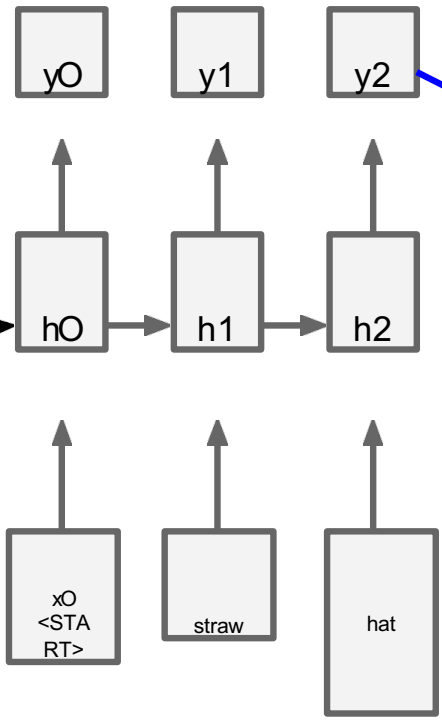
maxpool

FC-4096

FC-4096



test image



sample
<END> token
=> finish.

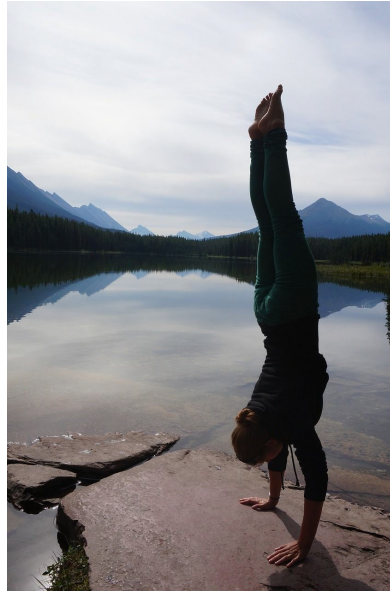
<START>

Image Captioning: Failure Cases

Captions generated using [neuralTalk2](#)
All images are [CC0 Public domain](#): [fur coat](#), [handstand](#), [spider web](#), [baseball](#)



A woman is holding a cat in her hand



A woman standing on a beach holding a surfboard



A bird is perched on a tree branch



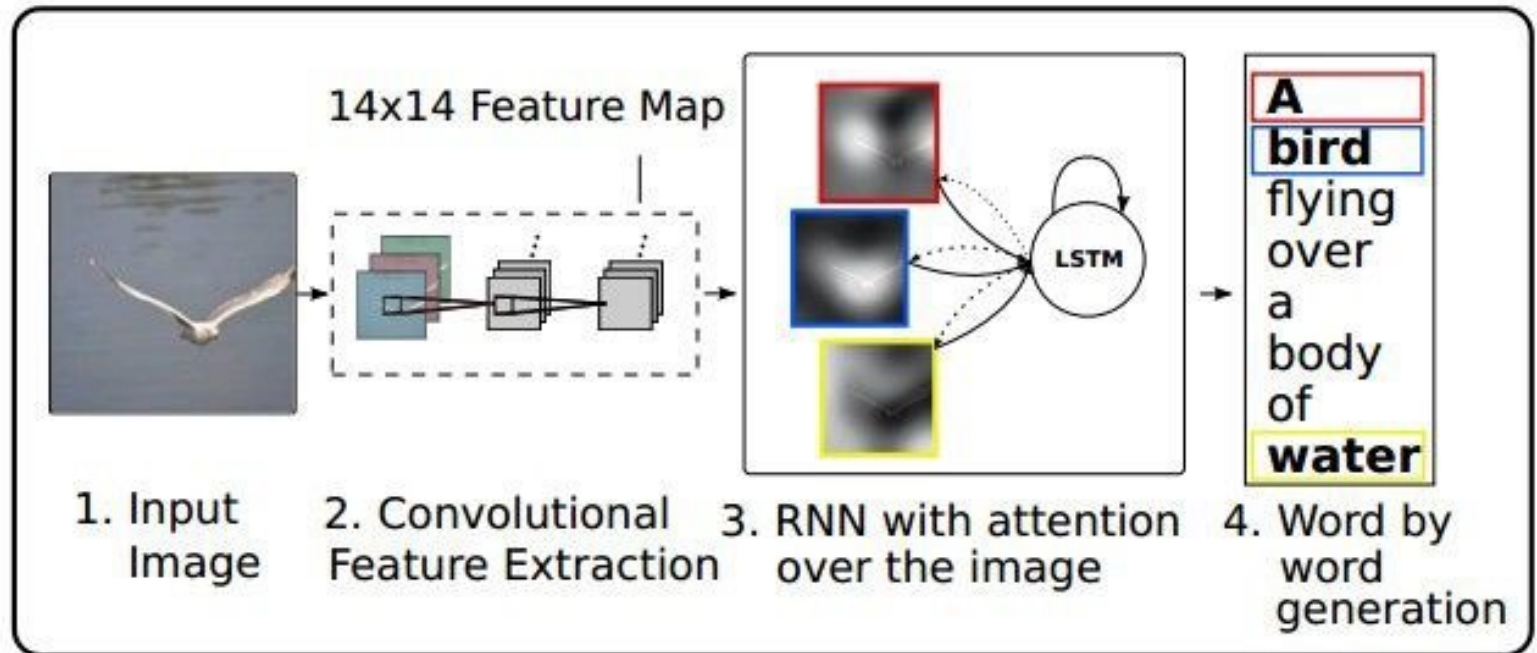
A person holding a computer mouse on a desk



A man in a baseball uniform throwing a ball

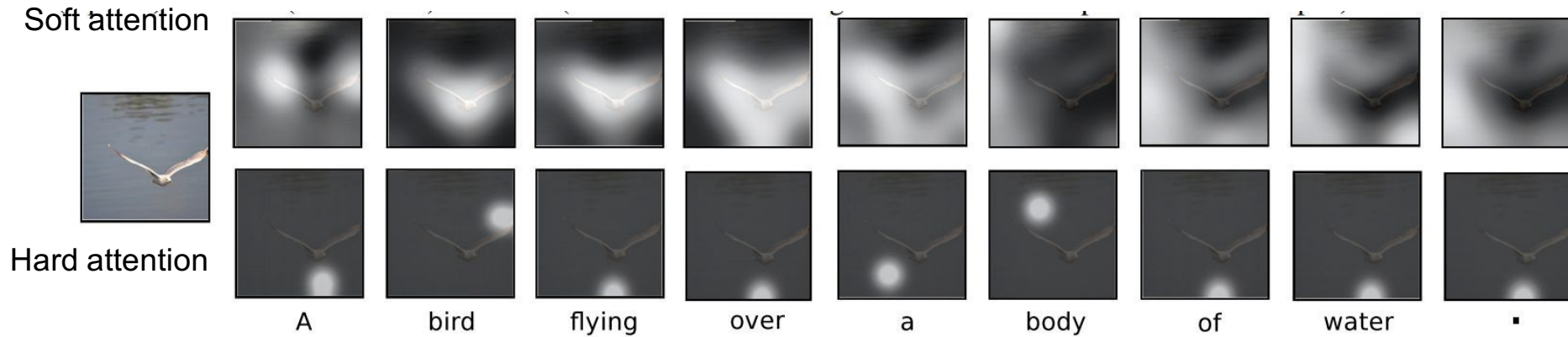
Image Captioning with Attention

RNN focuses its attention at a different spatial location when generating each word



Xu et al, "Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Image Captioning with Attention



Xu et al, "Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Image Captioning with Attention



A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

Xu et al, "Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Image Captioning using Transformers

Hybrid Solution

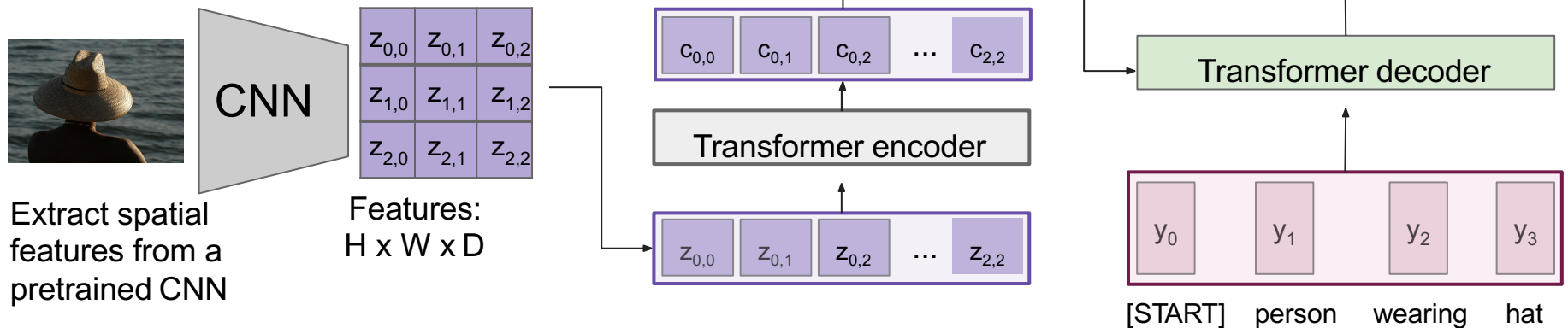


Image Captioning using transformers

- Perhaps we don't need convolutions at all?

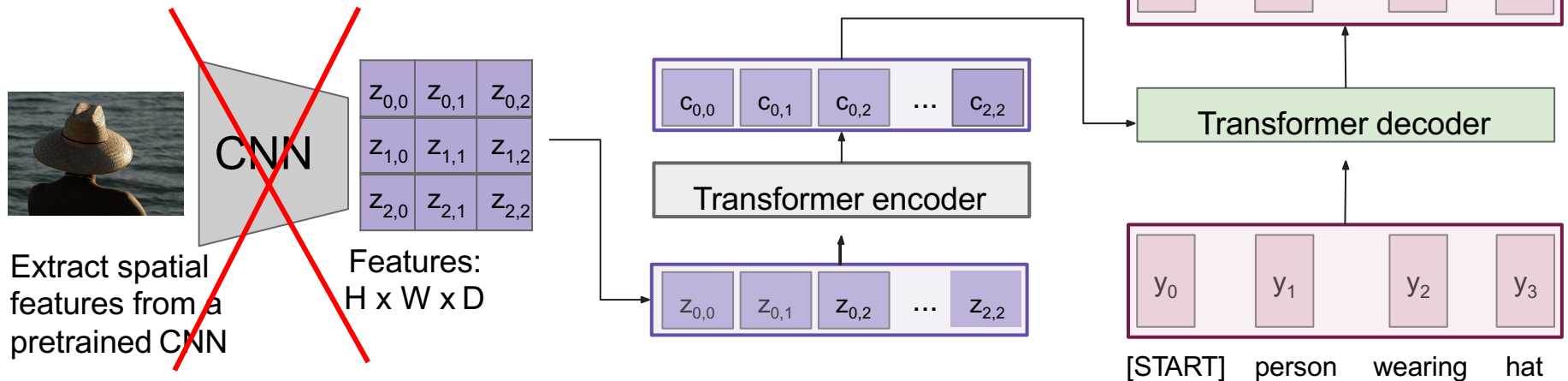
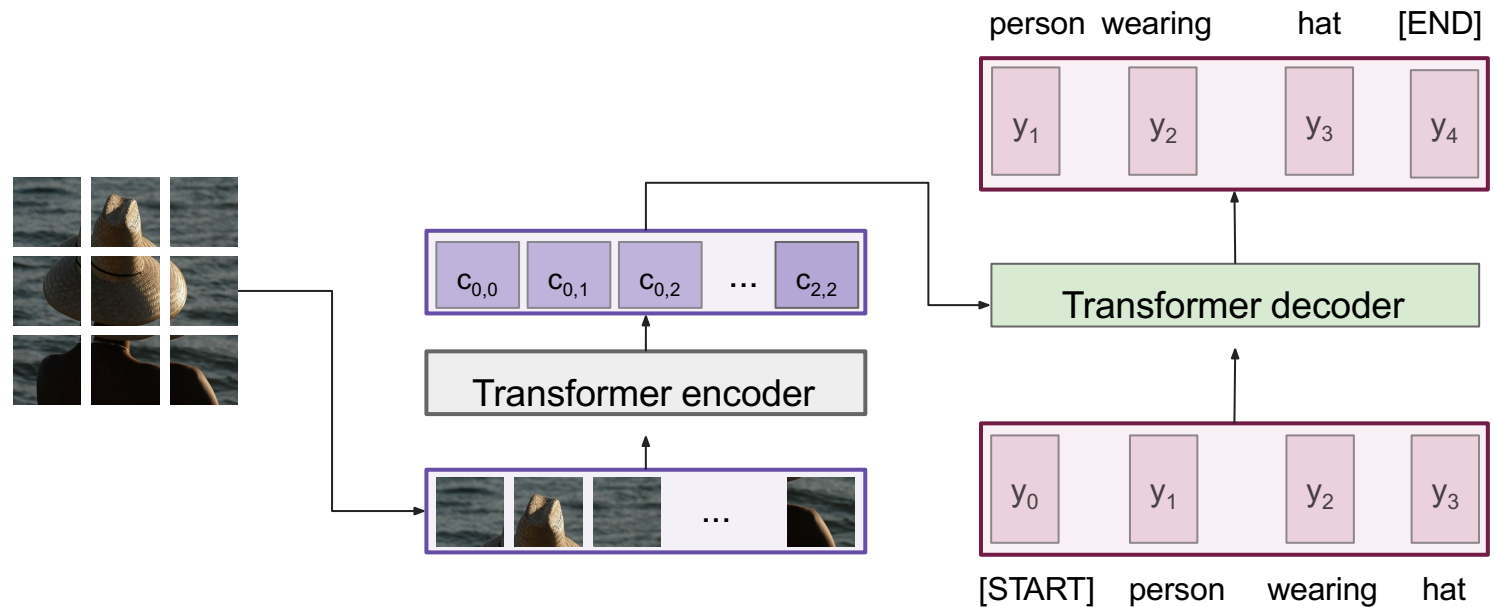


Image Captioning using **ONLY** transformers

- Transformers from pixels to language



Vision Transformers (ViT) vs. ResNets (BiT)

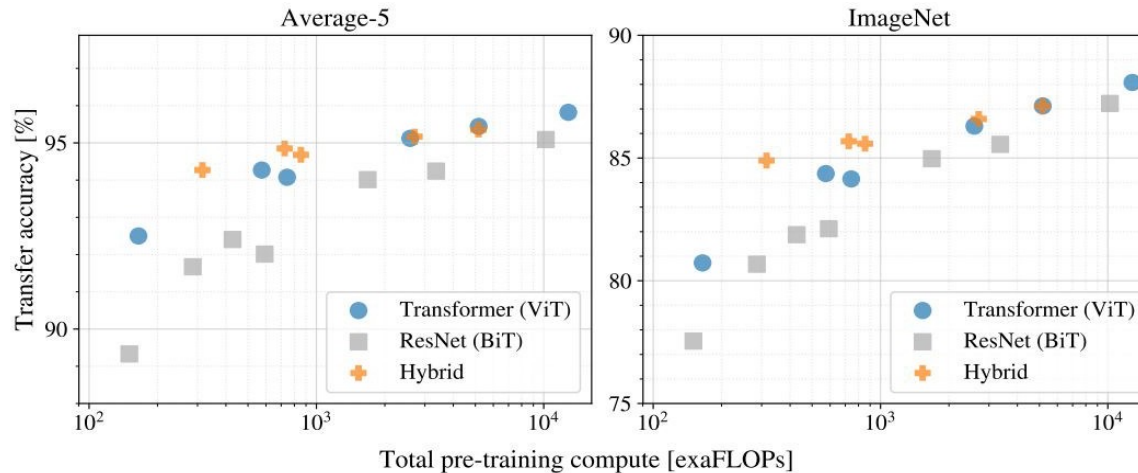


Figure 5: Performance versus cost for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

The BiT model was proposed in [Big Transfer \(BiT\): General Visual Representation Learning](#) by Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, Neil Houlsby. BiT is a simple recipe for scaling up pre-training of [ResNet](#)-like architectures (specifically, ResNetv2). The method results in significant improvements for transfer learning.

Intelligent Agents

Vision and Language

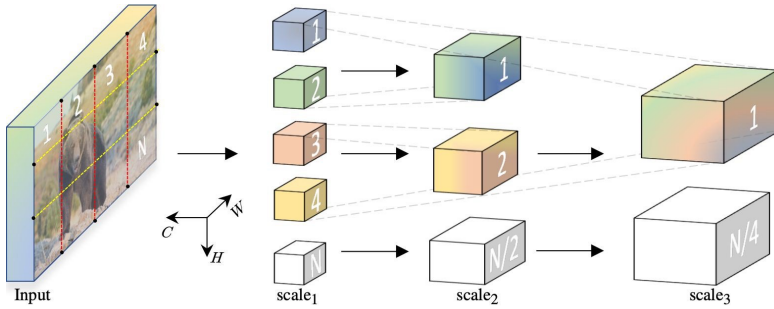
Prof. Dr. Ralf Möller

Universität zu Lübeck

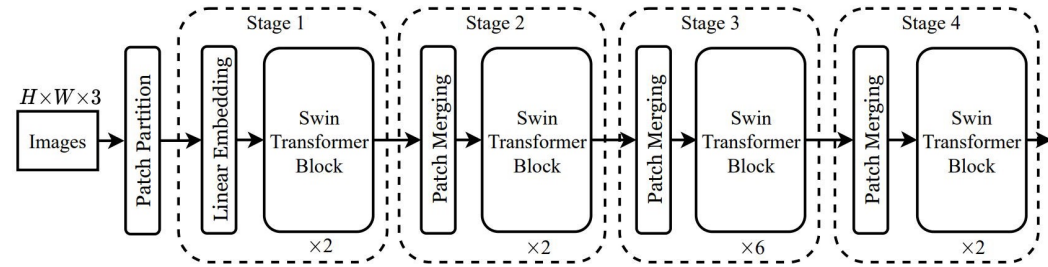
Institut für Informationssysteme



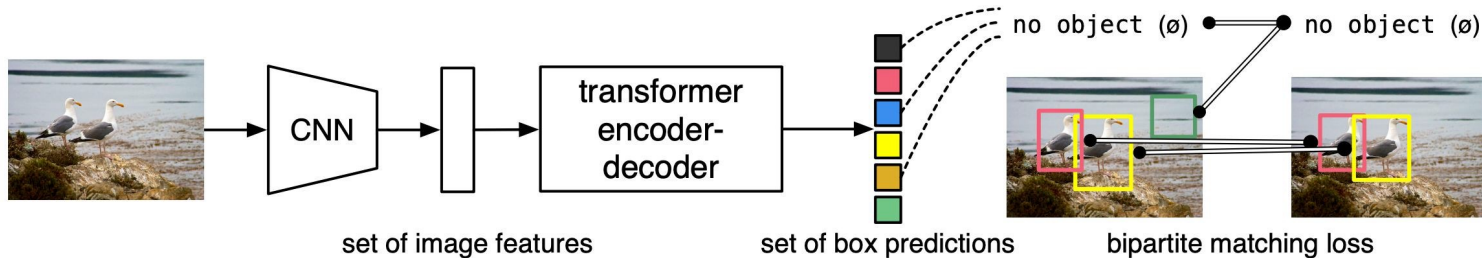
Vision Transformers



Fan et al, "Multiscale Vision Transformers", ICCV 2021



Liu et al, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", CVPR 2021



Carion et al, "End-to-End Object Detection with Transformers", ECCV 2020

ViLBERT (Vision and Language BERT)

ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, Article 2, 13–23. **2019**

Presented by - **Sidharth Singla**, 20774908



Model

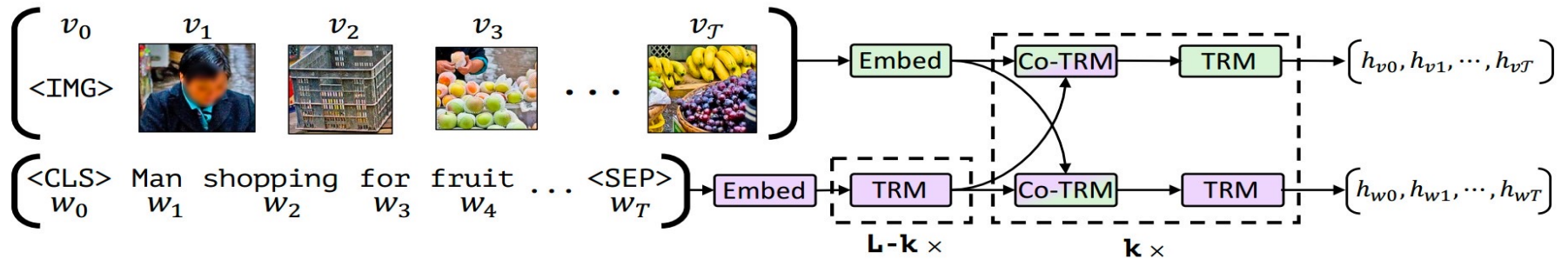
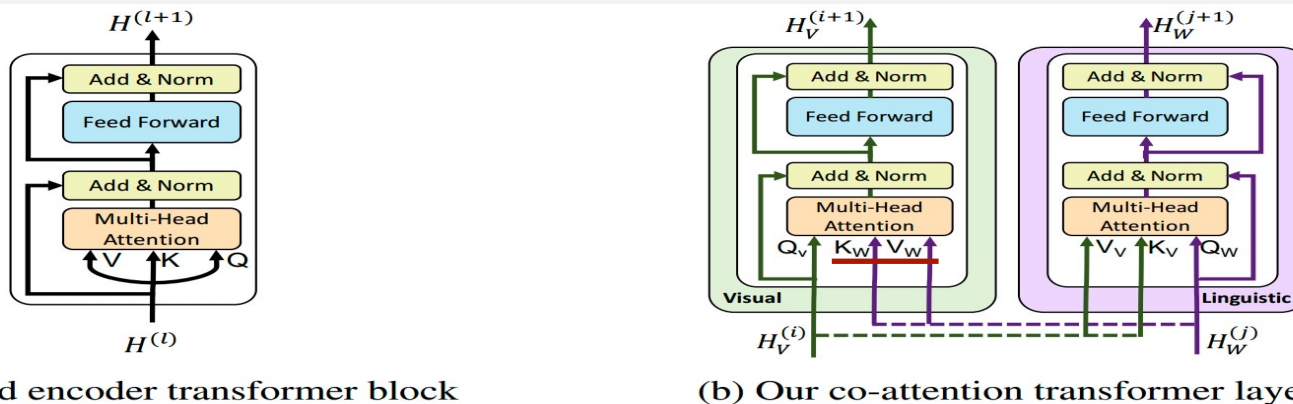


Figure 1: Our ViLBERT model consists of two parallel streams for visual (green) and linguistic (purple) processing that interact through **novel co-attentional transformer layers**. This structure allows for variable depths for each modality and enables sparse interaction through co-attention. Dashed boxes with multiplier subscripts denote repeated blocks of layers.

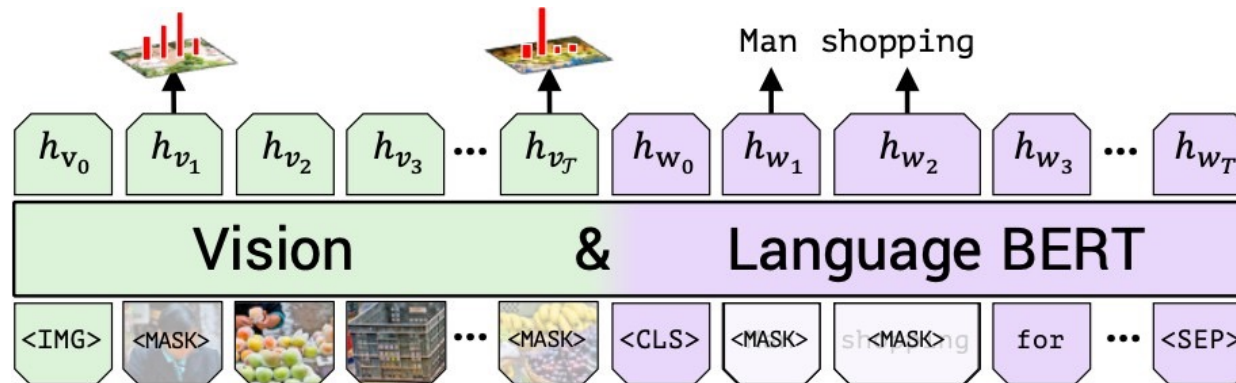


(a) Standard encoder transformer block

(b) Our co-attention transformer layer

Figure 2: We introduce a **novel co-attention mechanism** based on the transformer architecture. By **exchanging key-value pairs in multi-headed attention**, this structure enables vision-attended language features to be incorporated into visual representations (and vice versa).

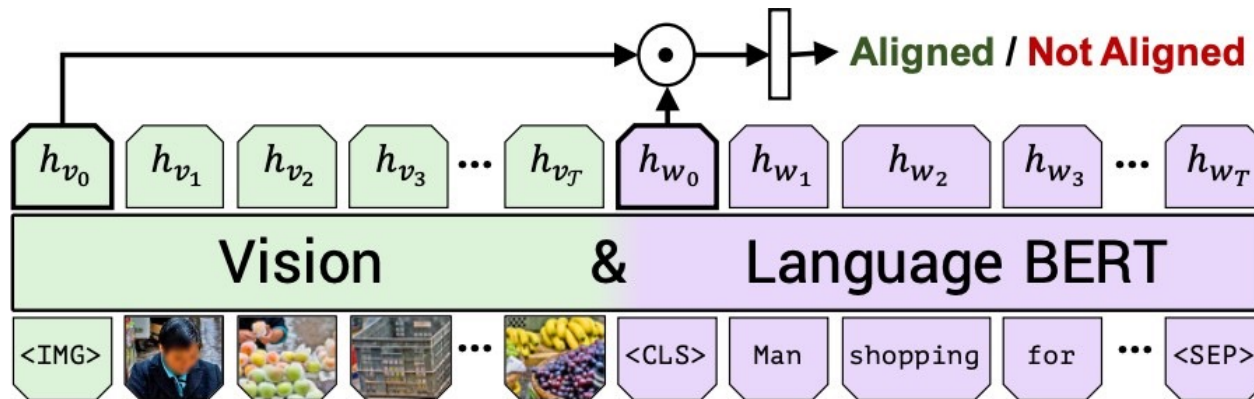
Pretraining: Masked Multi-Modal Learning Task



(a) Masked multi-modal learning

- Approximately 15% of both words and image region are masked and reconstructed given the remaining inputs
- Image features zeroed out 90% and unaltered 10%. Masked text inputs are handled as in BERT
- Model predicts a distribution over semantic classes rather than directly regressing the masked feature values for the corresponding image region
- Supervision by output distribution for the region from the pretrained detection model used. Minimize KL divergence

Pretraining: Multi-modal alignment task



(b) Multi-modal alignment prediction

- Prediction whether the text describes the image(image aligned with the text).
- Element-wise product between h_{IMG} and h_{CLS} and a linear layer is learnt to make the binary prediction
- Trained on Conceptual Captions Dataset
 - Collection of 3.3 million image-caption pairs automatically scraped from alt-text enabled web images

Transfer tasks

- **Pretrained** ViLBERT model is transferred to a set of four established **vision-and-language tasks**
- **Fine-tuning** strategy to modify the pretrained base model and perform the **new task** by training the entire model end-to-end

Visual Question Answering (VQA)

Training and Evaluation on VQA 2.0 dataset

- Fine-tuning:
Two layer MLP is learnt on top of the elementwise product of the image and text representations hIMG and hCLS.
- Multi-label classification task:
Binary cross-entropy loss.
Batch size 256. Maximum 20 epochs.
Initial learning rate 4e-5.




VQA

In [information theory](#), the **cross-entropy** between two [probability distributions](#) p and q over the same underlying set of events measures the average number of [bits](#) needed to identify an event drawn from the set if a coding scheme used for the set is optimized for an estimated probability distribution q , rather than the true distribution p .

Visual Commonsense Reasoning (VCR)

- Given an image, Visual Question Answering (Q->A) and Answer justification (QA->R).
- Trained on Visual Commonsense Reasoning (VCR) dataset having object tags integrated into the language
- Fine-tuning: Question and each possible response is concatenated and four different text inputs are passed along with the image. A linear layer is learnt on top of the post-element-wise product representation.
- Softmax prediction.
Cross-entropy loss.
20 epochs. Batch size 64.
Initial learning rate 2e-5.



Why is [person4] pointing at [person1]?

a) He is telling [person3] that [person1] ordered the pancakes.
b) He just told a joke.
c) He is feeling accusatory towards [person1].
d) He is giving [person1] directions.

VCR Q→A

Rationale: a) is correct because...

a) [person1] has the pancakes in front of him.
b) [person4] is taking everyone's order and asked for clarification.
c) [person3] is looking at the pancakes both she and [person2] are smiling slightly.
d) [person3] is delivering food to the table, and she might not know whose order is whose.

VCR QA→R

Grounding Referring Expressions

- Localize an image region given a natural language reference.
- Training and Evaluation is done on RefCOCO+ dataset.
- Bounding box proposals provided by *MAttNet* <https://arxiv.org/abs/1801.08186>, which uses a Mask R-CNN, are directly used.
- Fine-tuning: Final representation h_{vi} is passed into a learned linear layer to predict a matching score. IoU is computed with the ground truth box thresholding at 0.5.
- Binary cross-entropy loss.
Maximum 20 epochs. Batch size 256.
Initial learning rate $4e-5$.



Referring Expressions

Mask R-CNN is a **Convolutional Neural Network (CNN)** and **state-of-the-art in terms of image segmentation and instance segmentation**. Mask R-CNN was developed on top of Faster R-CNN, a Region-Based Convolutional Neural Network.

Intersection over Union (IoU), also known as the Jaccard index (Metric and A Loss for Bounding Box Regression)

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Caption-Based Image Retrieval

- **Caption-Based Image Retrieval**

- Identifying an image from a pool given a caption describing its content
- Training and Evaluation is done on the Flickr30k dataset

- **'Zero-shot' Caption-Based Image Retrieval**

- Pre-trained multi-modal alignment prediction model on Conceptual Captions dataset is used directly. No fine-tuning
- Demonstrates that the pretraining has developed the ability to ground text. Tested on the caption-based image retrieval task test-set

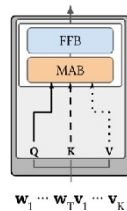


Nowadays: Many different V&L BERTs

Single- & Dual-Stream Architectures

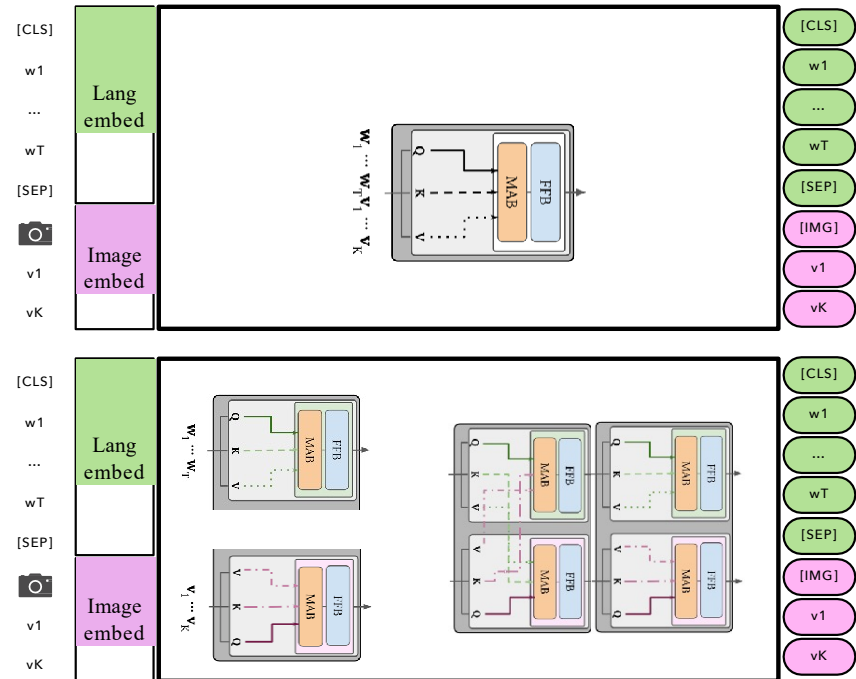
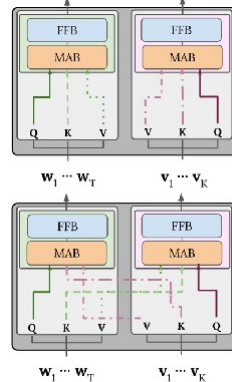
Single-Stream

- Concat image-text inputs



Dual-Stream

1. Image and text independently
2. Cross-modal layers



General approach

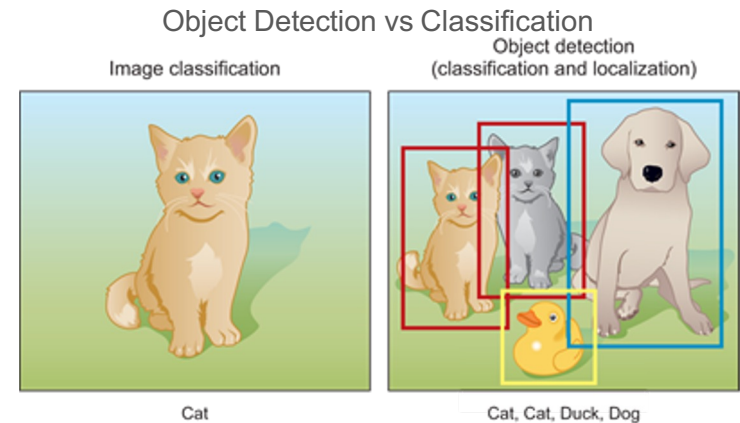
AI becomes successful:

Not just knowledge representation languages,
but systems that can be used out of the box and
that can be fine-tuned

- Unsupervised pretraining
 - + Zero-shot application / generalization
 - + Few-shot application / examples
 - Effective for very large vision&language models
- Fine-tuning for specific tasks
 - Reinforcement

Object Detection

- ❖ Object detection is the problem of both locating **AND** classifying objects
- ❖ Goal of YOLO algorithm is to do object detection both fast **AND** with high accuracy



Segmentation vs. Detection

Semantic Segmentation



GRASS, CAT,
TREE, SKY

No objects, just pixels

Classification + Localization



CAT

Single Object

Object Detection



DOG, DOG, CAT

Multiple Object

Instance Segmentation

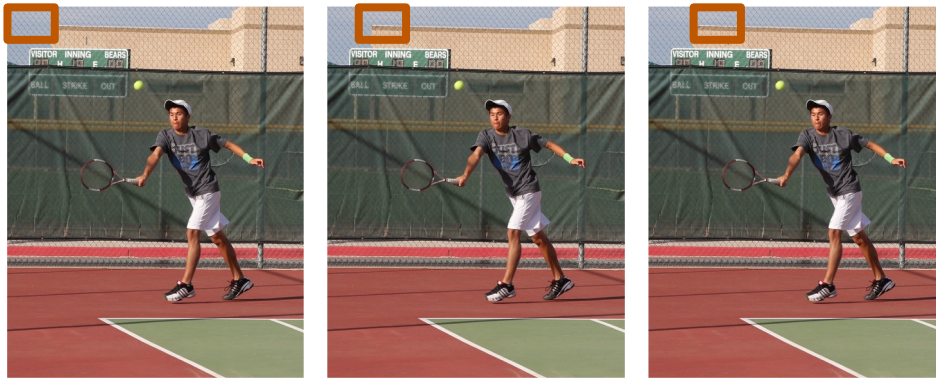


DOG, DOG, CAT

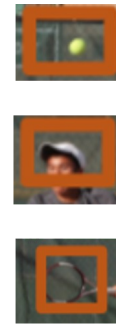
This image is CC0 public domain

Region-CNN (fast and faster, 2014 ongoing)

This pipeline was used in nearly all SOTA Object Detection prior:



Step 1: Scan the image to generate candidate bounding boxes



Label + confidence

hat - 0.92
racket - 0.2
ball - 0.23

Step 2: Run the bounding box through a classifier

Step 3: Conduct post-processing (filtering out redundant bounding boxes)

Diagram developed by Shivang Singh

Key Insights

- **Previous Approaches**

- ❖ A separate model for generating bounding boxes and for classification (more complicated model pipeline)
- ❖ Need to run classification many times (expensive computation)
- ❖ Looks at limited part of the image (lacks contextual information for detection)

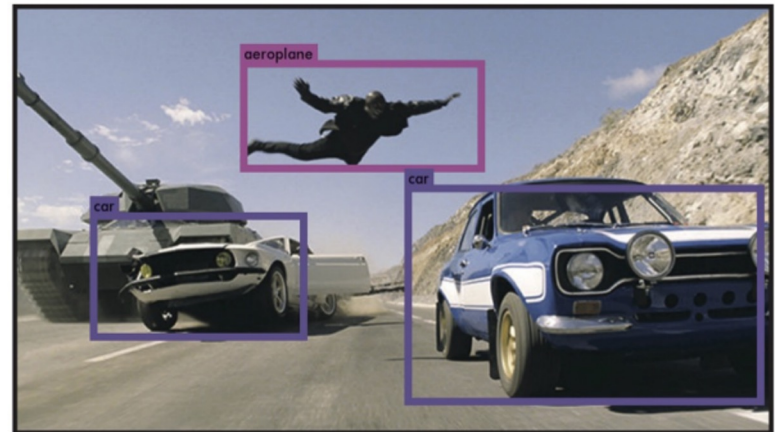
- **YOLO algorithm**

- ❖ A single embedding approach for localization and for classification (less complicated pipeline)
- ❖ Need to inference only once (efficient computation)
- ❖ Looks at the entire image each time leading to less false positives (has contextual information for detection)

Formal Problem Setting

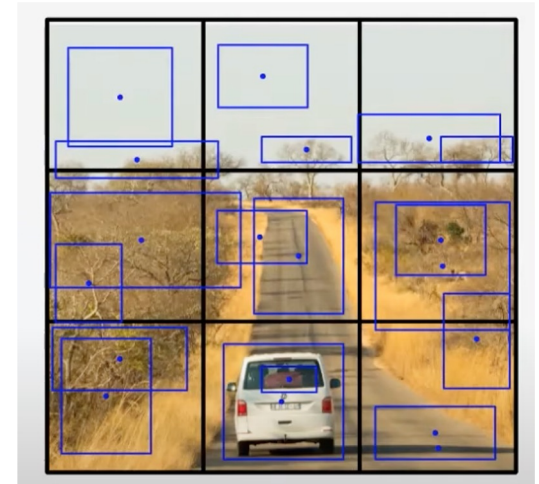
- ❖ Given an image generate bounding boxes, one for each detectable object in image
- ❖ For each bounding box, output 5 predictions: x , y , w , h , confidence. Also output class

- ❖ x , y (coordinates for center of bounding box)
- ❖ w, h (width and height)
- ❖ confidence (probability bounding box has object)
- ❖ class (classification of object in bounding box)



YOLO overview

- ❖ First, image is split into a $S \times S$ grid
- ❖ For each grid square, generate B bounding boxes
- ❖ For each bounding box, there are 5 predictions: x , y , w , h , confidence



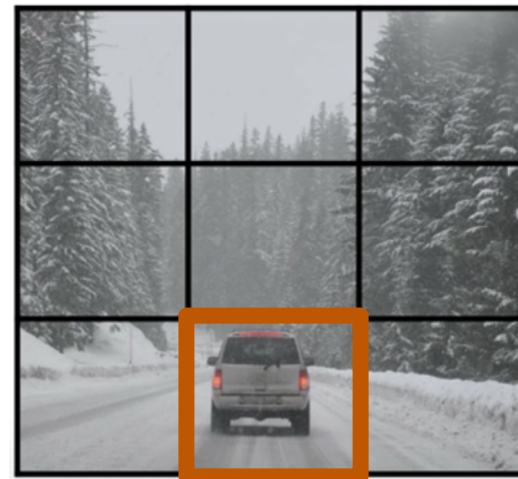
$S = 3, B = 2$

YOLO Training

- ❖ YOLO is a regression algorithm. What is X? What is Y?
- ❖ X is simple, just an image width (in pixels) * height (in pixels) * RGB values
- ❖ Y is a tensor of size $S * S * (B * 5 + C)$
- ❖ $B * 5 + C$ term represents the predictions + class predicted distribution for a grid block

For each grid block, we have a vector like this. For this example B is 2 and C is 2

p_1
b_x_1
b_y_1
b_h_1
b_w_1
p_2
b_x_2
b_y_2
b_h_2
b_w_2
c_1
c_2

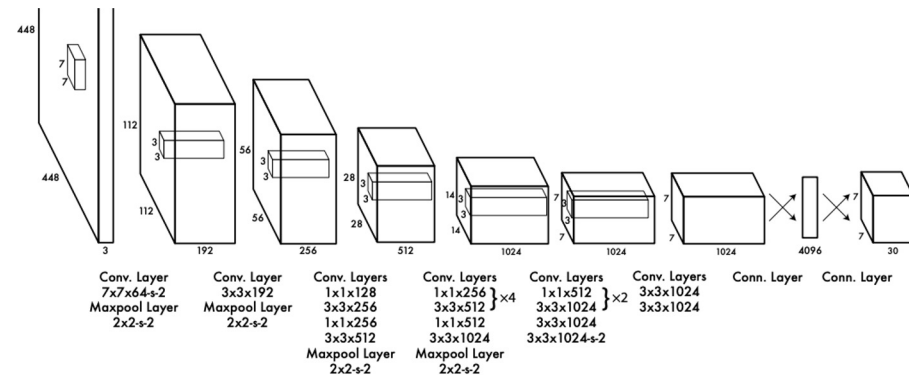


GT label
example:

1
b_x_1
b_y_1
b_h_1
b_w_1
0
?
?
?
?
c_1 = 1
c_2 = 0

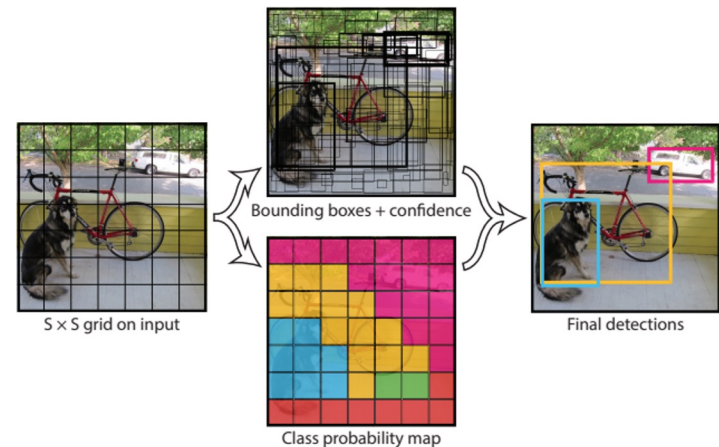
YOLO Architecture

- Now that we know the input and output, we can discuss the model
- We are given 448 by 448 by 3 as our input.
- Implementation uses 7 convolution layers
- Paper parameters: $S = 7$, $B = 2$, $C = 20$
- Output is $S * S * (5B + C) = 7 * 7 * (5 * 2 + 20) = 7 * 7 * 30$



YOLO Prediction

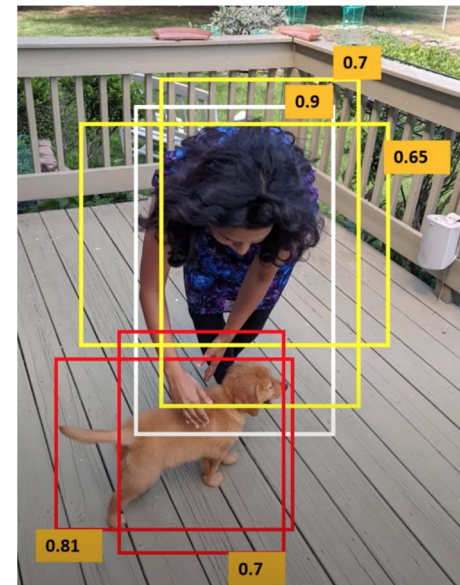
- ❖ We then use the output to make final detections
- ❖ Use a threshold to filter out bounding boxes with low $P(\text{Object})$
- ❖ In order to know the class for the bounding box compute score take argmax over the distribution $\Pr(\text{Class}|\text{Object})$ for the grid the bounding box's center is in



$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

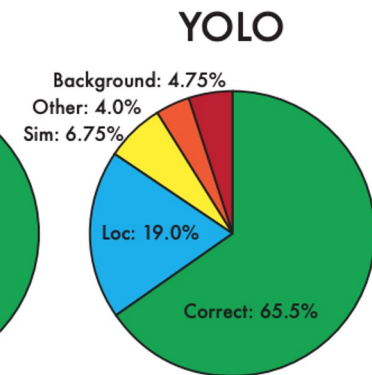
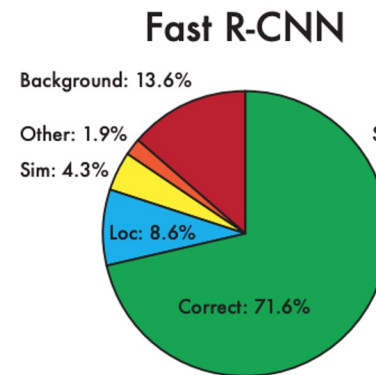
Non-maximal suppression

- ❖ Most of the time objects fall in one grid, however it is still possible to get redundant boxes (rare case as object must be close to multiple grid cells for this to happen)
- ❖ Discard bounding box with high overlap (keeping the bounding box with highest confidence)
- ❖ Adds 2-3% on final mAP score



Experimental Results - Error Analysis

- Makes far less background errors (less likely to predict false positives on background)
 - IOU is VERY small with any ground truth label
- But far more localization errors
 - Correct class, IOU is somewhat small



Localization error



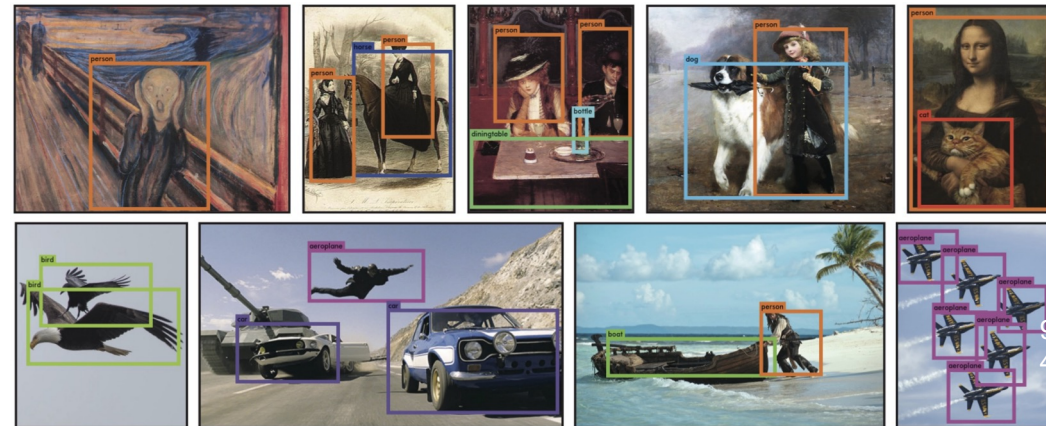
Background error



Experimental Results - Out of Domain

- ❖ Ran YOLO + competitors (trained on natural images) on art
- ❖ Does well on artistic datasets where more having global context greatly helps

	VOC 2007 AP	Picasso AP	Picasso Best F_1	People-Art AP
YOLO	59.2	53.3	0.590	45
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	



Discussion of Results

- ❖ Pro: YOLO is a lot faster than the other algorithms for image detection
- ❖ Pro: YOLO's use of global information rather than only local information allows it to understand contextual information when doing object detection
 - Does better in domains such as artwork due to this
- ❖ Con: YOLO lagged behind the SOTA models in object detection
 - This is attributed to making many localization errors and unable to detect small object

Critique / Limitations / Open Issues

- ❖ Performance lags behind SOTA
- ❖ Requires data to be labeled with bounding boxes, hard to collect for many classes
 - Previous work could generalize better since it used image classifier
 - 2014 COCO dataset (very large dataset) addressed this somewhat
- ❖ Regarding experiments: number of classes predicted is very limited
 - Not convinced that YOLO v1 is generalizable
- ❖ Confidence output of YOLO not confidence of class but $P(\text{Object})$, lowers interpretability
- ❖ Another limitation of YOLO is that it imposed spatial constraints on the objects in the image since only B boxes can be predicted on an $S \times S$ grid
- ❖ Since the architecture only predicts boxes, this might make it less useful for irregular shapes

YOLO Summary

- ❖ Object detection is the problem of detecting multiple objects in an image
- ❖ Almost real time object detection can make highly responsive robot systems without complex sensors
- ❖ Prior work relies on a large architecture with numerous parts to optimize
- ❖ YOLO proposes a unified architecture, which does all the tasks in one model and by one inference over the entire image
- ❖ They show enormous speed improvement and show that they can beat most other prior work in terms of mAPs

CLIP – Contrastive language-image pretraining

Acknowledgements

Learning Transferable Visual Models From Natural Language Supervision

Alec Radford, JongWook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever

OpenAI

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever:
Learning Transferable Visual Models From Natural Language Supervision. ICML **2021**: 8748-8763

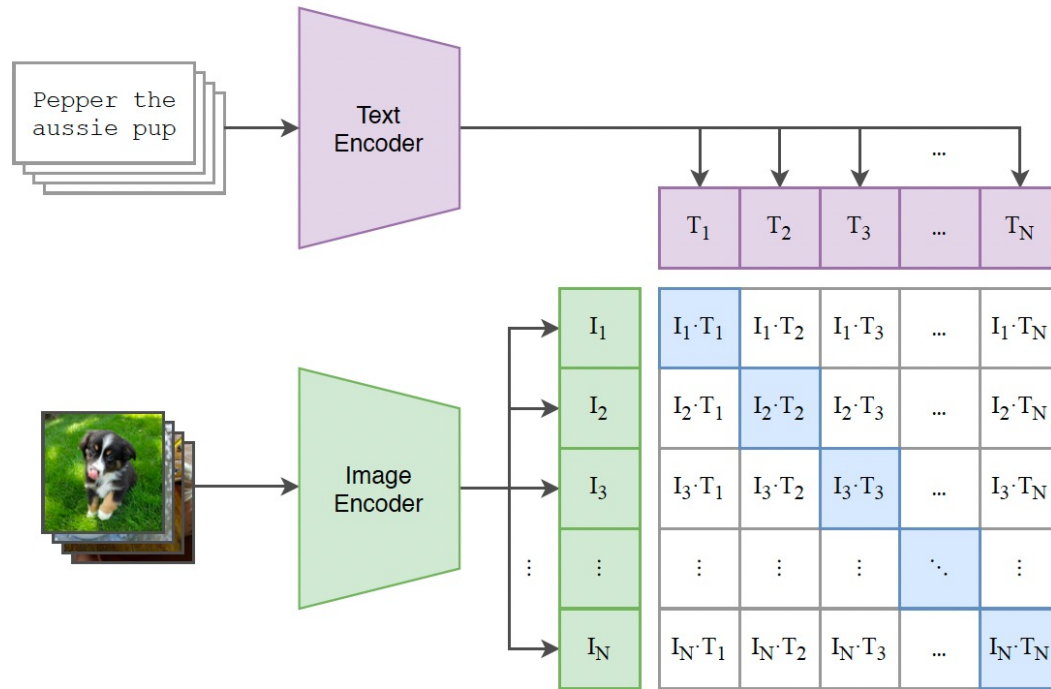
Contrastive pretraining

- During unsupervised contrastive pre-training,
- **the unlabeled images are clustered in the latent space,**
- **forming fairly good decision boundaries between different classes.**
- Based on this clustering, ...
- ... the subsequent supervised fine-tuning
- ... will achieve better performance than random initialization

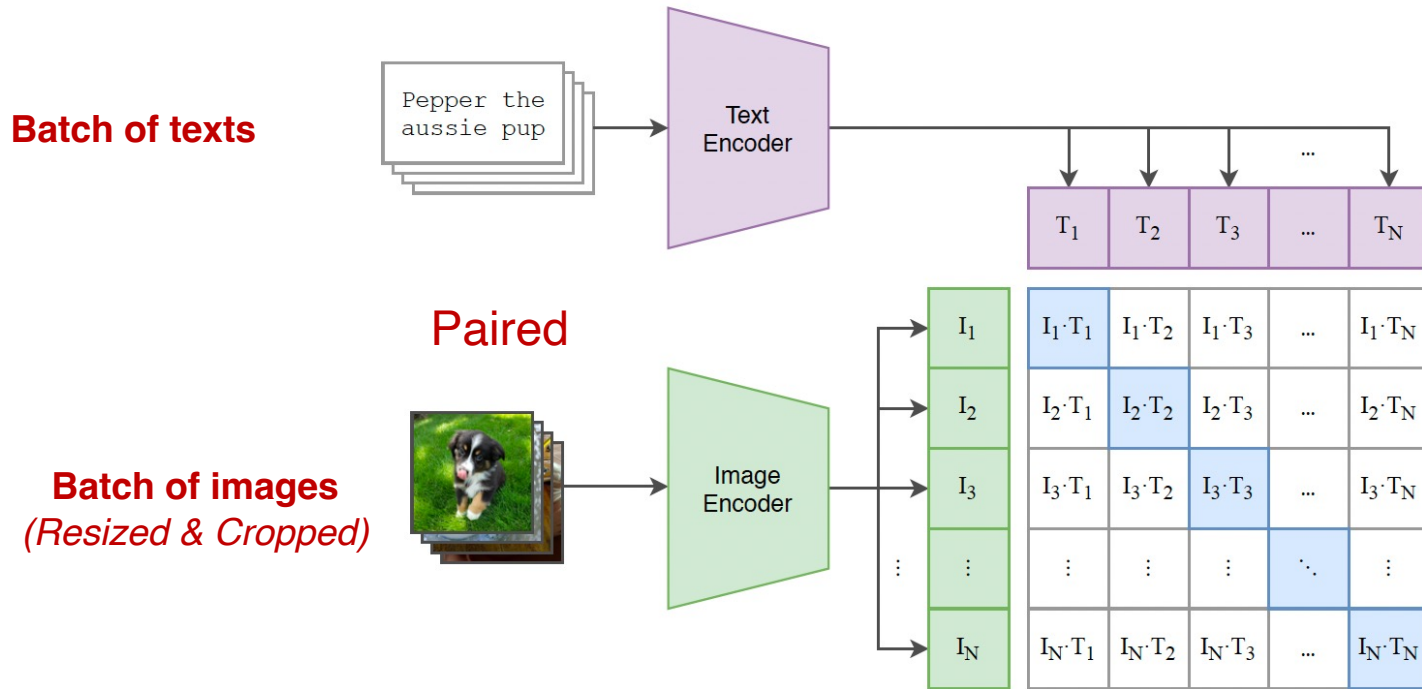
Contrastive language-image pretraining

- ViLBERT and similar methods (e.g., LXMERT) rely on small labeled datasets like MS COCO and Visual Genome (~100K images each)
- OpenAI collected 400 million (image, text) pairs from the web
- Then, they train an image encoder and a text encoder with a simple contrastive loss: given a collection of images and text, predict which (image, text) pairs actually occurred in the dataset

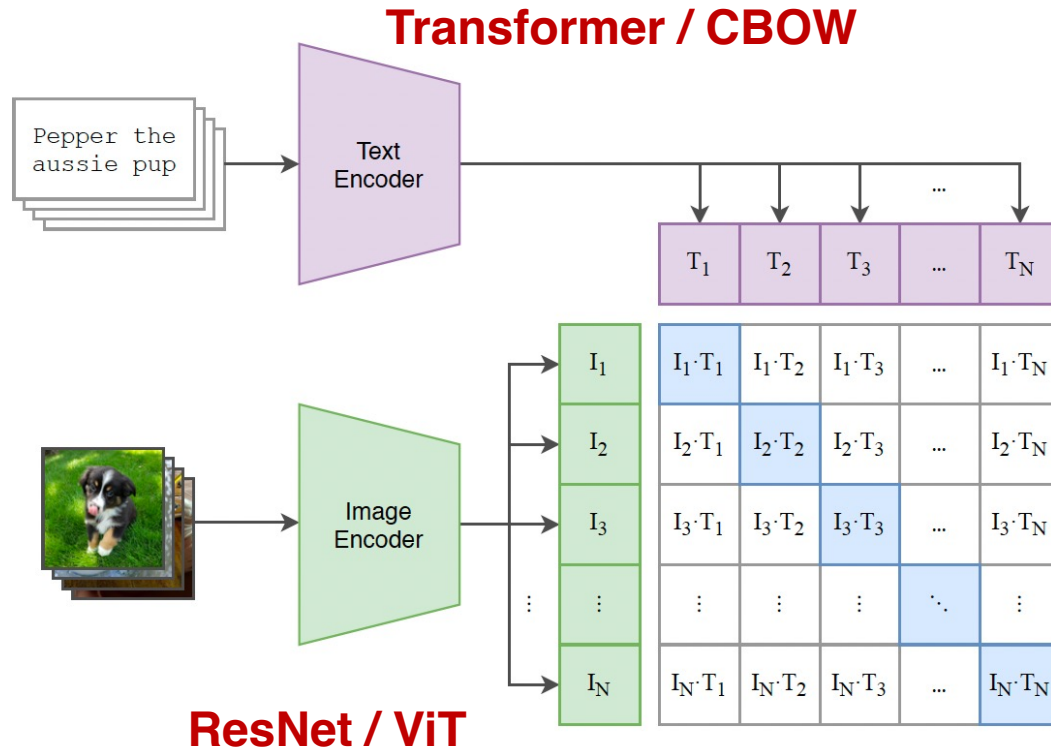
Method: Contrastive Pre-training



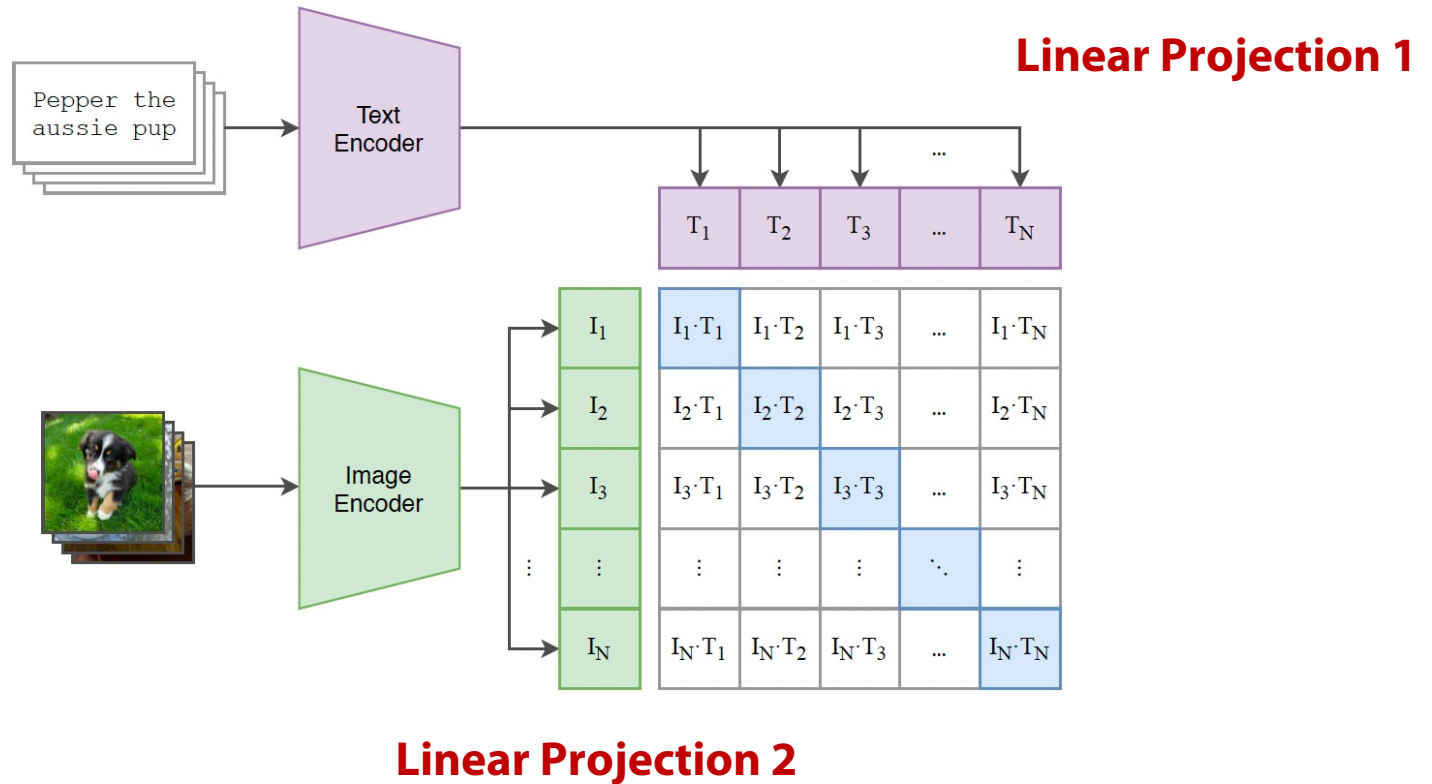
Method: Contrastive Pre-training



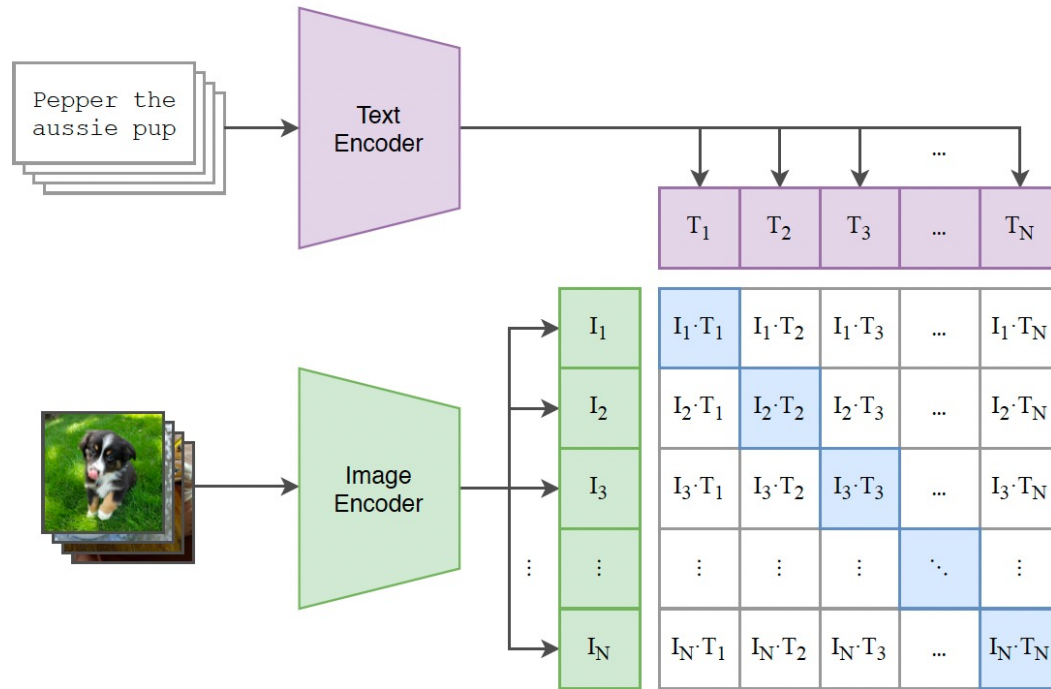
Method: Contrastive Pre-training



Method: Contrastive Pre-training

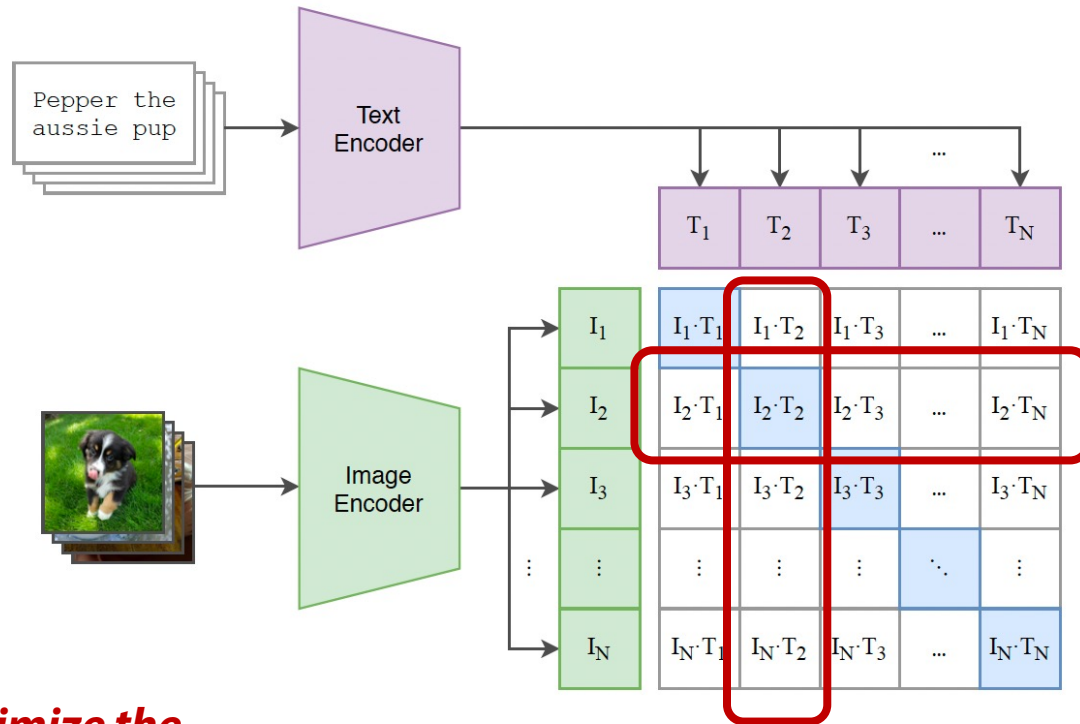


Method: Contrastive Pre-training



Cosine Similarity Matrix

Method: Contrastive Pre-training

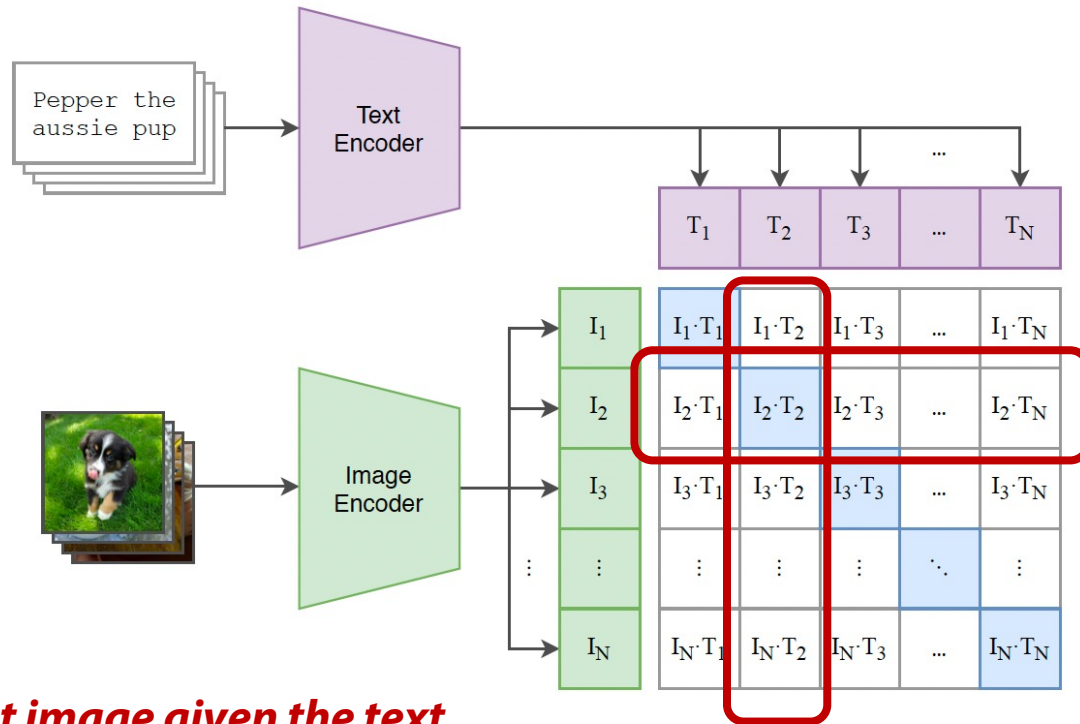


Maximize the diagonal and minimize the others

Cross-entropy on dimension 1

Cross-entropy on dimension 2

Method: Contrastive Pre-training



Cross-entropy on dimension 1

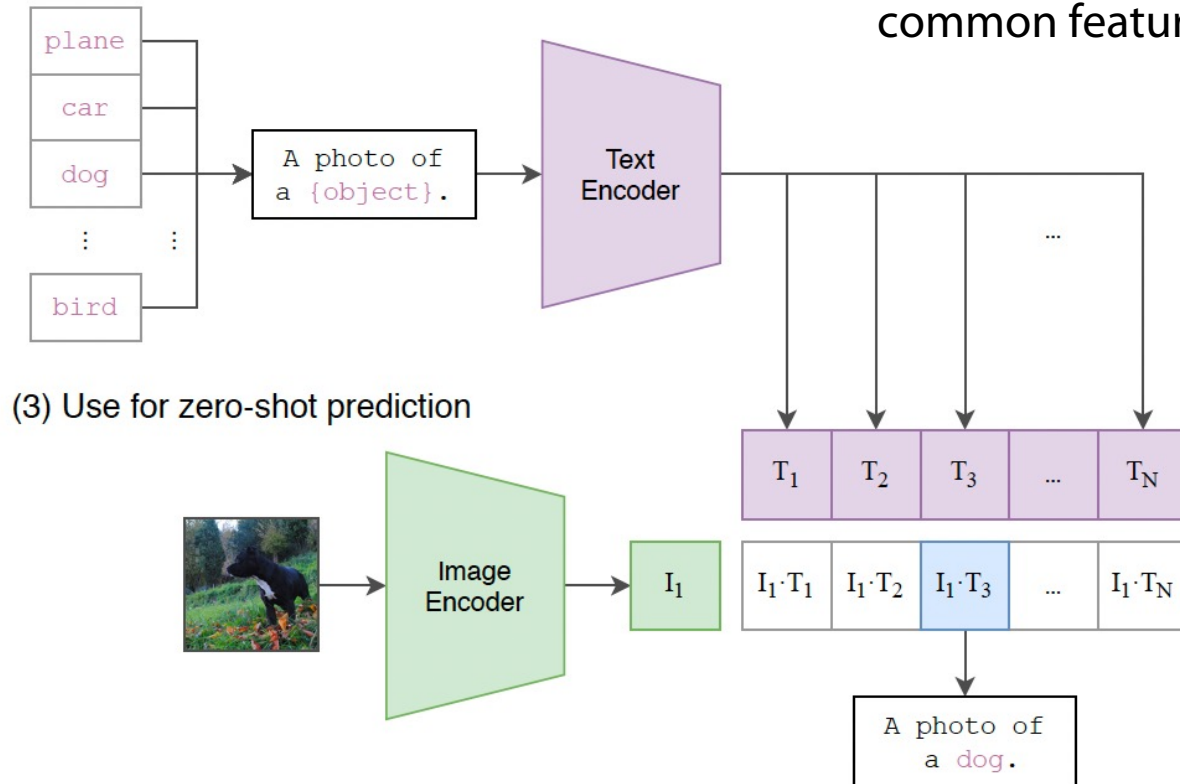
Cross-entropy on dimension 2

**Pick the right image given the text
&
Pick the right text given the image**

Method: Zero-Shot Testing

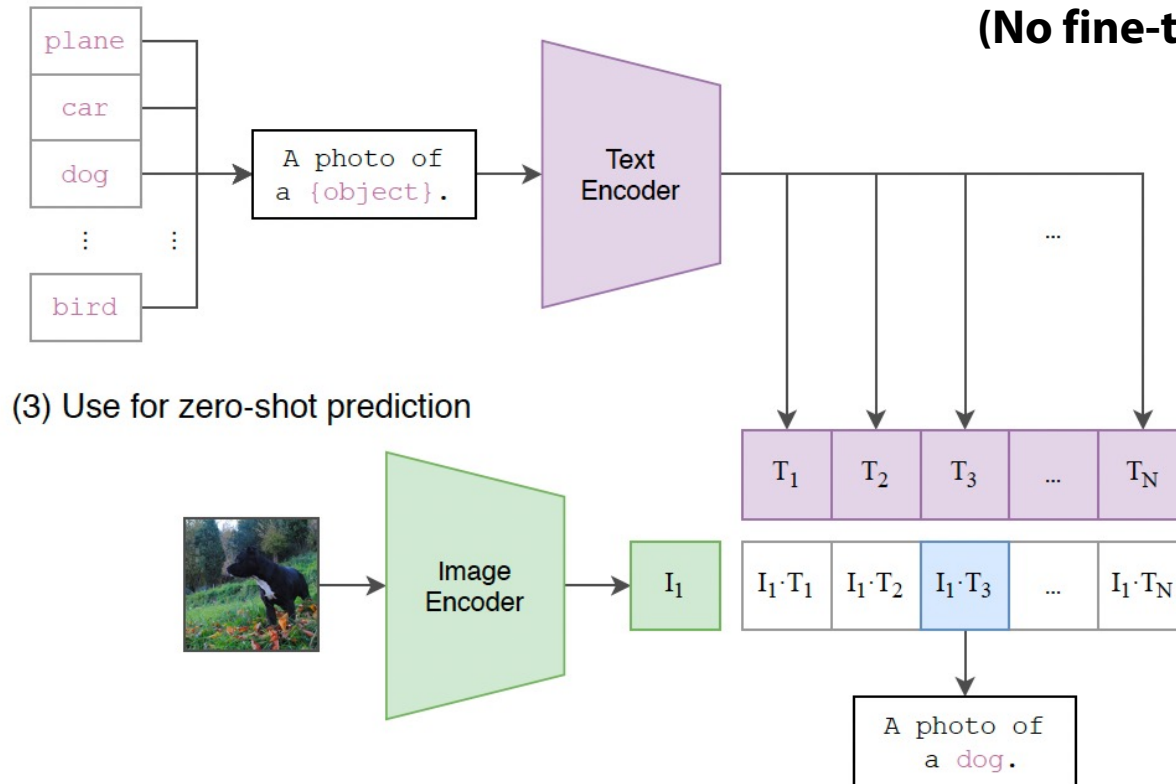
Core:

Images and text have been mapped into a common feature space



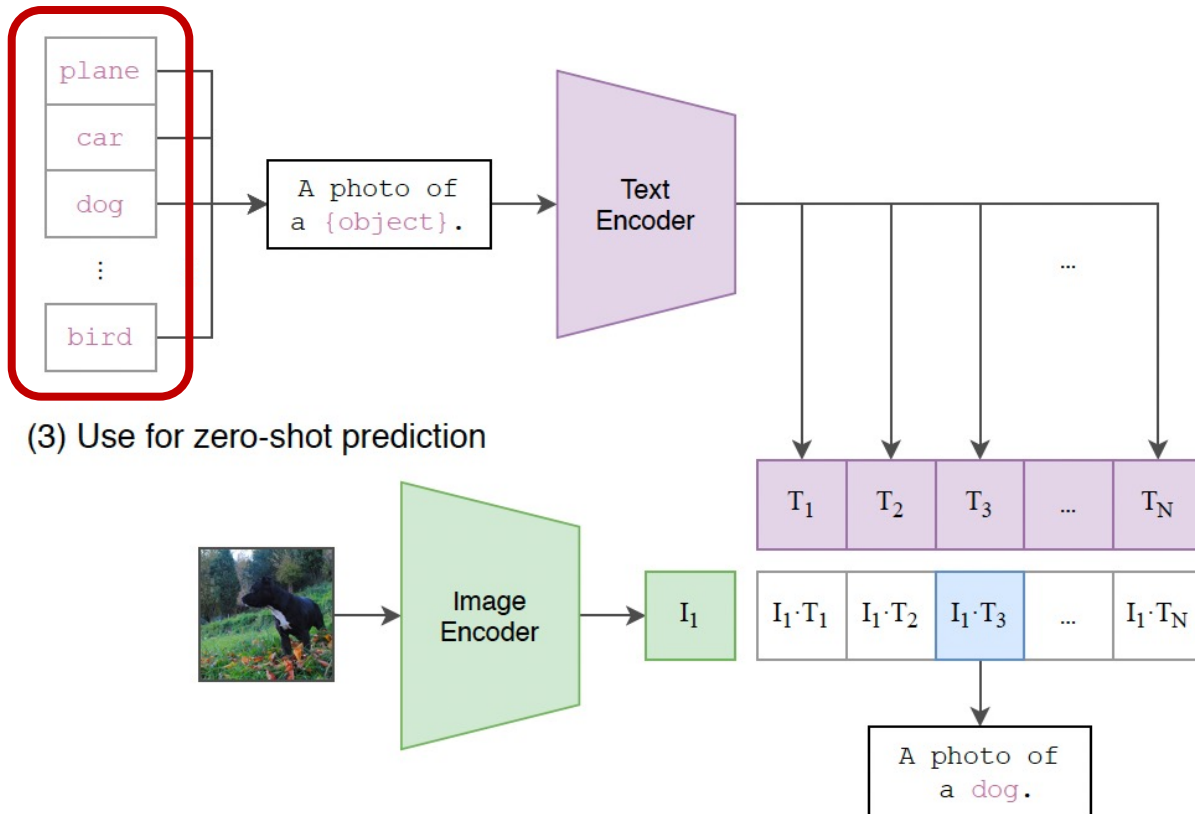
Method: Zero-Shot Testing

**The classes are not pre-defined
but chosen on demand
(No fine-tuning)**



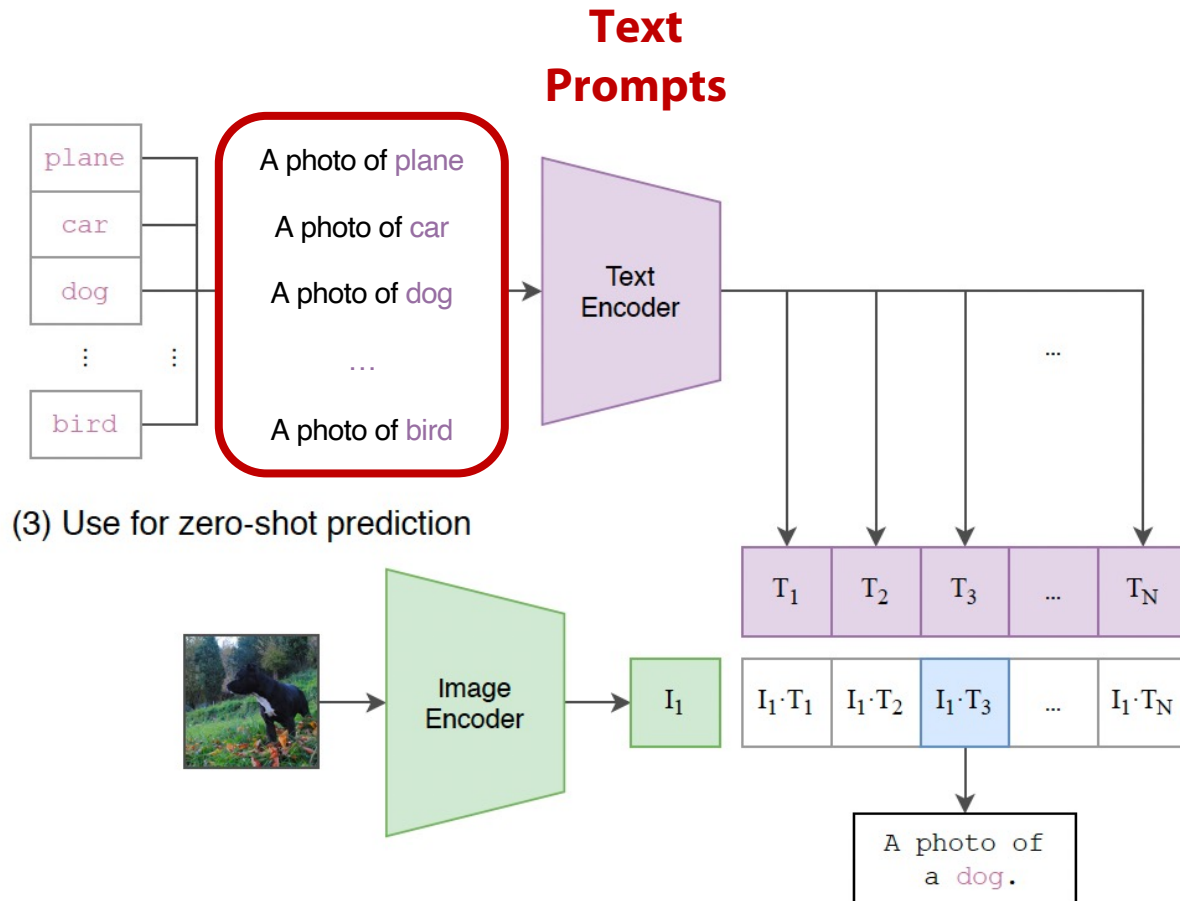
Method: Zero-Shot Testing

Classes on demand

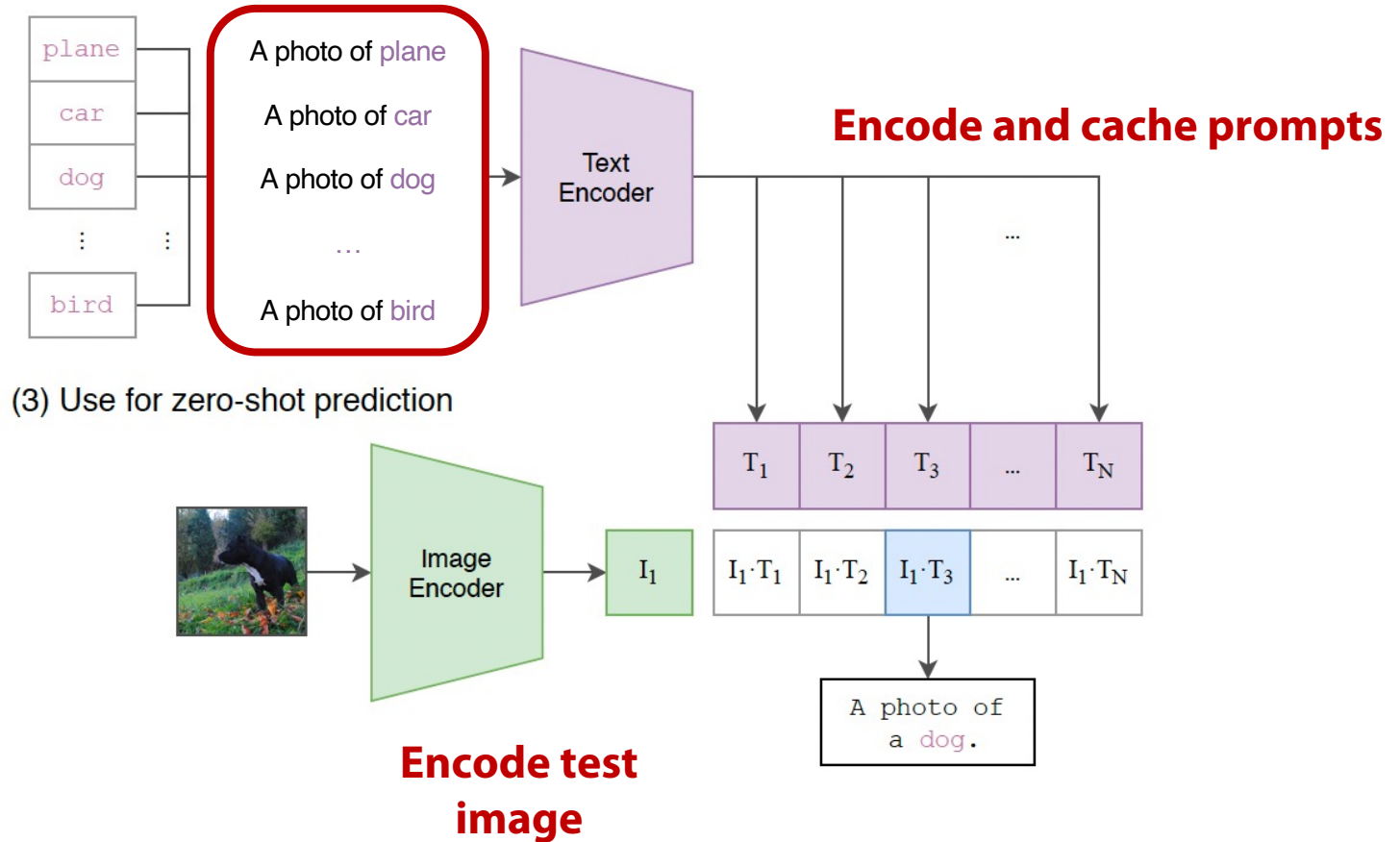


(3) Use for zero-shot prediction

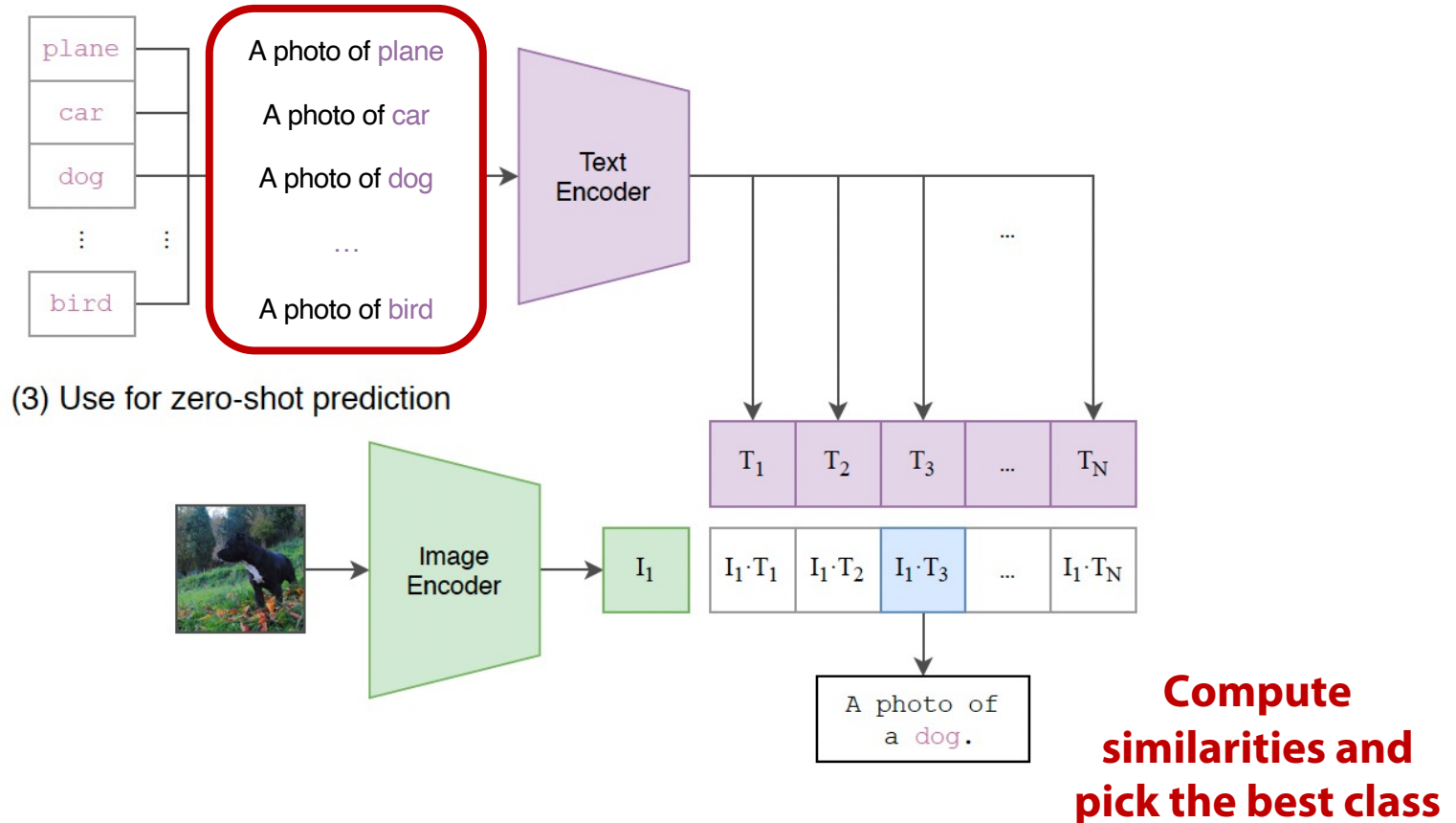
Method: Zero-Shot Testing



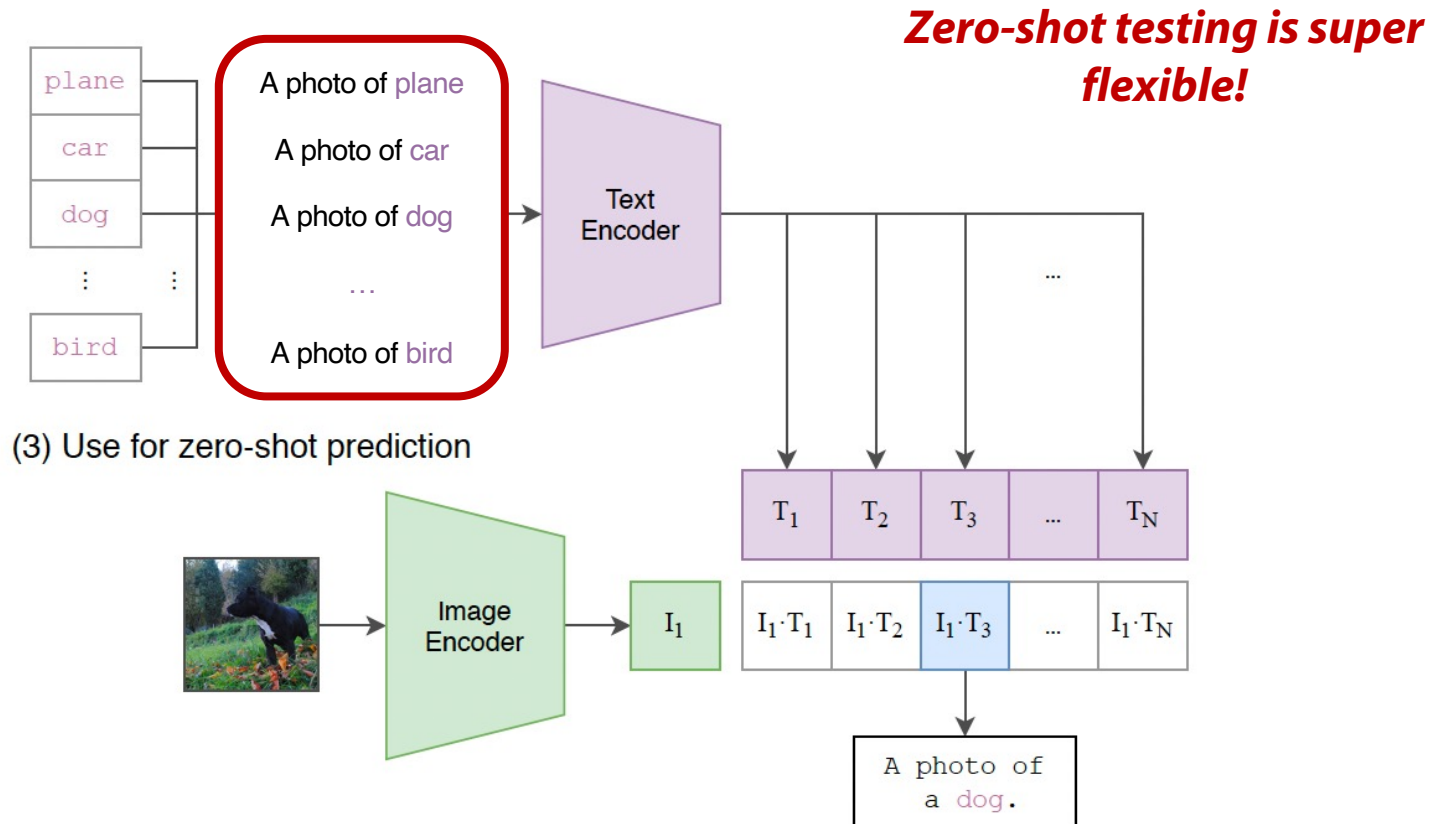
Method: Zero-Shot Testing



Method: Zero-Shot Testing



Method: Zero-Shot Testing



Method: Zero-Shot Testing – Prompt Engineering

Class names as baseline prompts

Problematic:

- A single word is often ambiguous, *i.e.*, the **dog 'boxer'** and the **athlete 'boxer'**
- *It is rare on the web that a image is paired with a single word*

Method: Zero-Shot Testing – Prompt Engineering

Class names as baseline prompts

Problematic:

- A single word is often ambiguous, *i.e.*, the **dog ‘boxer’** and the **athlete ‘boxer’**
- *It is rare on the web that a image is paired with a single word*

Prompt engineering examples:

A photo of a {label}.

(For general classification)

This is a {label}.

(For general classification)

A photo of a {label}, a type of pet.

(For pet classification)

A photo of a {label}, a type of food.

(For food classification)

A satellite photo of a {label}.

(For satellite image classification)

A digit “{label}”.

(For digit classification)

Food101

correct label: guacamole

correct rank: 1/101 correct probability: 90.15%



- a photo of guacamole, a type of food.
- a photo of ceviche, a type of food.
- a photo of edamame, a type of food.
- a photo of tuna tartare, a type of food.
- a photo of hummus, a type of food.



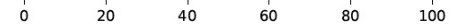
SUN397

correct label: television studio

correct rank: 1/397 correct probability: 90.22%



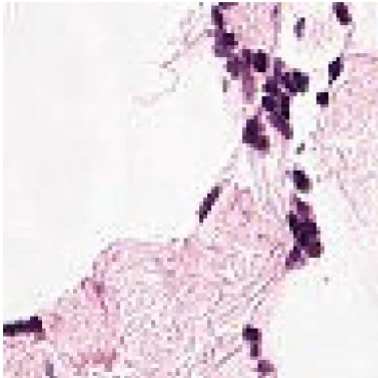
- a photo of a television studio.
- a photo of a podium indoor.
- a photo of a conference room.
- a photo of a lecture room.
- a photo of a control room.



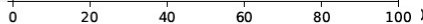
PatchCamelyon (PCam)

correct label: healthy lymph node tissue

correct rank: 2/2 correct probability: 22.81%

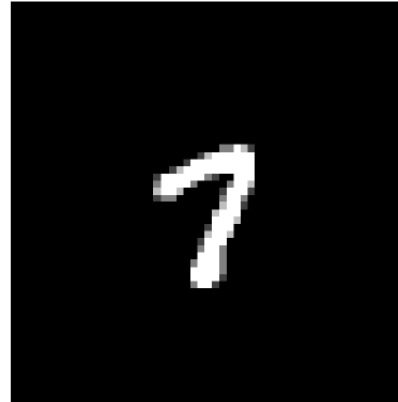


- this is a photo of lymph node tumor tissue
- this is a photo of healthy lymph node tissue

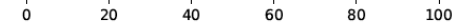


correct label: 7

correct rank: 1/10 correct probability: 85.32%



- a photo of the number: "7".
- a photo of the number: "2".
- a photo of the number: "1".
- a photo of the number: "6".
- a photo of the number: "4".



Method: Zero-Shot Testing – Prompt Engineering

Class names as baseline prompts

Problematic:

- A single word is often ambiguous, *i.e.*, the **dog ‘boxer’** and the **athlete ‘boxer’**
- *It is rare on the web that a image is paired with a single word*

Prompt engineering examples:

A photo of a {label}.

(For general classification)

This is a {label}.

(For general classification)

A photo of a {label}, a type of pet.

(For pet classification)

A photo of a {label}, a type of food.

(For food classification)

A satellite photo of a {label}.

(For satellite image classification)

A digit "{label}".

(For digit classification)

Prompt ensemble examples (average the prompt features):

A photo of a {label}.

A photo of a small {label}.

A photo of a big {label}.

(This could match the object no matter its size)

Method: Zero-Shot Testing – Prompt Engineering

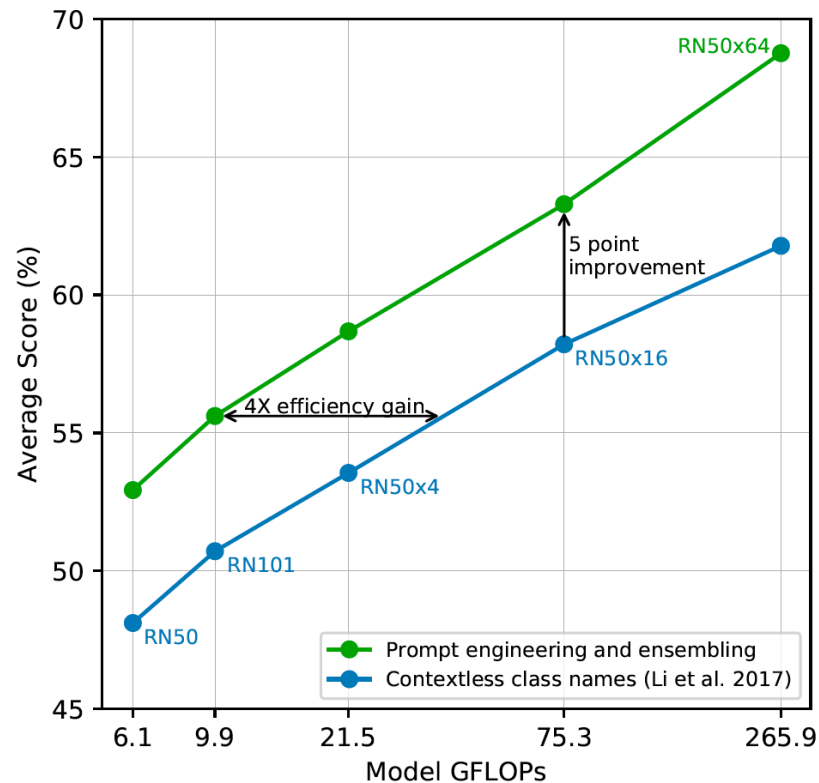
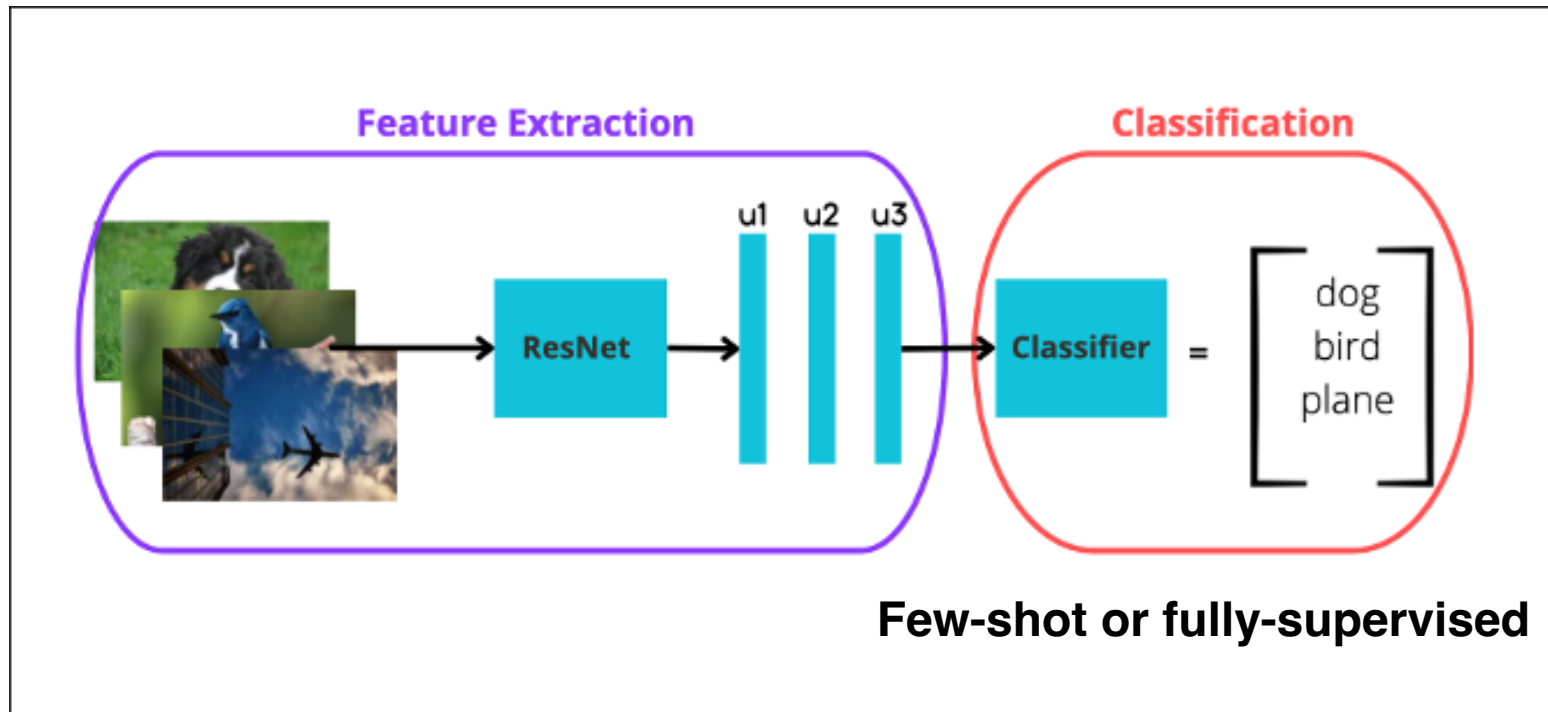


Figure 4. Prompt engineering and ensembling improve zero-shot performance.

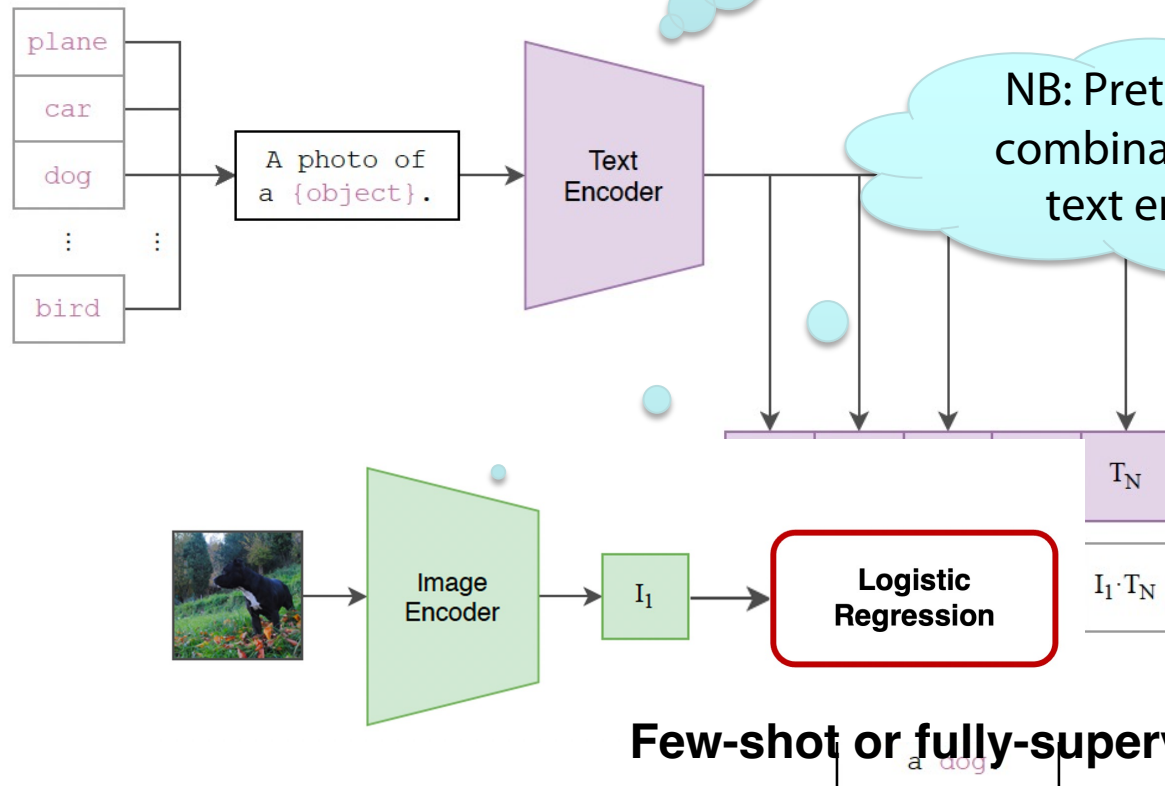
Compare with decimated image classifier?



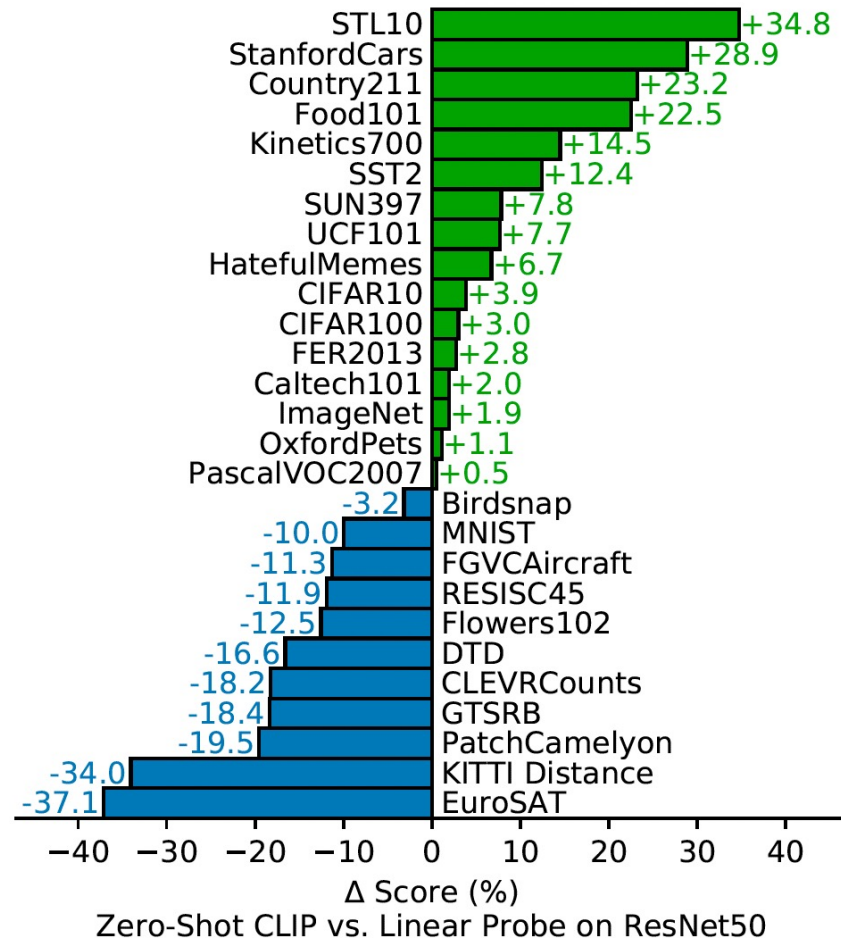
- For training, class labels must be known beforehand
- Using an image extractor paired with a classifier is also known as **linear probe evaluation**

Linear Probe CLIP

Use only the **CLIP's Image Encoder** to get the image features and fed them into a linear classifier. Even with this setup, **CLIP's** few-shot-learning capabilities are outstanding.

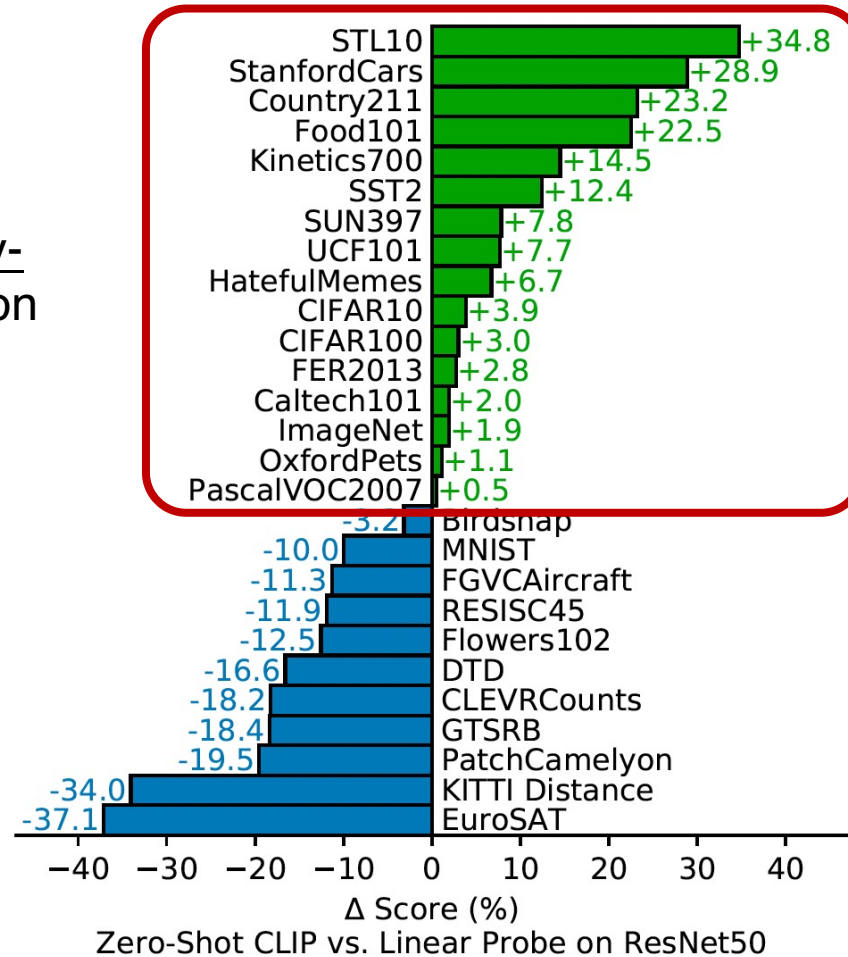


Experiments: Zero-shot



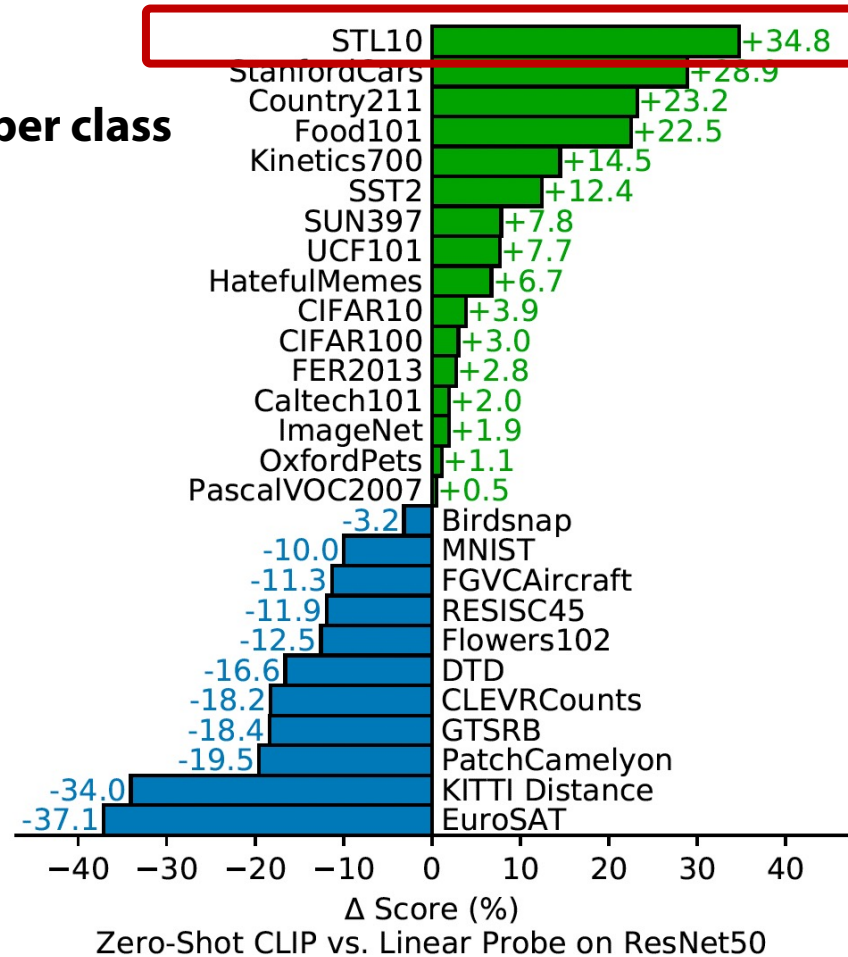
Experiments: Zero-shot

Zero-shot CLIP outperforms fully-supervised ResNet linear probe on 16 datasets



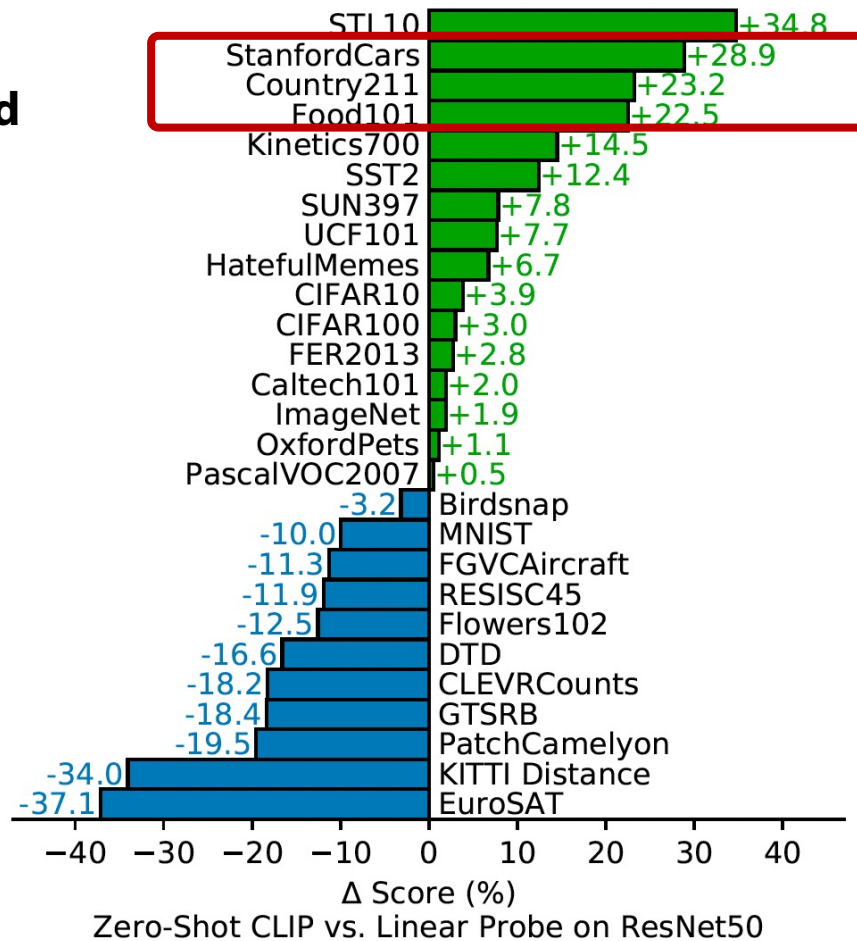
Experiments: Zero-shot

Limited examples per class



Experiments: Zero-shot

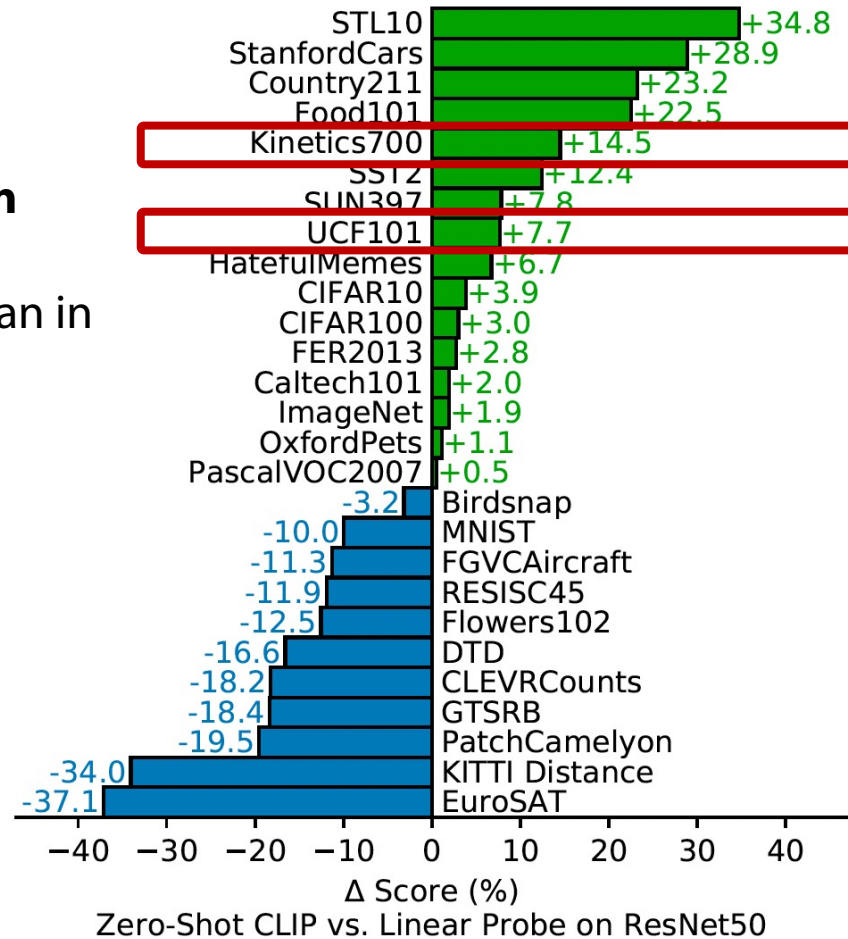
Fine-grained



Experiments: Zero-shot

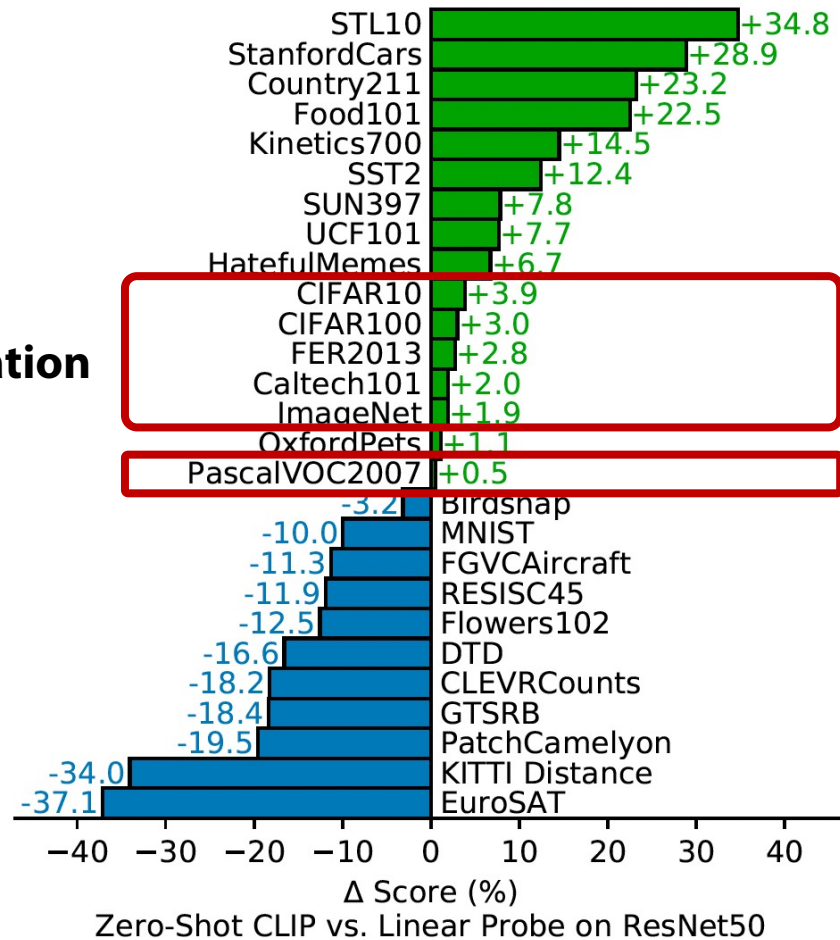
Action recognition

(More verbs on web than in ImageNet)



Experiments: Zero-shot

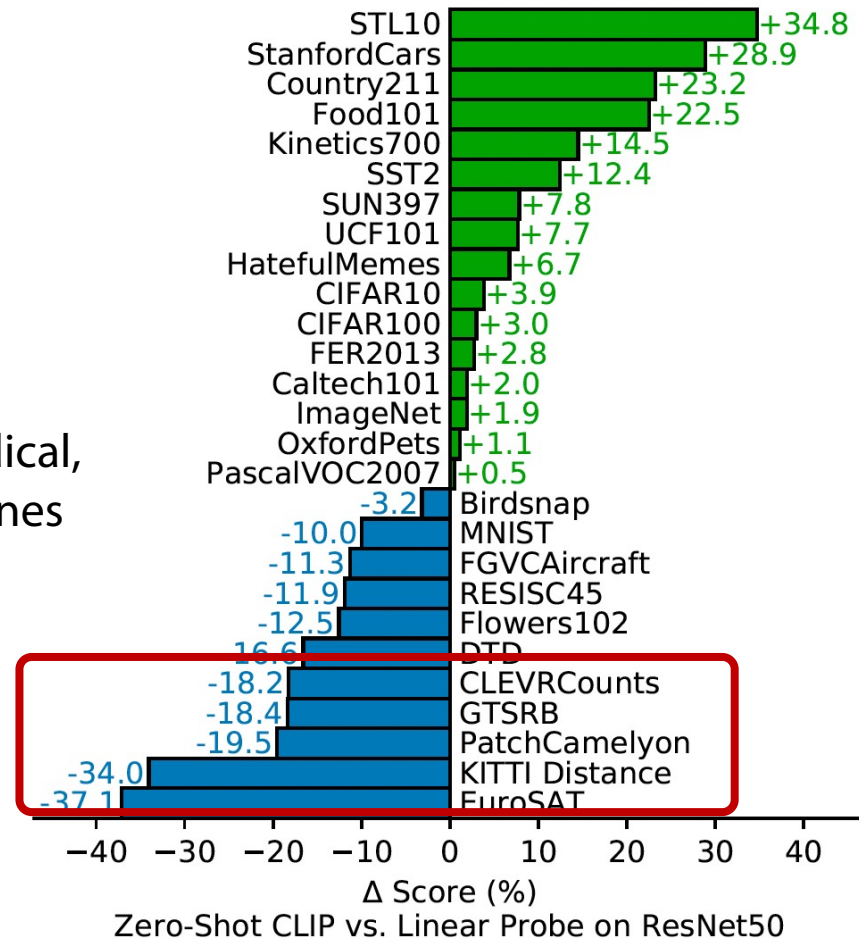
General classification



Experiments: Zero-shot

Specialized: satellite, medical,
self-driving, synthetic scenes

(Rare on web)



Experiments: Zero-shot

Still large room for zero-shot CLIP

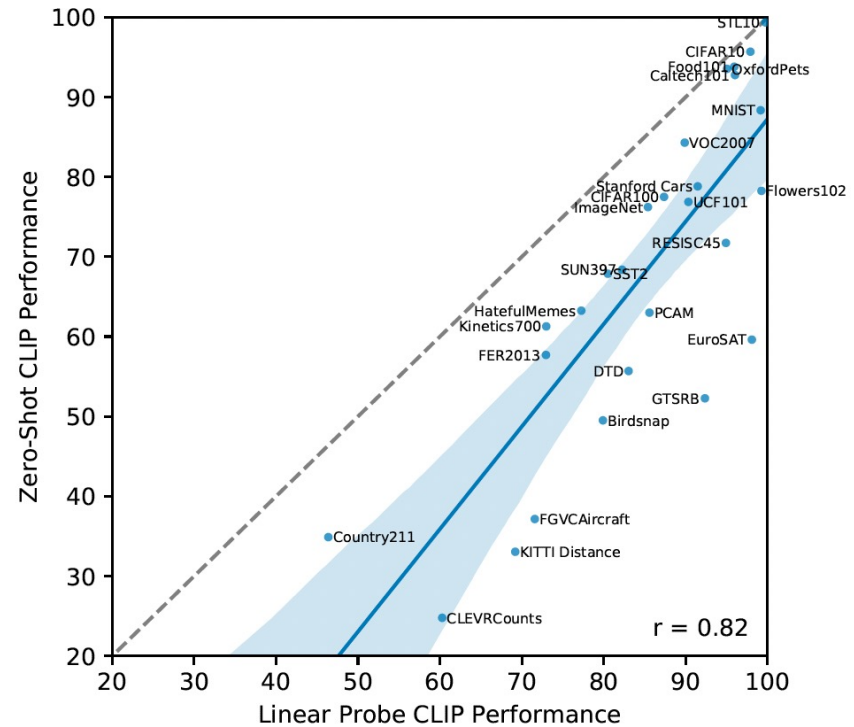


Figure 8. Zero-shot performance is correlated with linear probe performance but still mostly sub-optimal.

Experiments: Zero-shot

More compute power
could help

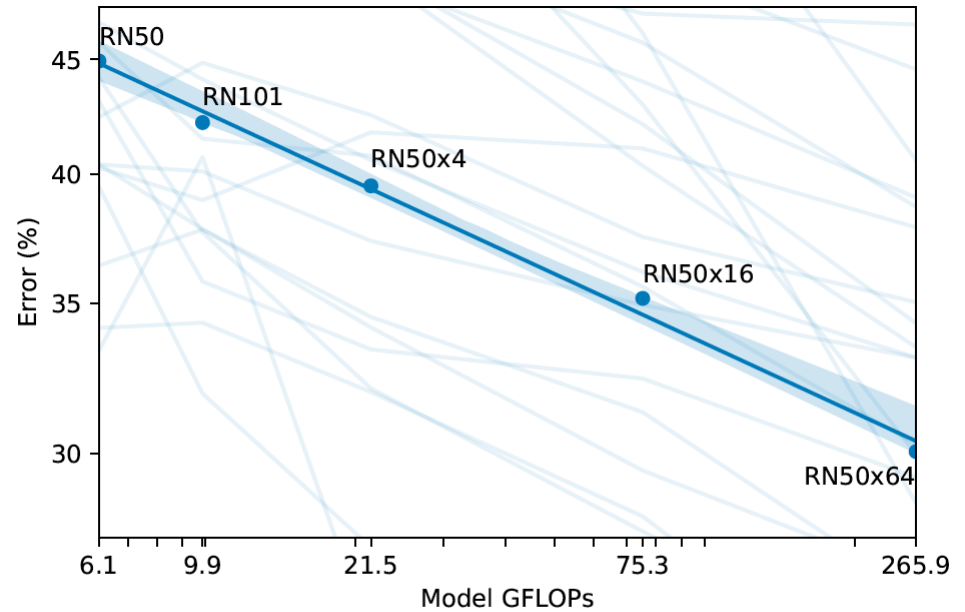


Figure 9. Zero-shot CLIP performance scales smoothly as a function of model compute power.

Experiments: Few-shot

Zero-shot CLIP = 4-shot Linear CLIP

Few-shot Linear CLIP > Others

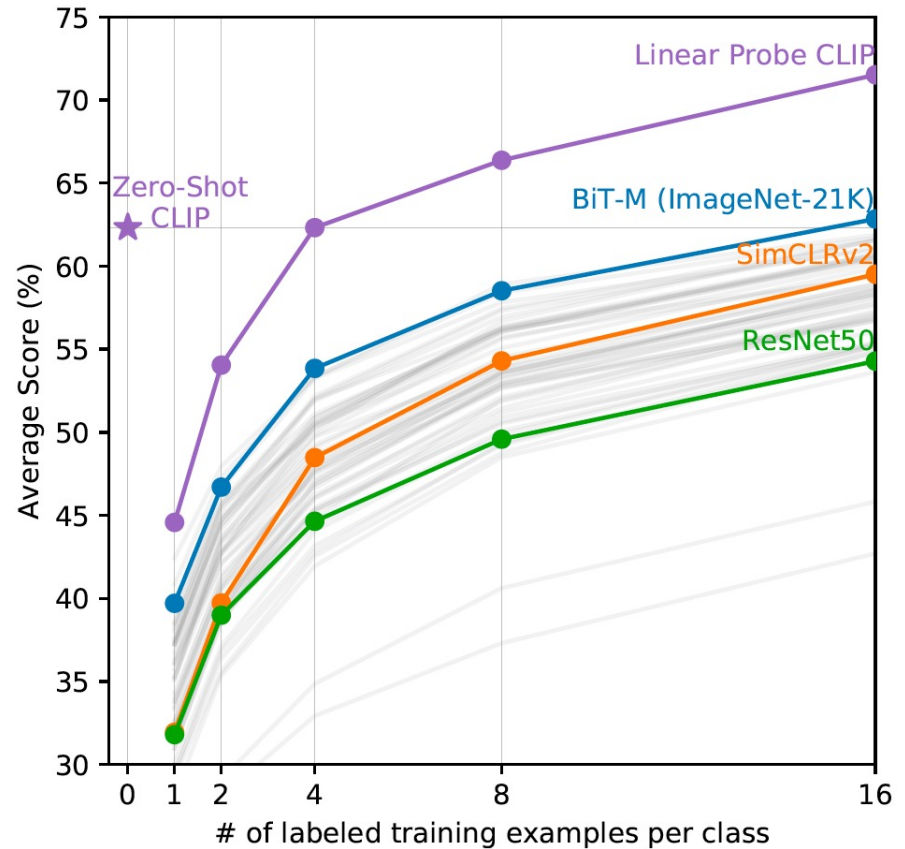


Figure 6. Zero-shot CLIP outperforms few-shot linear probes.

Experiments: Linear probe

Linear probe CLIP is STOA

Event better on more diverse datasets

Transformer is better than ConvNet with enough data

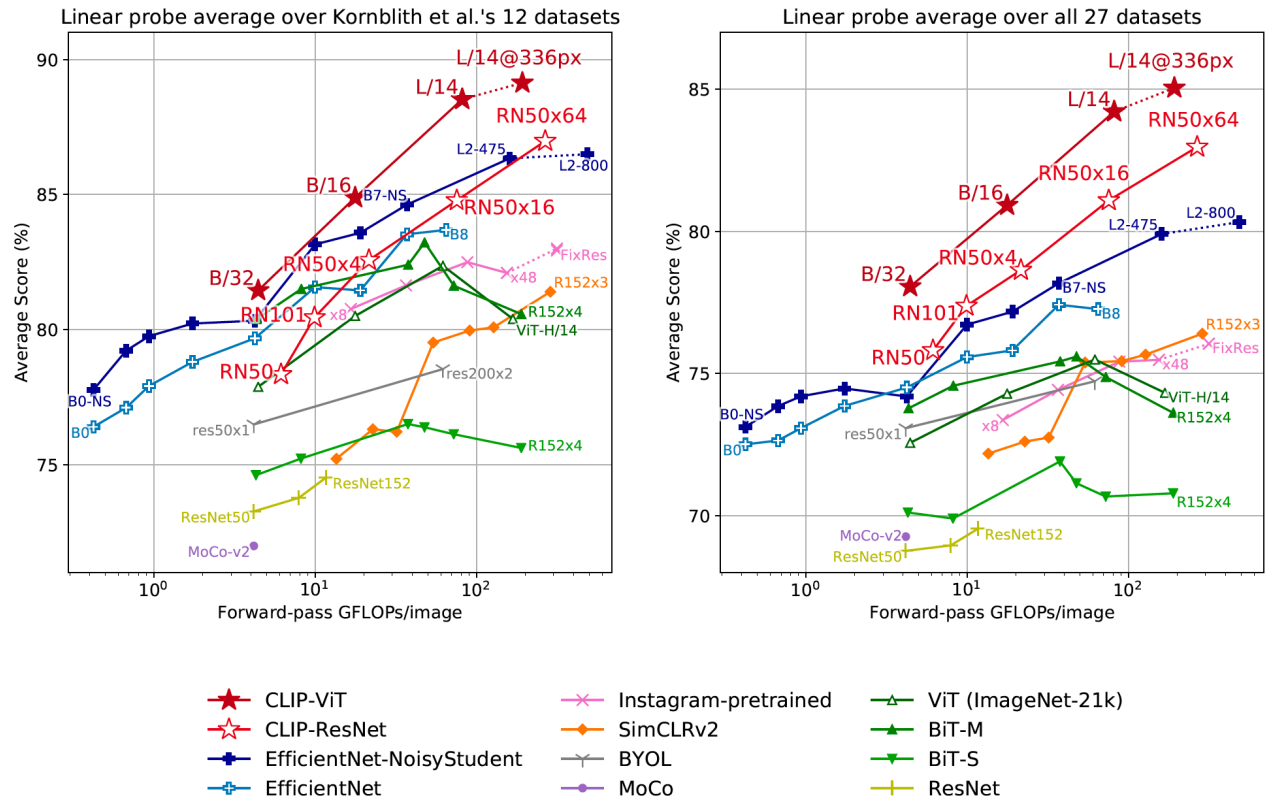


Figure 10. Linear probe performance of CLIP in comparison with state-of-the-art computer vision models, including

ImageNet-like datasets

More diverse datasets

Experiments: CLIP is more robust to domain shift

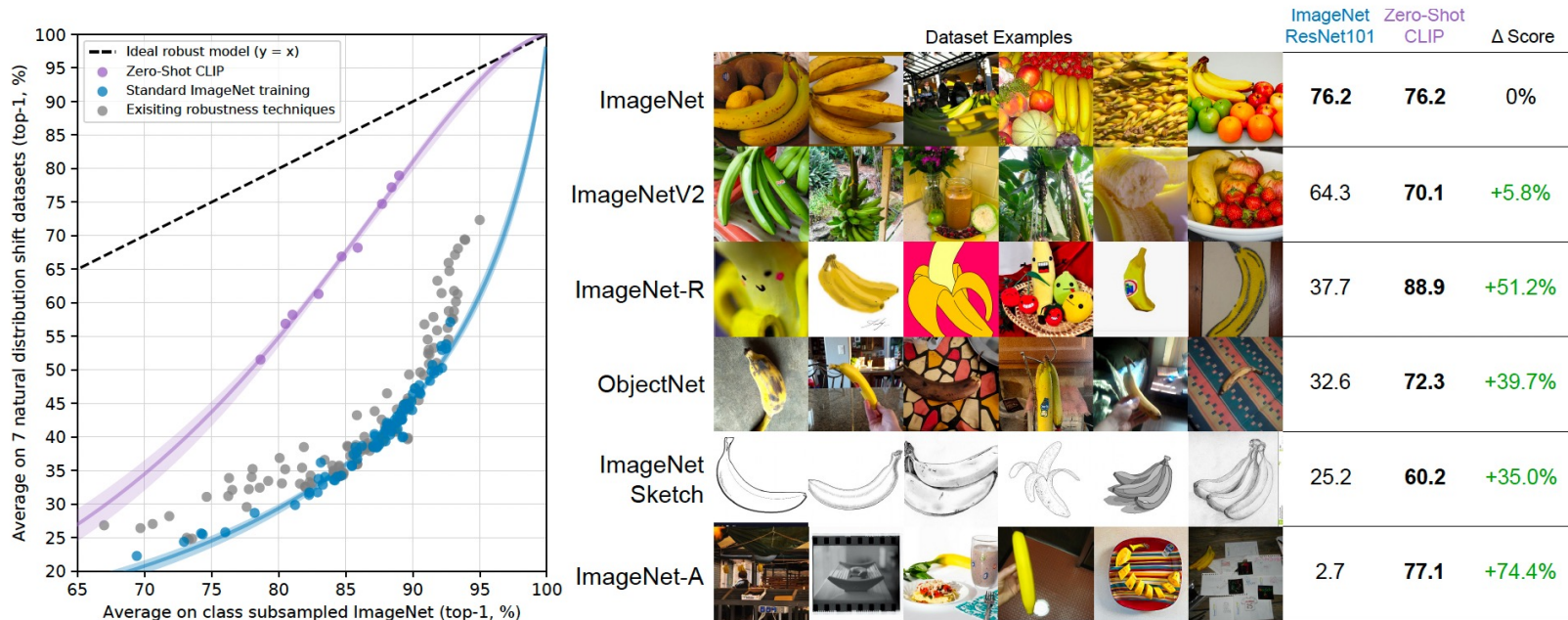


Figure 13. Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models. (Left) An ideal robust model

Experiments: CLIP is more robust to domain shift

Semantically similar datasets in similar or distinct domains

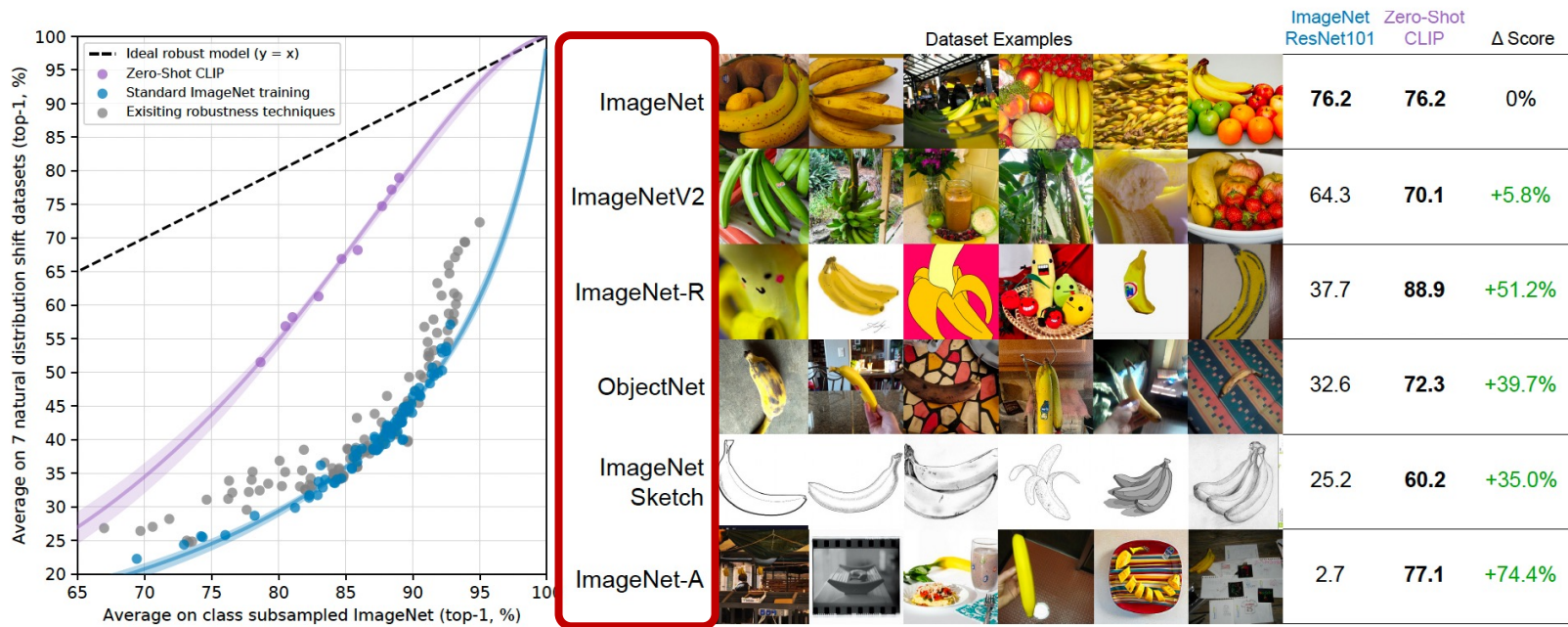


Figure 13. Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models. (Left) An ideal robust model

Experiments: CLIP is more robust to domain shift

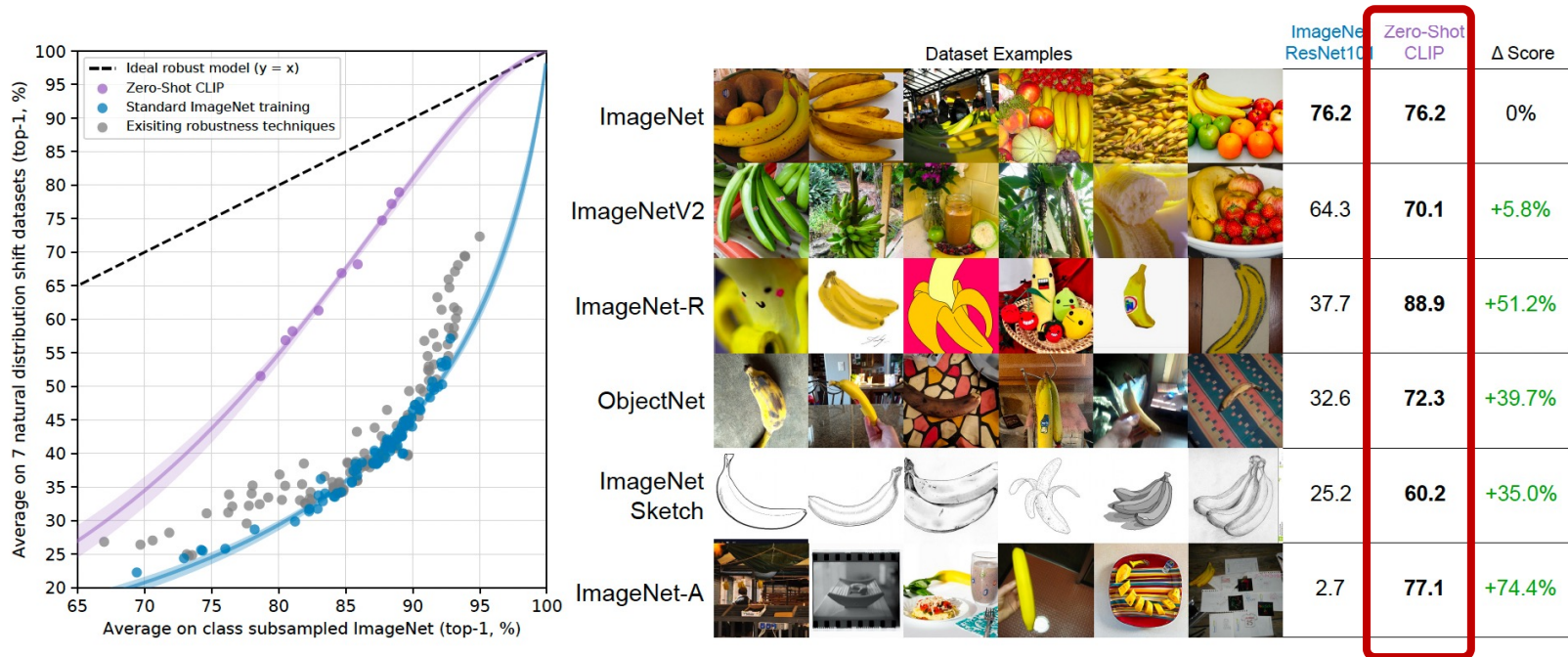


Figure 13. Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models. (Left) An ideal robust model

Zero-shot CLIP is robust

Experiments: CLIP is more robust to domain shift

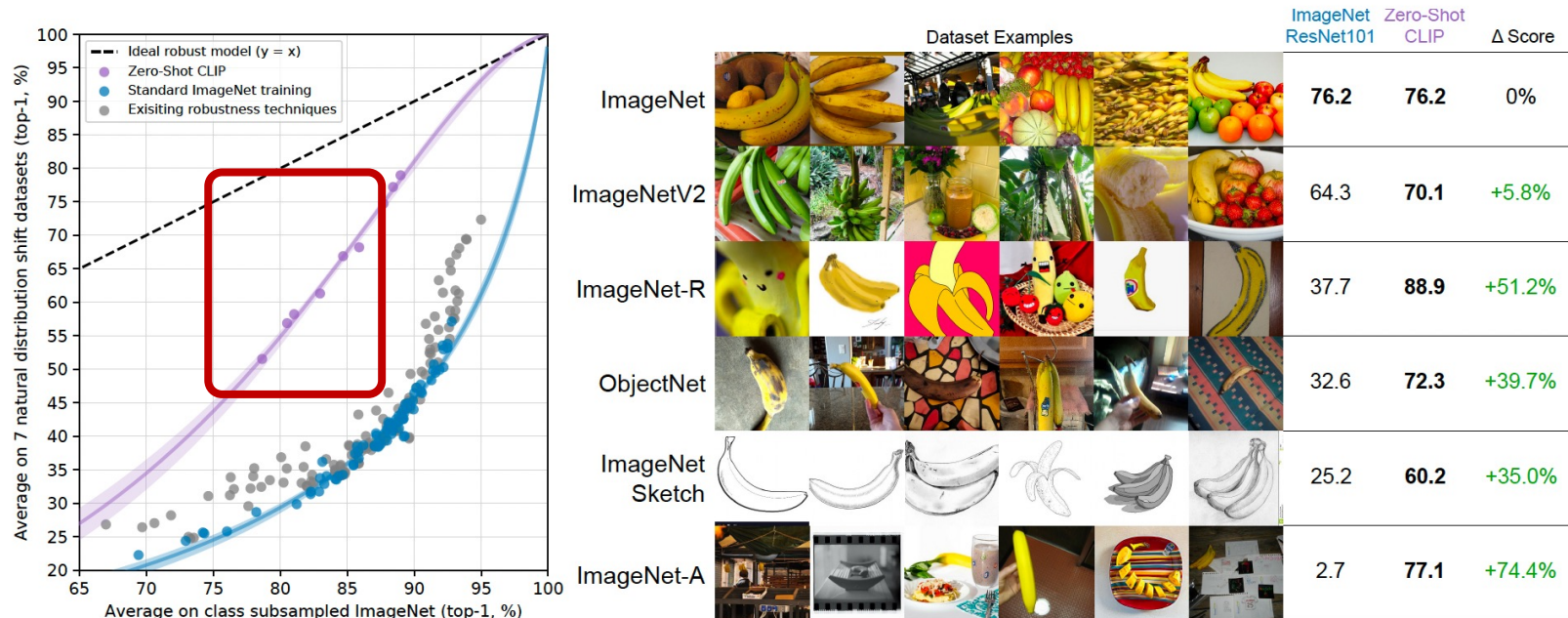


Figure 13. Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models. (Left) An ideal robust model

Zero-shot CLIP is robust

Code Released

```
import torch
import clip
from PIL import Image

device = "cuda" if torch.cuda.is_available() else "cpu"
model, preprocess = clip.load("ViT-B/32", device=device)

image = preprocess(Image.open("CLIP.png")).unsqueeze(0).to(device)
text = clip.tokenize(["a diagram", "a dog", "a cat"]).to(device)

with torch.no_grad():
    image_features = model.encode_image(image)
    text_features = model.encode_text(text)

    logits_per_image, logits_per_text = model(image, text)
    probs = logits_per_image.softmax(dim=-1).cpu().numpy()

print("Label probs:", probs) # prints: [[0.9927937  0.00421068  0.00299572]]
```

Code Released

```
import torch
import clip
from PIL import Image

device = "cuda" if torch.cuda.is_available() else "cpu"
model, preprocess = clip.load("ViT-B/32", device=device)

image = preprocess(Image.open("CLIP.png")).unsqueeze(0).to(device)
text = clip.tokenize(["a diagram", "a dog", "a cat"]).to(device)

with torch.no_grad():
    image_features = model.encode_image(image)
    text_features = model.encode_text(text)

    logits_per_image, logits_per_text = model(image, text)
    probs = logits_per_image.softmax(dim=-1).cpu().numpy()

print("Label probs:", probs) # prints: [[0.9927937  0.00421068  0.00299572]]
```

Easy to get CLIP features

Conclusion CLIP

Multi-modal pre-training on a web scale gives STOA performances

Zero-shot may enable a new paradigm to develop vision systems

- No data annotation, model training, hyper-parameter tuning is needed
- Only `import clip` and design the prompts
- Especially for non-specialized tasks
- At least, CLIP features are more accurate and robust than ResNet features

Images and languages are mapped into a common space

- This is how human understand concepts
- Towards general intelligence
- But currently, more like a super fuzzy reverse search engine

Easy to use:

- Released codes and models
- Unreleased data and prompts

GPT-4V(ision)

Sec. 2.3 Interleaved Image-text Inputs

Interleaved Image-text Pairs

Prompt:

How much did I pay for tax? [Prompt Continued](#)
Where is the information?



GPT-4V:

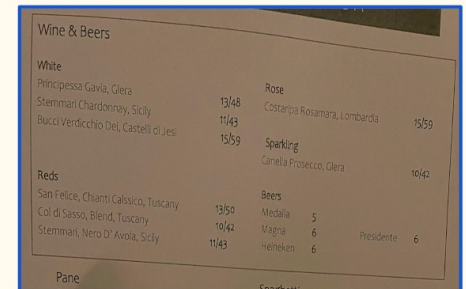
In the first receipt, you paid \$3.72 for tax. The information is located at the bottom of the receipt. In the second receipt, you paid \$42.23 for tax. The information is located near the bottom of the receipt, under the "Sales Tax" line. In the third receipt, you paid \$4.58 for tax. The information is located at the bottom of the receipt, under the "TAX" line. In total, you paid \$50.53 in tax across all three receipts.

Prompt:

How much should I pay for the beer on the table according to the price on the menu?



Here is a photo of the menu.



GPT-4V:

According to the menu, the price for a Magna beer is \$6. Since there are two Magna beers on the table, the total cost for the beers would be \$12.

The Dawn of LMMs: Preliminary Explorations with GPT-4V(ision)

Zhengyuan Yang*, Linjie Li*, Kevin Lin*, Jianfeng Wang*, Chung-Ching Lin*,
Zicheng Liu, Lijuan Wang*♠
Microsoft Corporation

* Core Contributor ♠ Project Lead

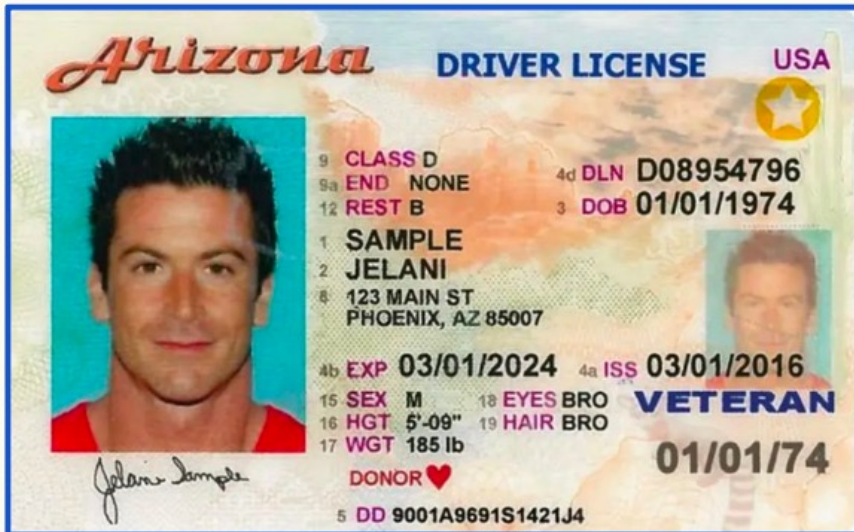


Constrained Prompting

Prompt:

Please read the text in this image and return the information in the following JSON format (note xxx is placeholder, if the information is not available in the image, put "N/A" instead).

```
{"class": xxx, "DLN": xxx, "DOB": xxx, "Name": xxx, "Address": xxx, "EXP": xxx, "ISS": xxx, "SEX": xxx, "HGT": xxx, "WGT": xxx, "EYES": xxx, "HAIR": xxx, "DONOR": xxx}
```



GPT-4V:

```
{"class": "D", "DLN": "D08954796", "DOB": "01/01/1974", "Name": "Jelani Sample", "Address": "123 Main St, Phoenix, AZ 85007", "EXP": "03/01/2024", "ISS": "03/01/2016", "SEX": "M", "HGT": "5-09", "WGT": "185 lbs", "EYES": "BRO", "HAIR": "N/A", "DONOR": "Y"}
```


Condition on Good Performance

Prompt:

Count the number of apples in the image.



GPT-4V:

There are 12 apples in the image.

Prompt:

Count the number of apples in the image. Let's think step-by-step.



Note: Same query image as shown above, resized to save space

GPT-4V:

Step 1: Look at the image and identify all the apples.

Step 2: Start counting the apples one by one.

Step 3: Make sure not to miss any apples or count the same apple twice.

Step 4: After counting all the apples, double check to make sure the count is correct.

Answer: There are 12 apples in the image.

Visual Pointing and Visual Referring Prompting

(1) Prompt:

Describe the pointed region in the image.



GPT-4V:

The pointed region in the image is a row of hanging lights on a wire. The lights are small and round, and they are evenly spaced along the wire. The wire is stretched between two poles and is located above a patio area with tables and umbrellas. The lights appear to be decorative and are likely used to provide ambient lighting in the evening.

(3) Prompt:

What is in the circled glass? Object 1 or 2? Describe what is Object 1 and object 2. Then check what is in the circled glass.

(2) Prompt:

Describe the pointed region in the image.

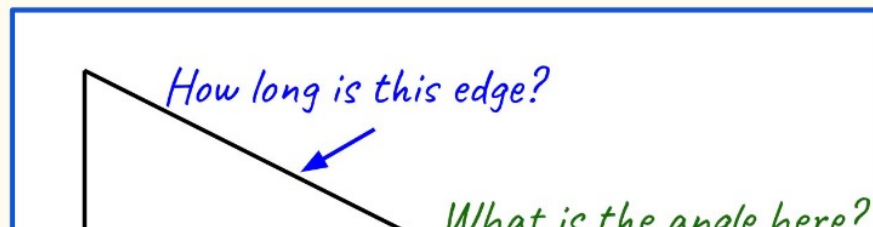
Method	Validataion set								Test set							
	in.		near.		out.		overall		in.		near.		out.		overall	
	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
OSCAR	85.4	11.9	84.0	11.7	80.3	10.0	83.4	11.4	84.8	12.1	82.1	11.5	73.8	9.7	80.9	11.3
Human	84.4	14.3	85.0	14.3	95.7	14.0	87.1	14.2	80.6	15.0	84.6	14.7	91.6	14.2	85.3	14.6
VIVO	92.2	12.9	87.8	12.6	87.5	11.5	88.3	12.4	89.0	12.9	87.8	12.6	80.1	11.1	86.6	12.4
VinVL	103.7	13.7	95.6	13.4	83.8	11.9	94.3	13.1	98.0	13.6	95.2	13.4	78.0	11.5	92.5	13.1
UFO	103.9	14.5	95.5	13.8	83.5	12.3	94.3	13.6	98.9	14.3	94.7	13.9	77.9	12.1	92.3	13.6
mPLUG	-	-	-	-	-	-	114.8	14.8	-	-	-	-	-	-	-	-
SimVLM	113.7	-	110.9	-	115.2	-	115.2	-	113.7	-	110.9	-	115.2	-	115.2	-
LEMON	118.0	15.4	116.3	15.1	120.2	14.5	117.3	15.0	112.8	15.2	115.5	15.1	110.1	13.7	114.3	14.9
UniversalCap	123.2	15.0	121.5	15.3	123.4	14.4	122.1	15.0	118.9	15.4	120.6	15.3	114.3	14.1	119.3	15.1
CoCa	-	-	-	-	-	-	122.4	15.5	-	-	-	-	-	-	120.6	15.5
GIT _B	100.7	13.8	97.7	13.5	89.6	12.5	96.6	13.4	-	-	-	-	-	-	-	-
GIT _L	107.7	14.9	107.8	14.5	102.5	13.7	106.9	14.4	-	-	-	-	-	-	-	-
GIT	129.8	16.3	124.1	16.0	127.1	15.7	125.5	16.0	122.4	16.2	123.9	16.0	122.0	15.7	123.4	15.9
GIT2	126.9	16.1	125.8	16.2	130.6	15.8	126.9	16.1	124.2	16.4	125.5	16.1	122.3	15.6	124.8	16.1

GPT-4V:

The highlighted red region in the image is a table cell that contains a numerical value of 122.3. It is located in the "Test set" section of the table, under the "out" column and the "GIT2" row. This value represents the overall performance metric for the GIT2 method on the test set, in the "out" category.

(4) Prompt:

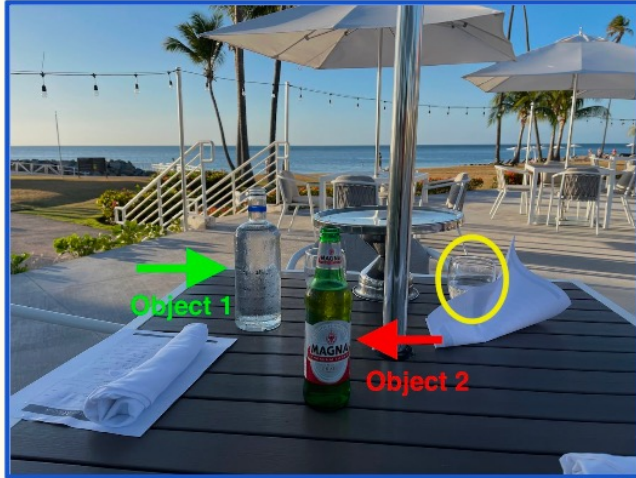
Answer the question in the following figure. Provide intermediate steps.



poles and is located under a patio area with tables and umbrellas. The lights appear to be decorative and are likely used to provide ambient lighting in the evening.

(3) Prompt:

What is in the circled glass? Object 1 or 2? Describe what is Object 1 and object 2. Then check what is in the circled glass.



GPT-4V:

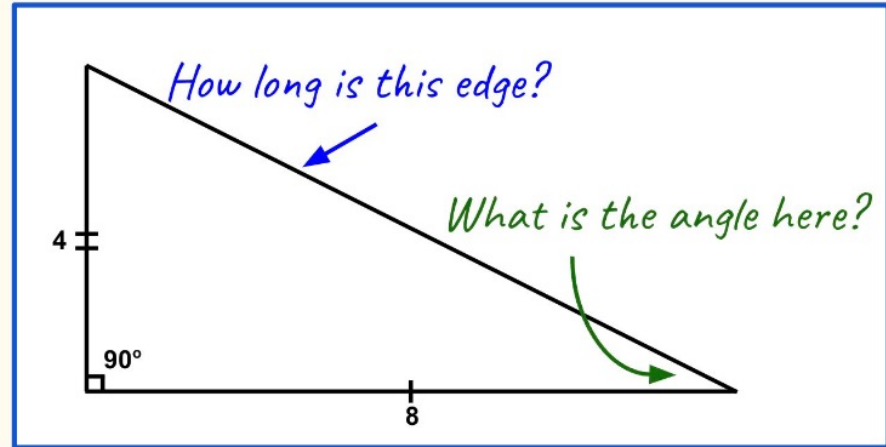
Object 1 is a glass bottle of water. It is clear and has a blue label on it.

Object 2 is a glass bottle of Magna beer. It is green and has a red and white label on it.

In the circled glass, it appears to be a clear liquid, likely water. So, it is likely that the contents of the circled glass are from Object 1, the glass bottle of water.

(4) Prompt:

Answer the question in the following figure. Provide intermediate steps.



GPT-4V:

To find the length of the blue arrowed edge, we can use the Pythagorean theorem since we have a right triangle. The theorem states that $a^2 + b^2 = c^2$, where a and b are the two legs of the right triangle and c is the hypotenuse. Plugging in the values we have, we get:

$$4^2 + 8^2 = c^2 \quad 16 + 64 = c^2 \quad 80 = c^2 \quad \sqrt{80} = c \quad 8.94 \approx c$$

Therefore, the length of the blue arrowed edge is approximately 8.94.

To find the angle at the green arrow, we can use the tangent function since we have the opposite and adjacent sides of the angle. The tangent function states that $\tan\theta =$

opposite/adjacent. Plugging in the values we have, we get:

$$\tan\theta = 4/8 \quad \tan\theta = 1/2 \quad \theta = \tan^{-1}(1/2) \quad \theta \approx 26.57^\circ$$

Therefore, the angle at the green arrow is approximately 26.57° .

Medical Image Understanding

Prompt:

What's wrong?



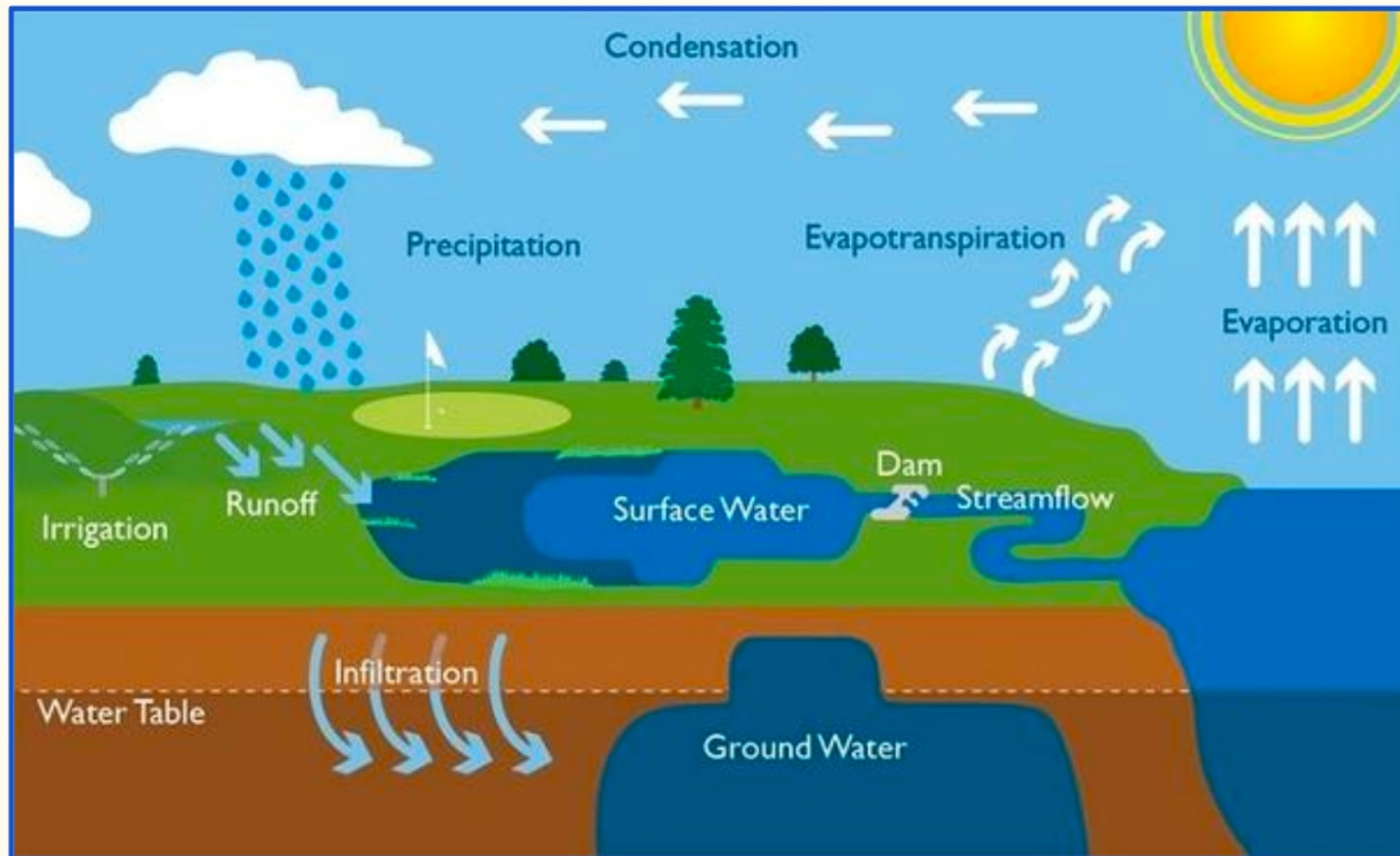
GPT-4V:

There appears to be a fracture or break in the fifth metatarsal bone (the bone on the outside of the foot that connects to the pinky toe). This is commonly known as a "Jones fracture."

Science and Knowledge

Prompt:

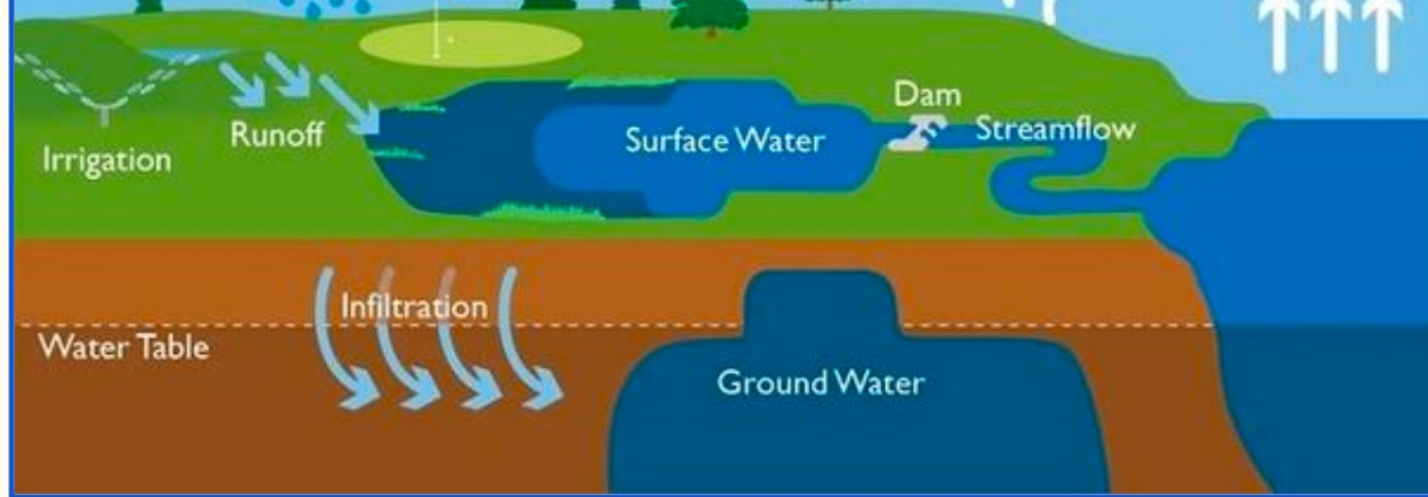
Suppose you are a teacher, please use the figure to explain the distinction between evaporation and evapotranspiration.



GPT-4V:

Okay class, today we are going to learn about the distinction between evaporation and evapotranspiration. Let's take a look at this figure.





GPT-4V:

Okay class, today we are going to learn about the distinction between evaporation and evapotranspiration. Let's take a look at this figure.

First, let's understand what evaporation is. Evaporation is the process by which water is changed from its liquid state to its gaseous state. This usually takes place at the surface of a liquid, like the ocean or a lake, when the water molecules gain enough energy to break away and form water vapor. So in this figure, we can see that the water in the lake and the dam is evaporating and becoming water vapor.

Now, let's take a look at evapotranspiration. This is a combination of two processes - evaporation and transpiration. Transpiration is the process by which plants release water vapor through their leaves. So, in this figure, we can see that the trees and plants are releasing water vapor into the air, which is the transpiration part of evapotranspiration. At the same time, the water in the soil and the surface is also evaporating. This combination of evaporation and transpiration is what we call evapotranspiration.

In summary, evaporation is the process by which water is changed from its liquid state to its gaseous state, while evapotranspiration is the combination of evaporation and transpiration. Both these processes contribute to the water cycle and are important for maintaining the balance of water in our environment.