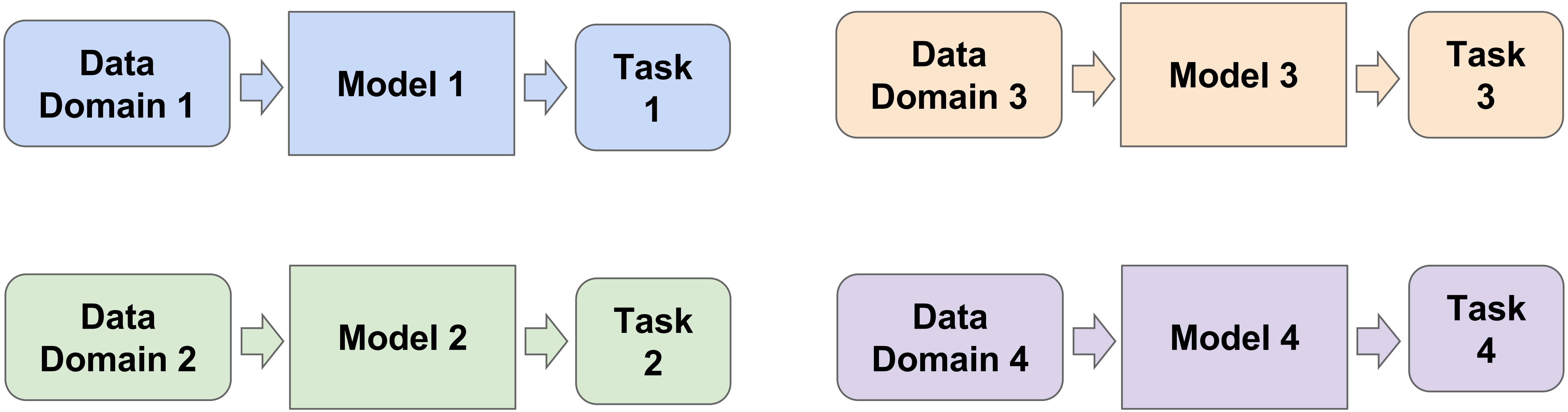
Lecture 16:
Multi-Modal Foundation Models

How have we been thinking about models in this class so far?

Train a *specialized* model for *each* task



Data Domain 1 → Model 1 → Task 1

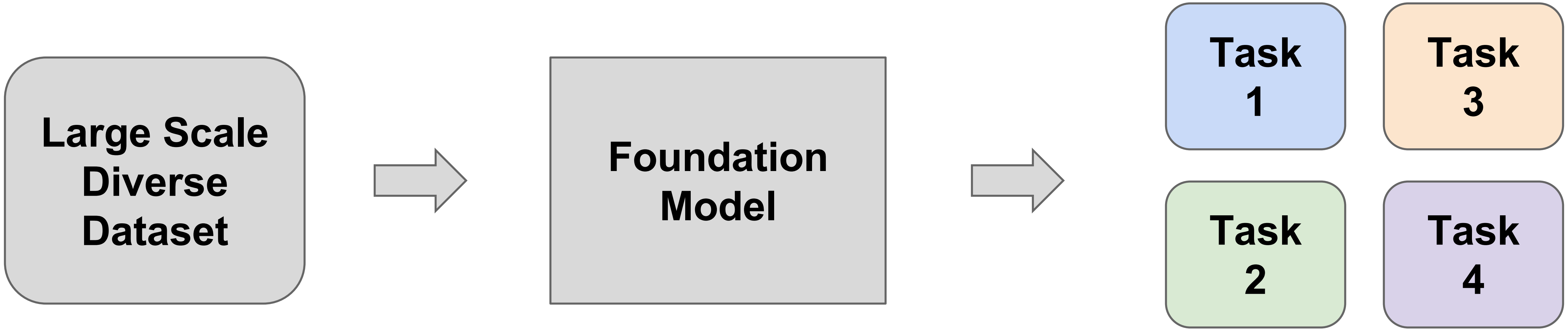
Data Domain 3 → Model 3 → Task 3

Data Domain 2 → Model 2 → Task 2

Data Domain 4 → Model 4 → Task 4

Now, we build Foundation Models

Pre-train one model that acts as the *foundation* for many different tasks



Large Scale Diverse Dataset

Foundation Model

Task 1

Task 3

Task 2

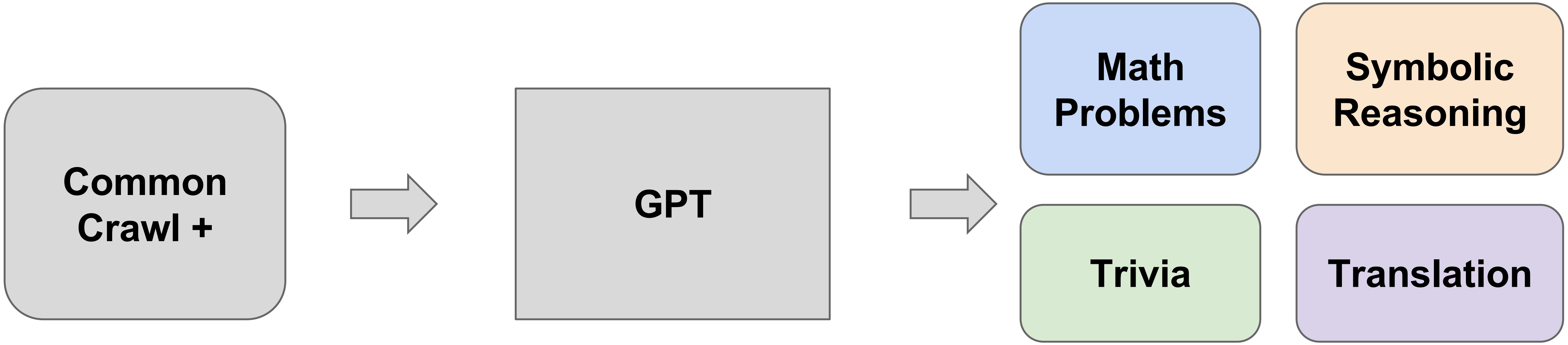
Task 4

Pre-training

Fine-tuning / zero-shot / few-shot

Foundation Models

Language



Common Crawl +

GPT

Math Problems

Symbolic Reasoning

Trivia

Translation

Pre-training

Fine-tuning / zero-shot / few-shot

There are many classes of Foundation Models

Language

ELMo
BERT
GPT
T5

Classification

CLIP
CoCa

LM + Vision

LLaVA
Flamingo
GPT-4V
Gemini
Molmo

And More!

Segment Anything
Whisper
Dalle
Stable Diffusion
Imagen

Chaining

LMs + CLIP
Visual Programming

How do identify a model as a Foundation?

Always see with foundation models:

- general /robust to many different tasks

Often see with foundation models:

- Large # params
- Large amount of data
- Self-supervised pre-training objective

Language models are out of scope for this class

Language	Classification	LM + Vision	And More!	Chaining
ELMo	CLIP	LLaVA	Segment Anything	LMs + CLIP
BERT	CoCa	Flamingo	Whisper	Visual Programming
GPT		GPT-4V	Dalle	
T5		Gemini	Stable Diffusion	
		Molmo	Imagen	

We will focus on multimodal (vision) foundation models

Language

ELMo
BERT
GPT
T5

Classification

CLIP
CoCa

LM + Vision

LLaVA
Flamingo
GPT-4V
Gemini
Molmo

And More!

Segment Anything
Whisper
Dalle
Stable Diffusion
Imagen

Chaining

LMs + CLIP
Visual Programming

Let’s start with the foundation models for classification

Language

ELMo
BERT
GPT
T5

Classification

CLIP
CoCa

LM + Vision

LLaVA
Flamingo
GPT-4V
Gemini
Molmo

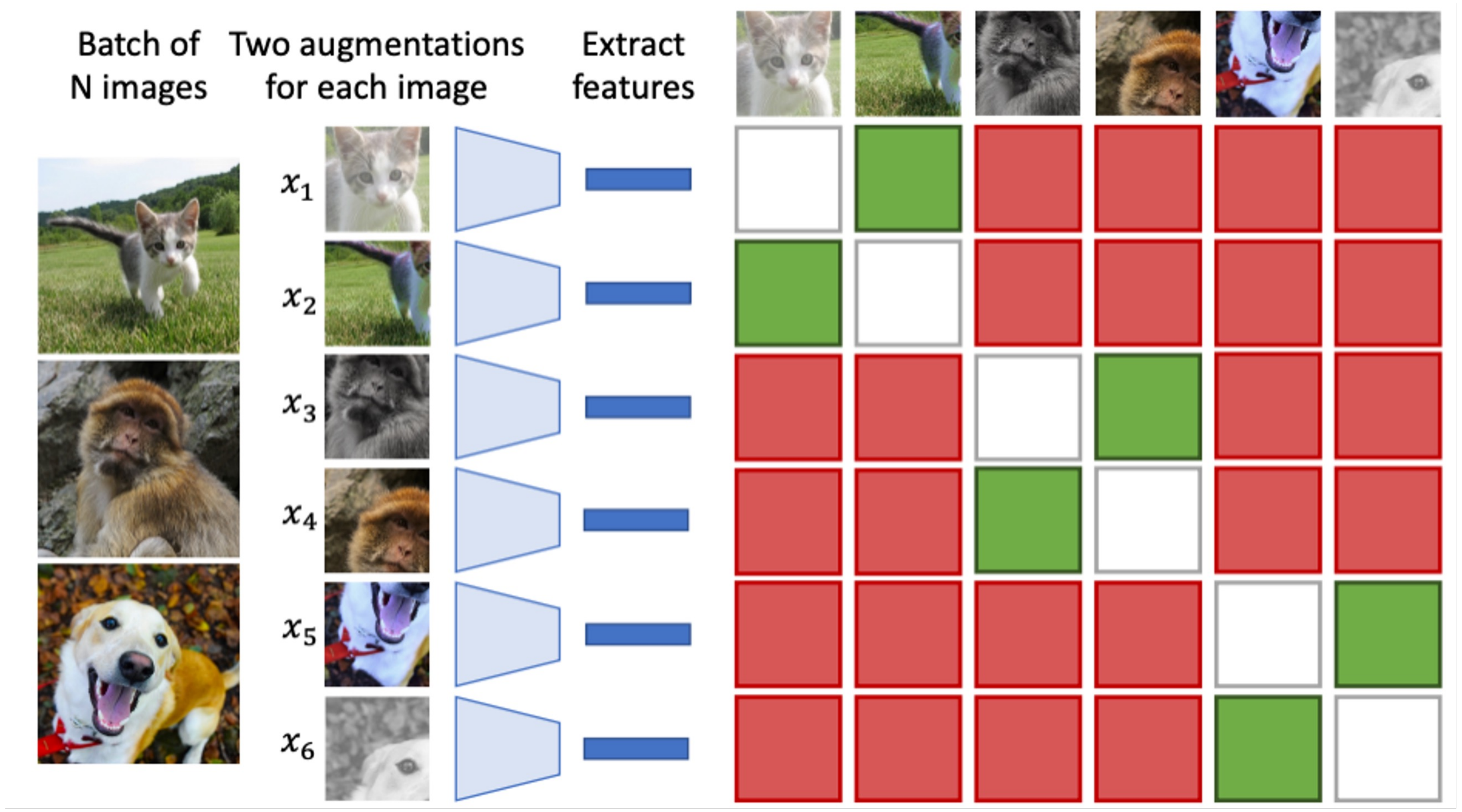
And More!

Segment Anything
Whisper
Dalle
Stable Diffusion
Imagen

Chaining

LMs + CLIP
Visual Programming

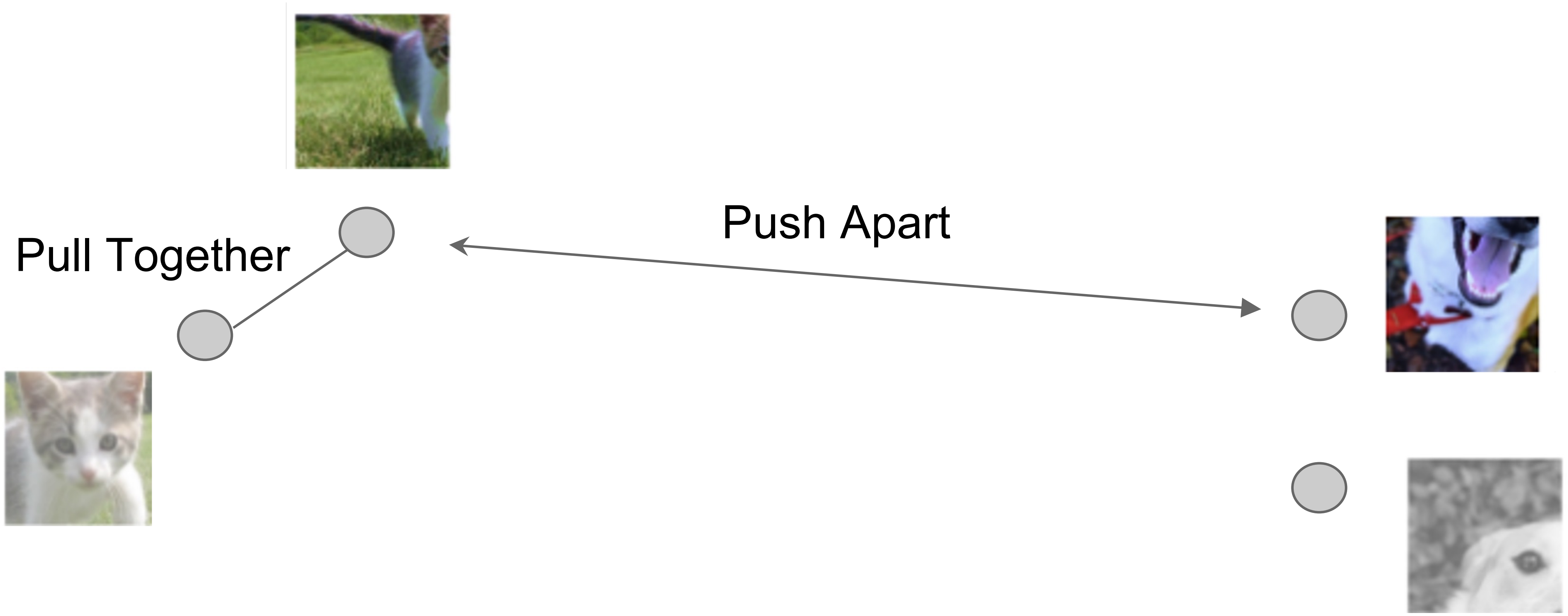
Recall this self-supervised objective from SimCLR



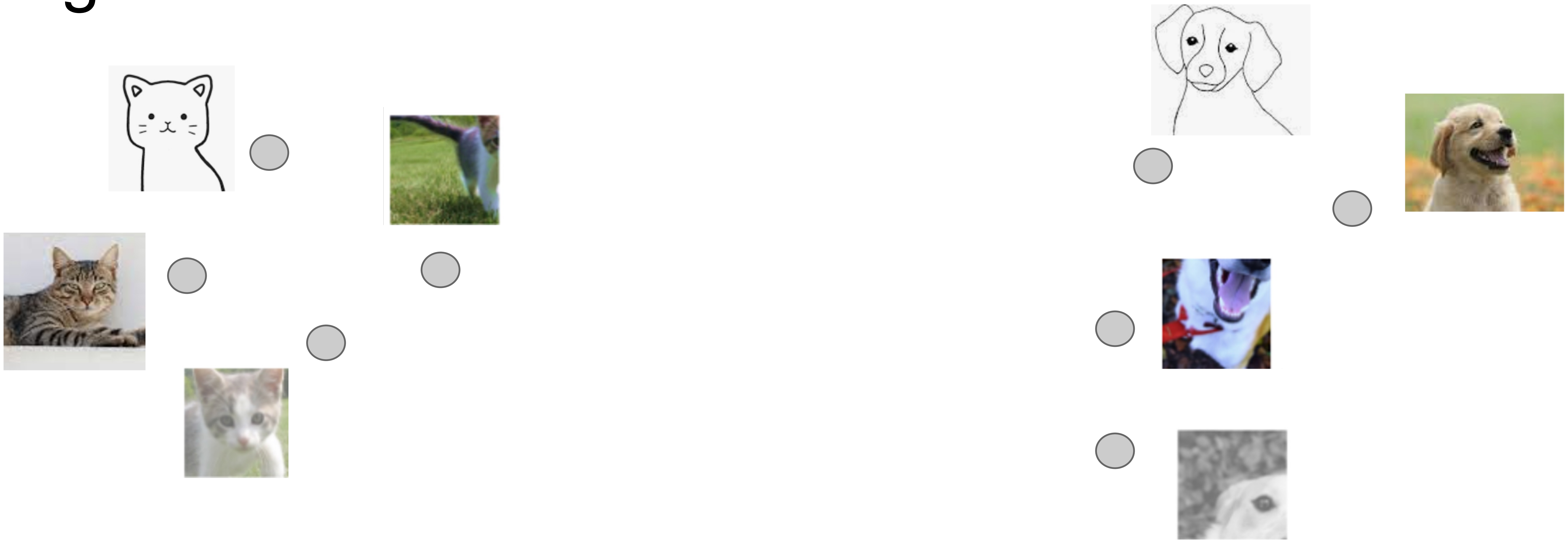
Use Self Supervised learning to learn good image features

Can train small classifiers on top of these features using supervised learning

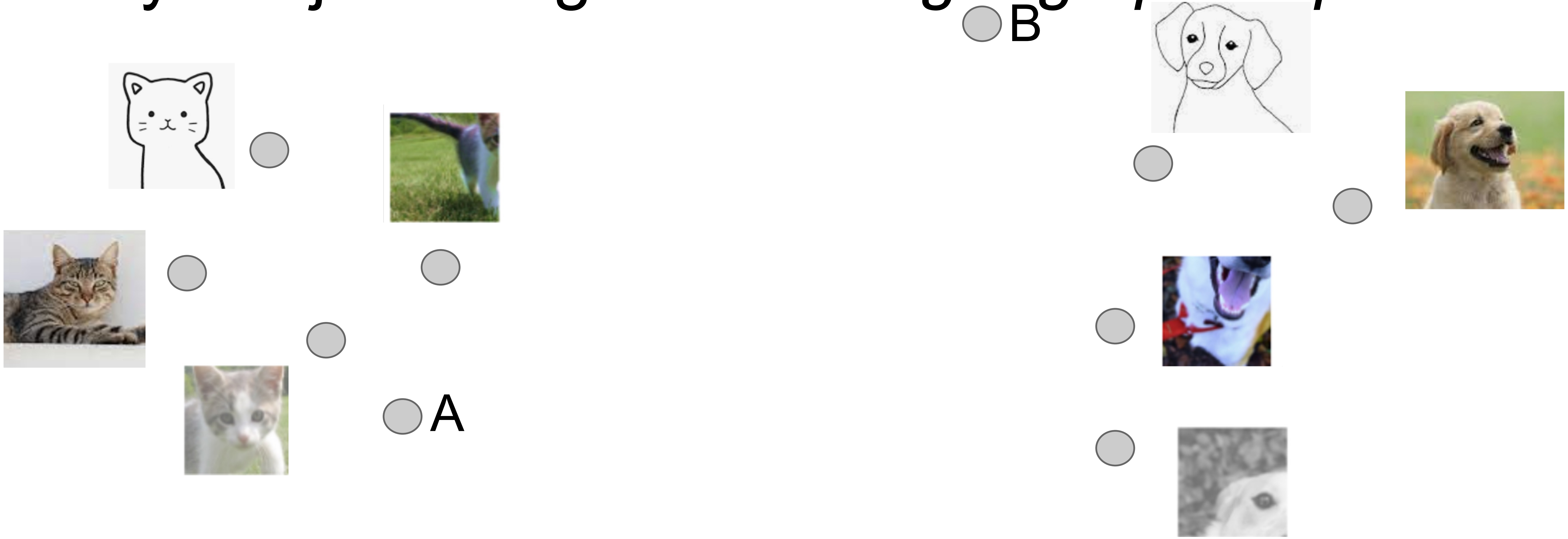
The main idea was to learning concepts without labels -> a self-supervised pretraining objective



The hope was that the learned representations generalize to new instances

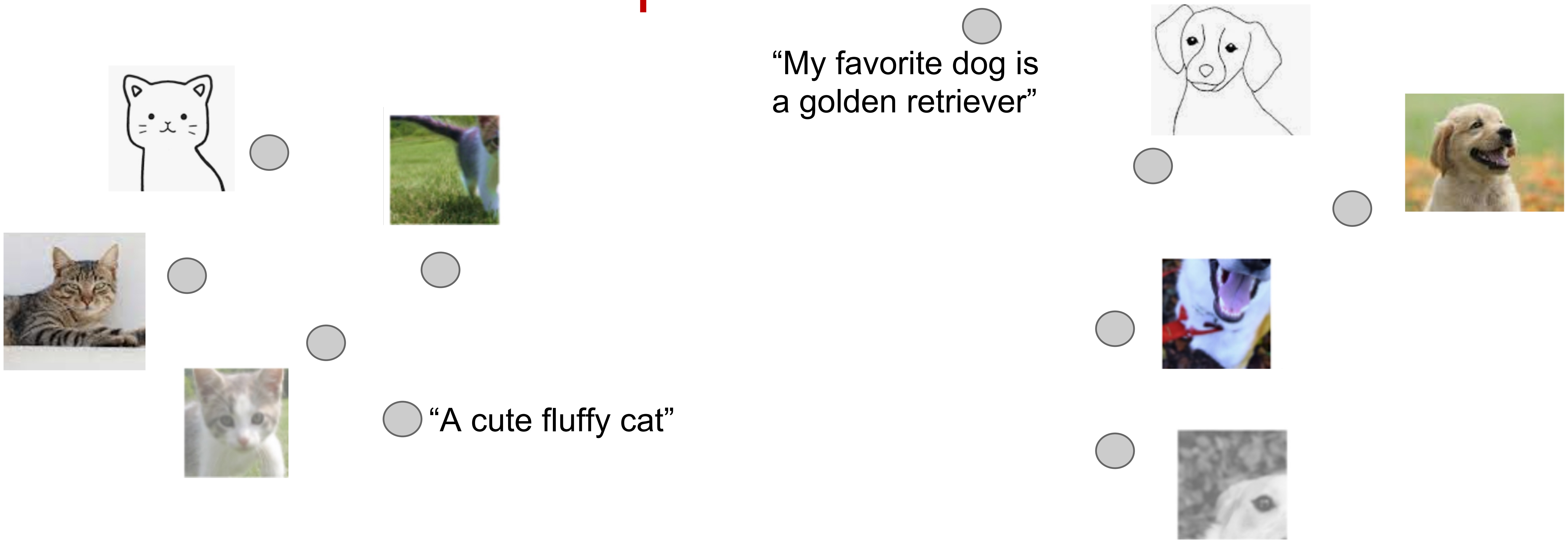


Can we generalize these representations beyond just images? *To language perhaps?*



1. “A cute fluffy cat”
2. “My favorite dog is a golden retriever”

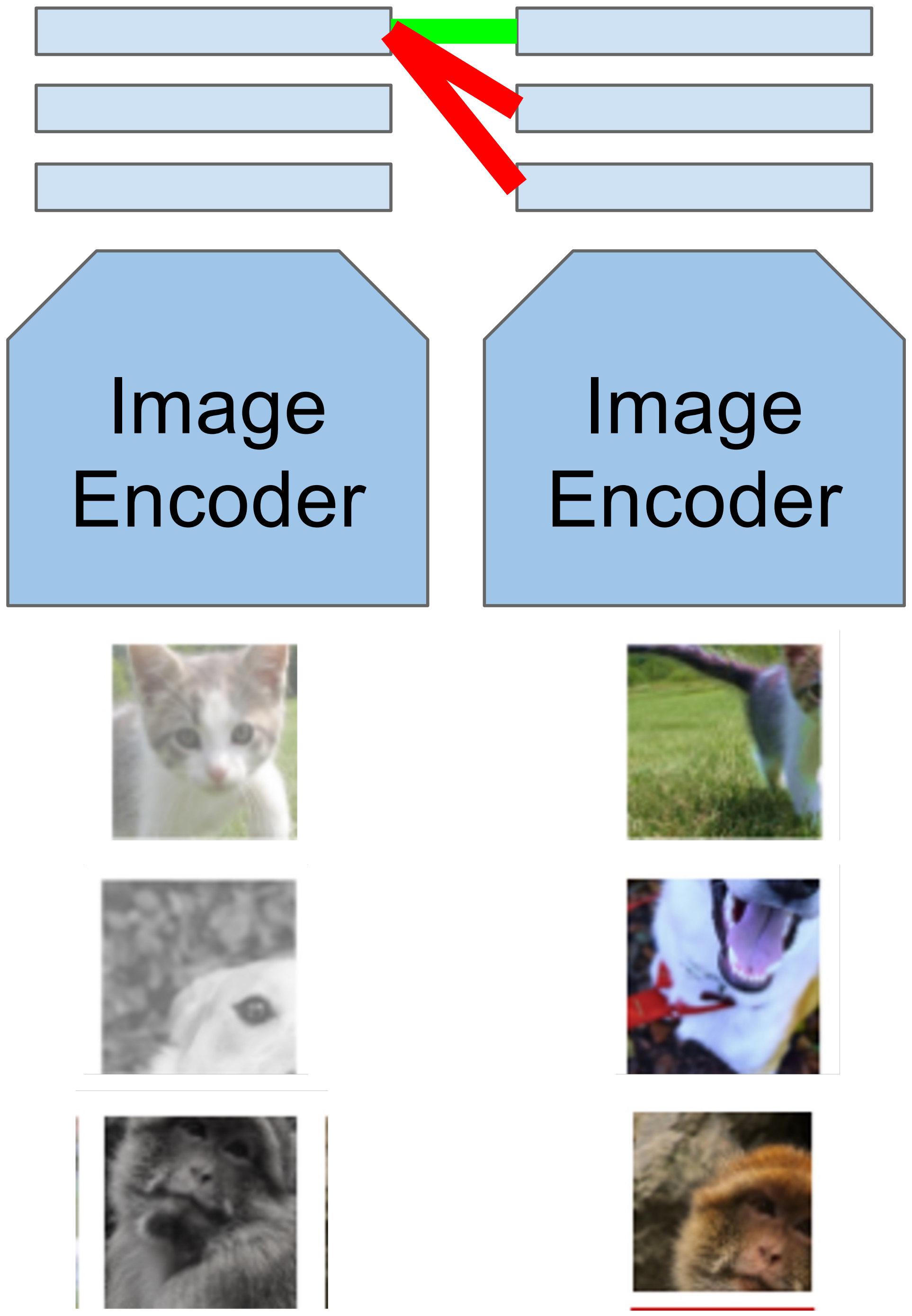
What if this representation space could also embed sentences/phrases?



“My favorite dog is a golden retriever”

“A cute fluffy cat”

SimClr



SimClr

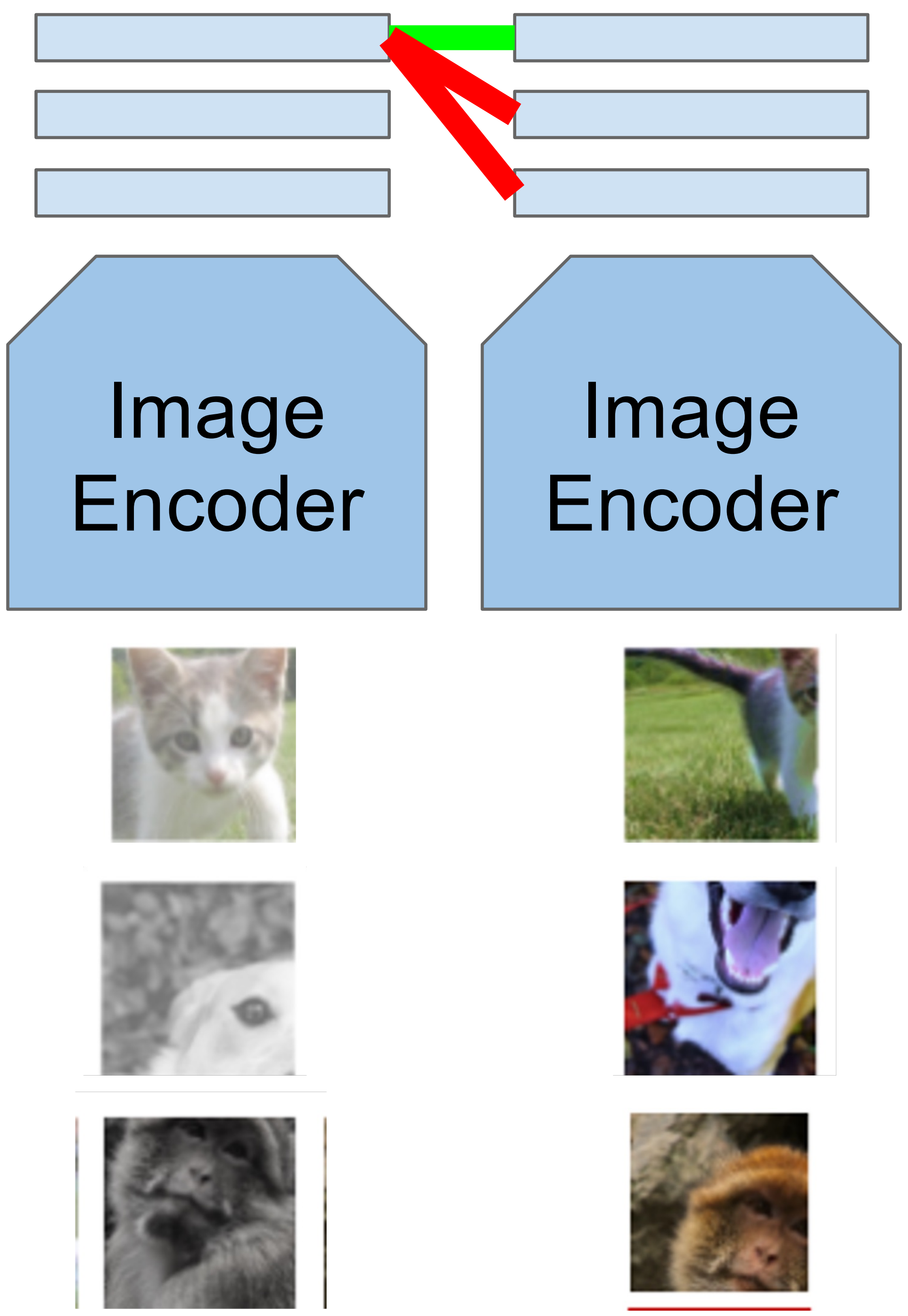


Image Encoder

Image Encoder

CLIP

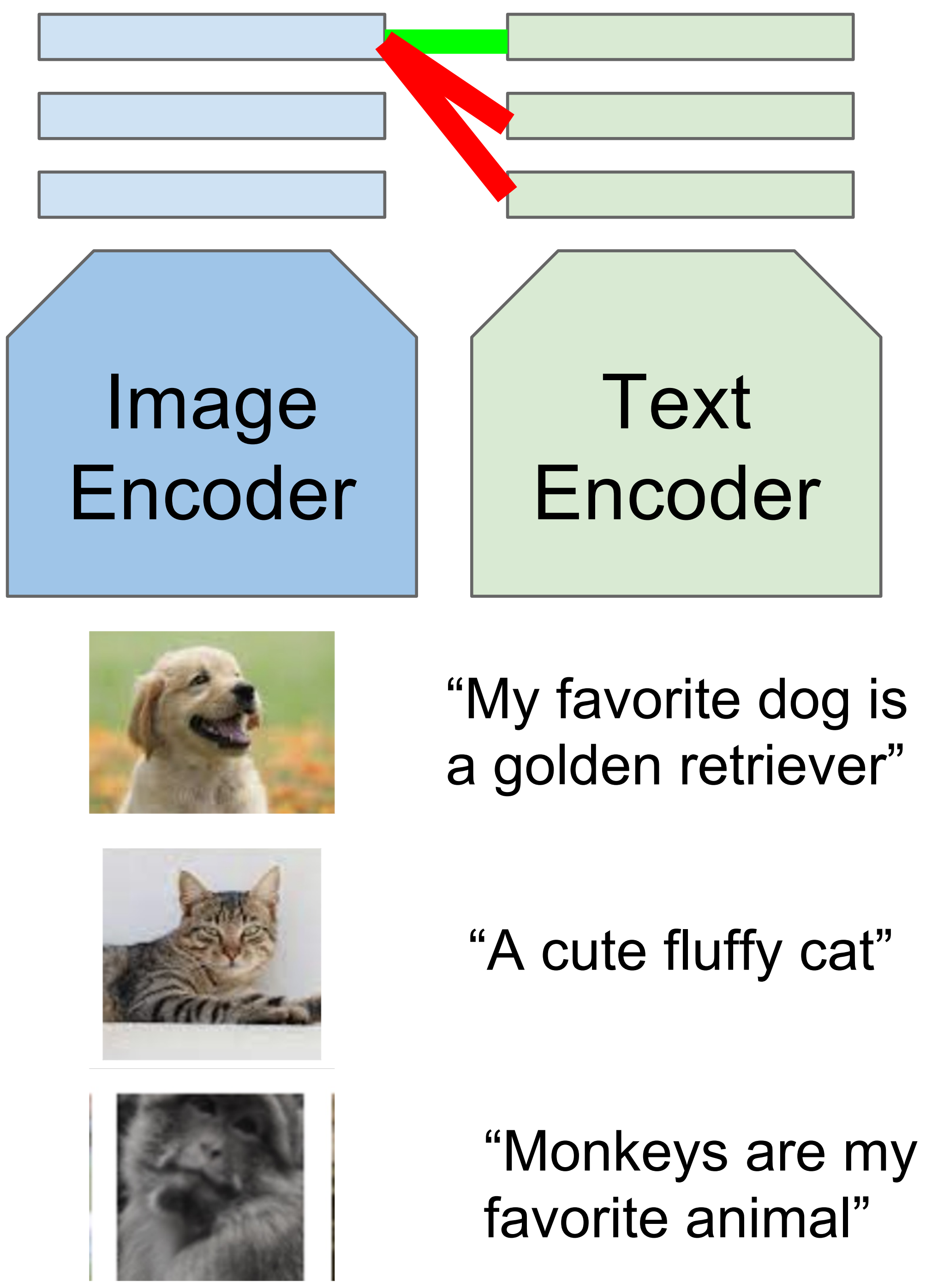


Image Encoder

Text Encoder

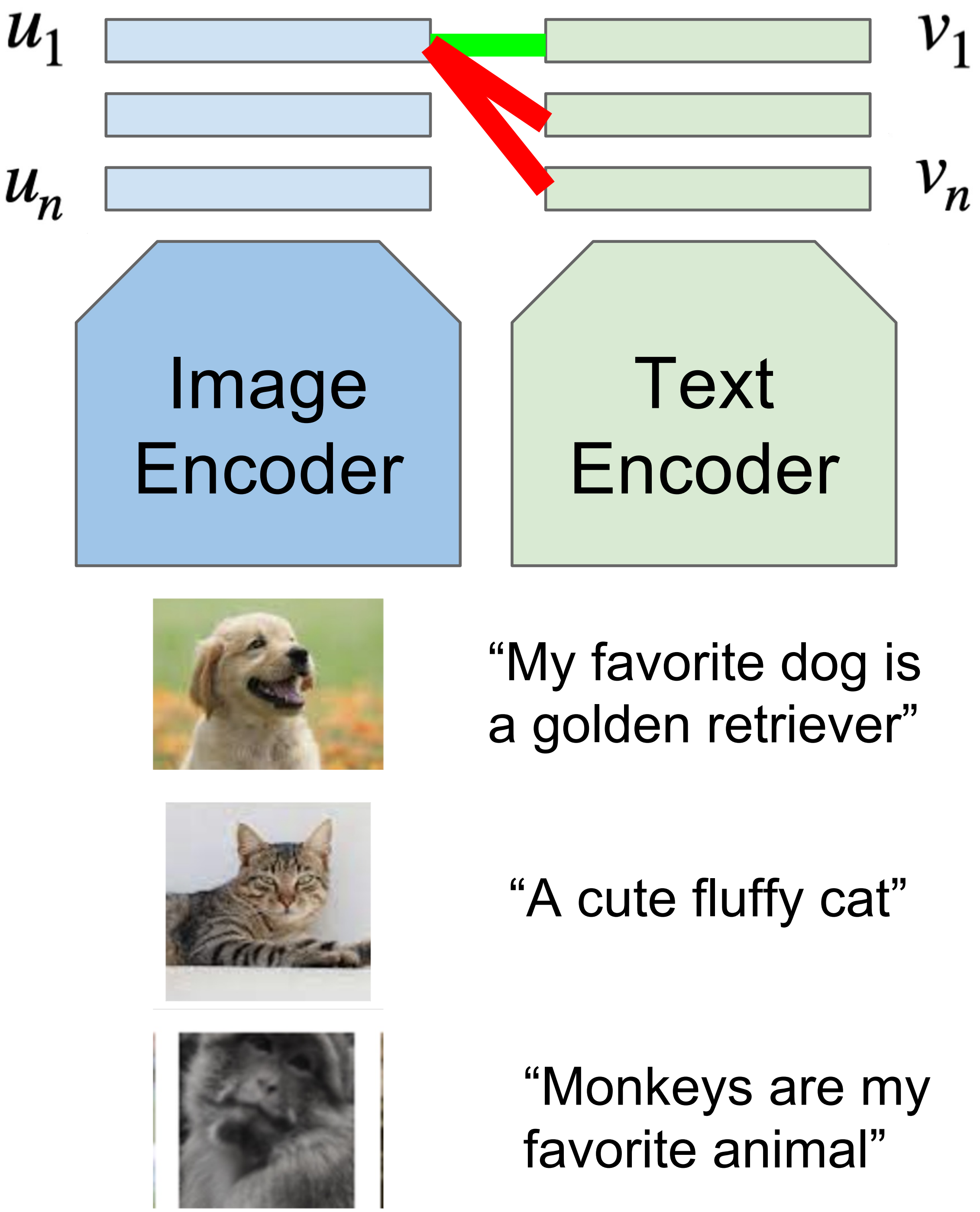
“My favorite dog is a golden retriever”

“A cute fluffy cat”

“Monkeys are my favorite animal”

CLIP is trained with the same contrastive objective

$$\sum_{i=1}^{n} -\log\left(\frac{e^{\langle u_i,v_i\rangle}}{\sum_{j=1}^{n} e^{\langle u_i,v_j\rangle}}\right)$$



“My favorite dog is a golden retriever”

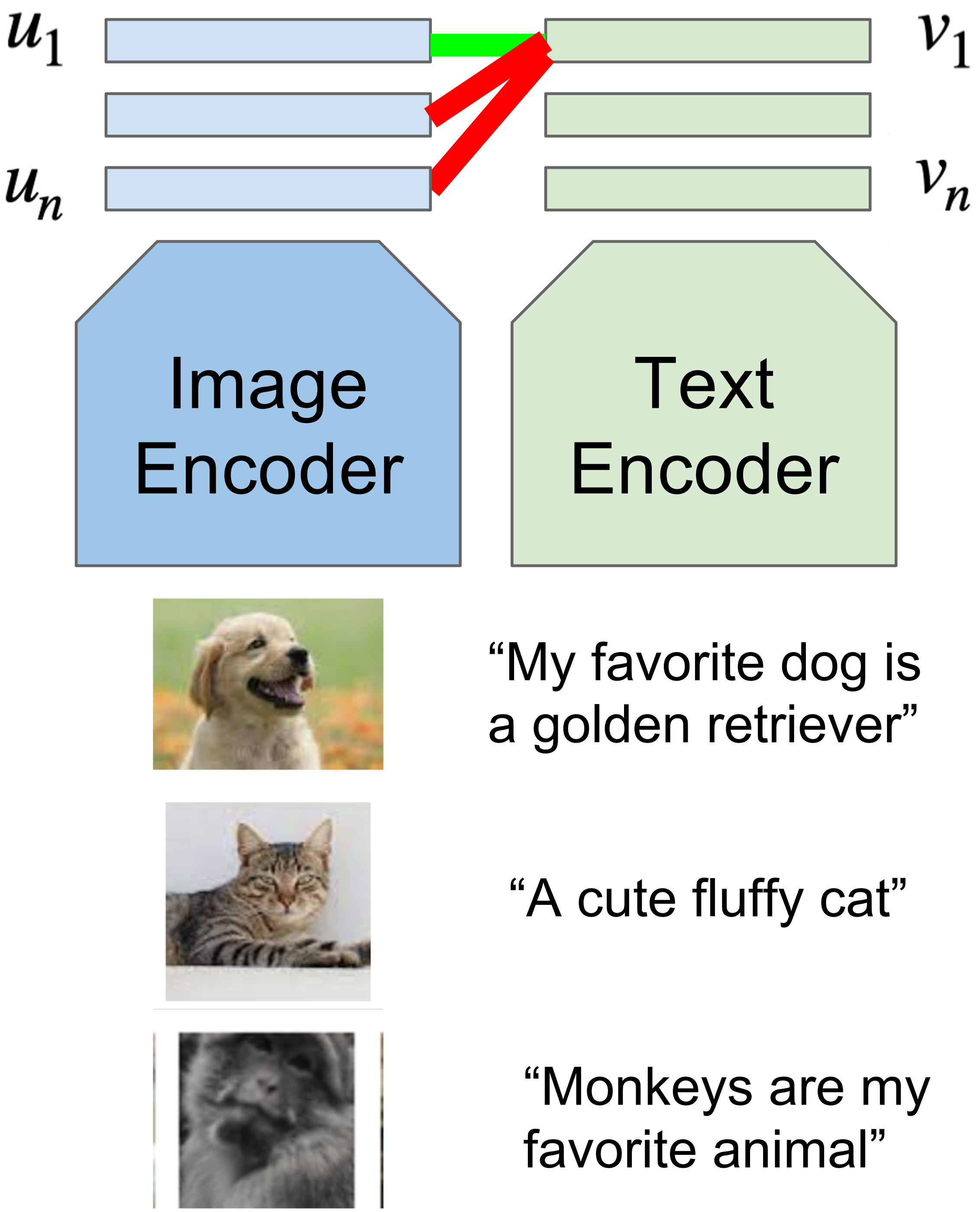
“A cute fluffy cat”

“Monkeys are my favorite animal”

CLIP Training Objective

$$\sum_{i=1}^{n} -\log\left(\frac{e^{\langle u_i, v_i \rangle}}{\sum_{j=1}^{n} e^{\langle u_i, v_j \rangle}}\right)$$

$$+\sum_{i=1}^{n} -\log\left(\frac{e^{\langle u_i, v_i \rangle}}{\sum_{j=1}^{n} e^{\langle u_j, v_i \rangle}}\right)$$



"My favorite dog is a golden retriever"

"A cute fluffy cat"

"Monkeys are my favorite animal"

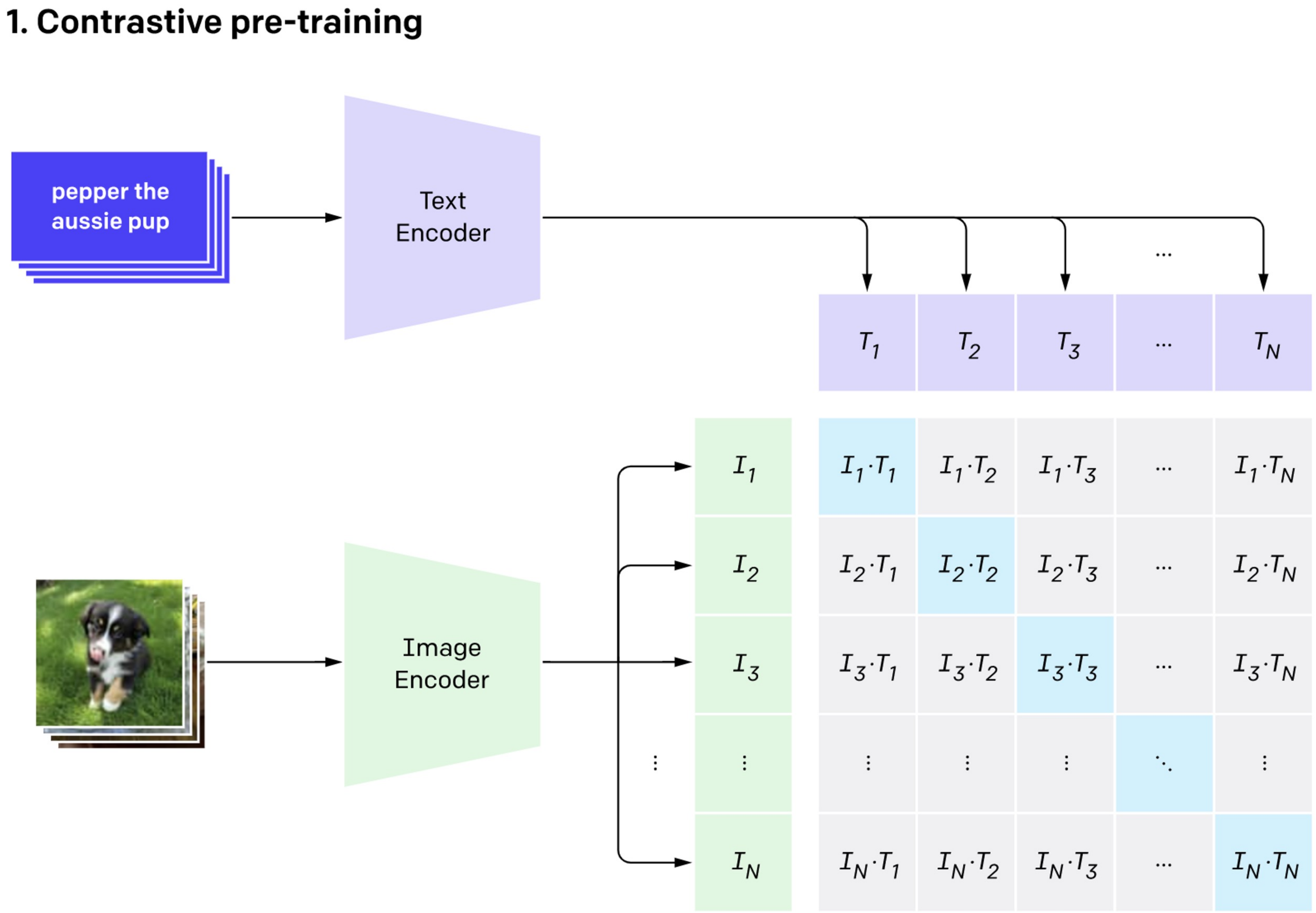
Lots of image-text data can be found online

Mount Rainier's northwestern slope viewed aerially just before sunset on September 6, 2020

CLIP training data was scraped at scale from images and their associated alt-text from the internet

https://en.wikipedia.org/wiki/Mount_Rainier

CLIP Training Objective

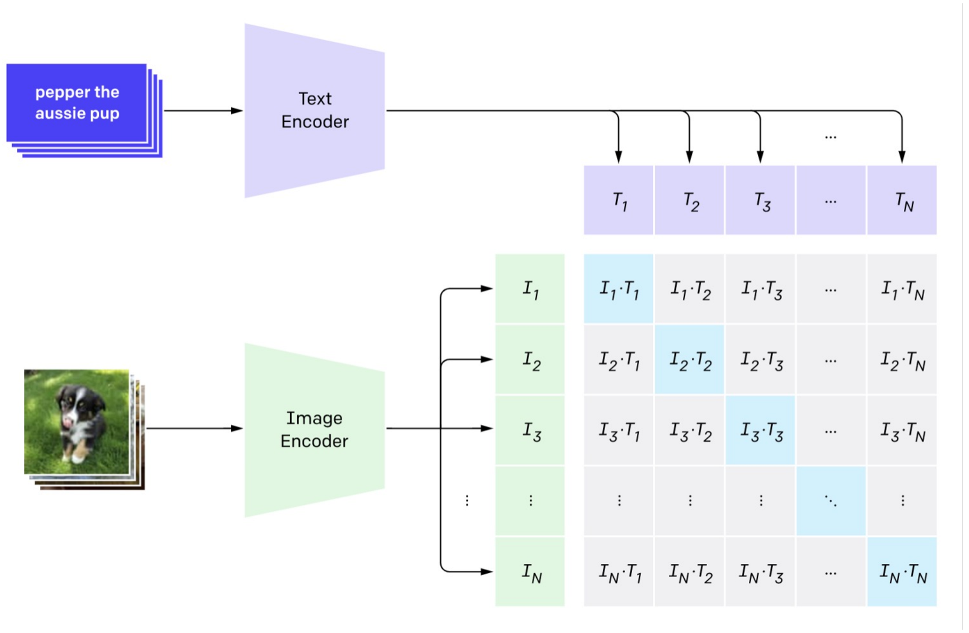


1. Contrastive pre-training
pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ⋮ I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
⋮ ⋮ ⋮ ⋱ ⋮
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$

At the end of training, you have a model that will give you a similarity score between an image and a text

Using pre-trained models out of the box

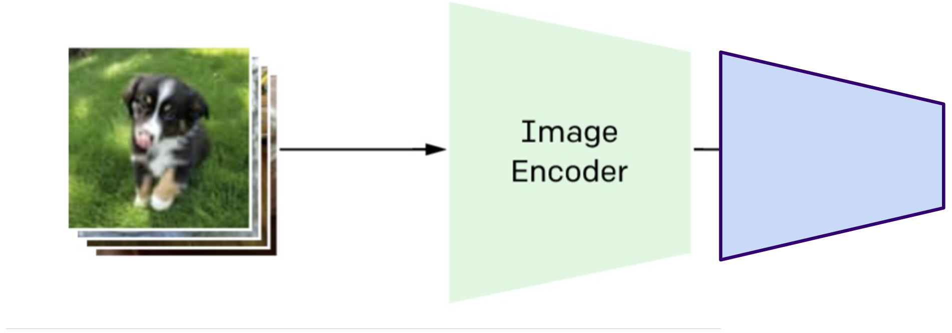
Step 1: Pretrain a network on a pretext task that doesn't require supervision



pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ⋮ I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
⋮ ⋮ ⋮ ⋱ ⋮
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$

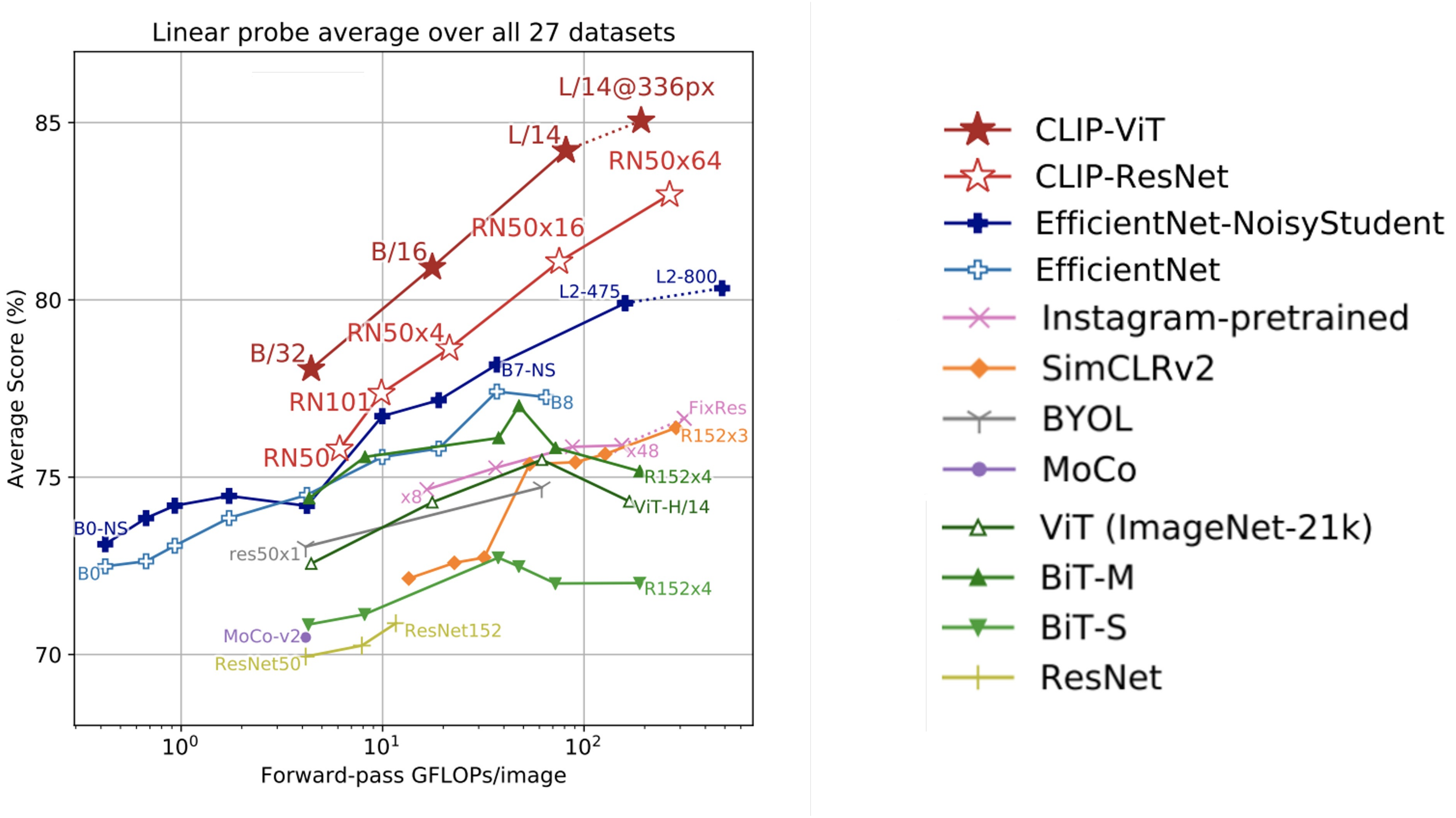
Pre-training tasks:
Contrastive Objective

Step 2: Transfer encoder to downstream tasks via linear classifiers



Downstream tasks:
Image classification, object detection, semantic segmentation

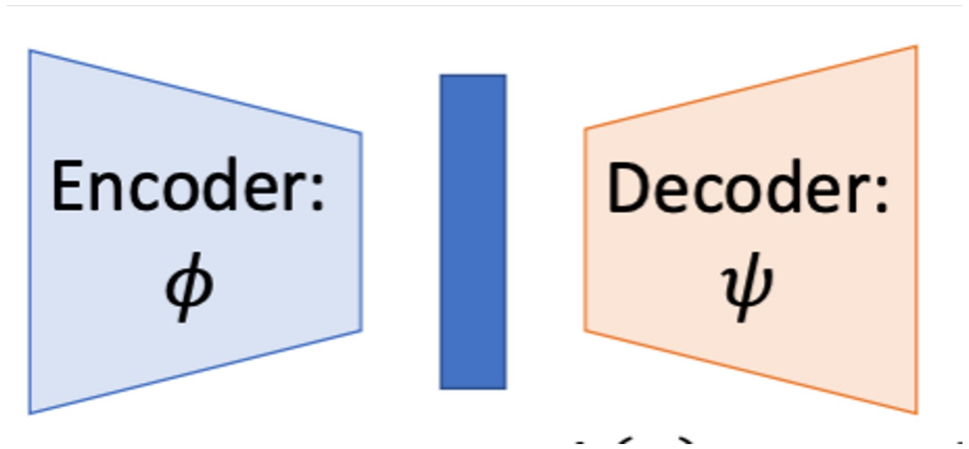
CLIP features w/ linear probe across multiple datasets



Big difference with language models: We can use LLMs zero-shot for new downstream tasks

Step 1: Pretrain a network on a pretext task that doesn't require supervision

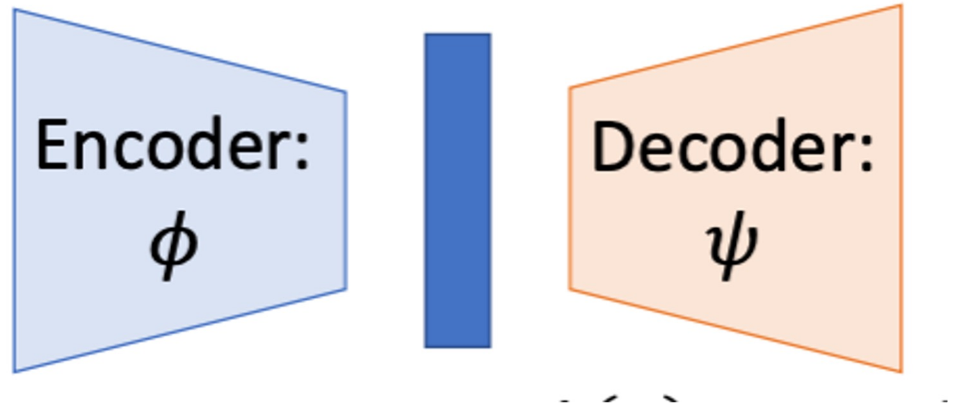
"I love ___"



"cake"

Step 2: Use the model out of the box in a creative way!

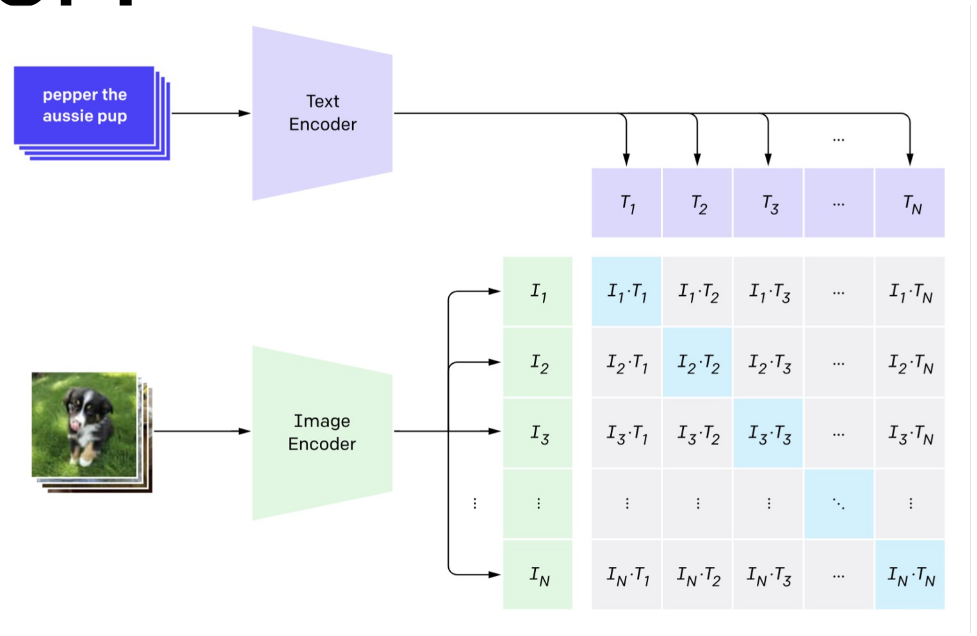
"The movie review 'I hated the movie' is ___"



"negative"

But how do we use pre-trained vision-language models in a zero-shot manner?

Step 1: <u>Pretrain</u> a network on a <u>pretext task</u> that doesn’t require supervision



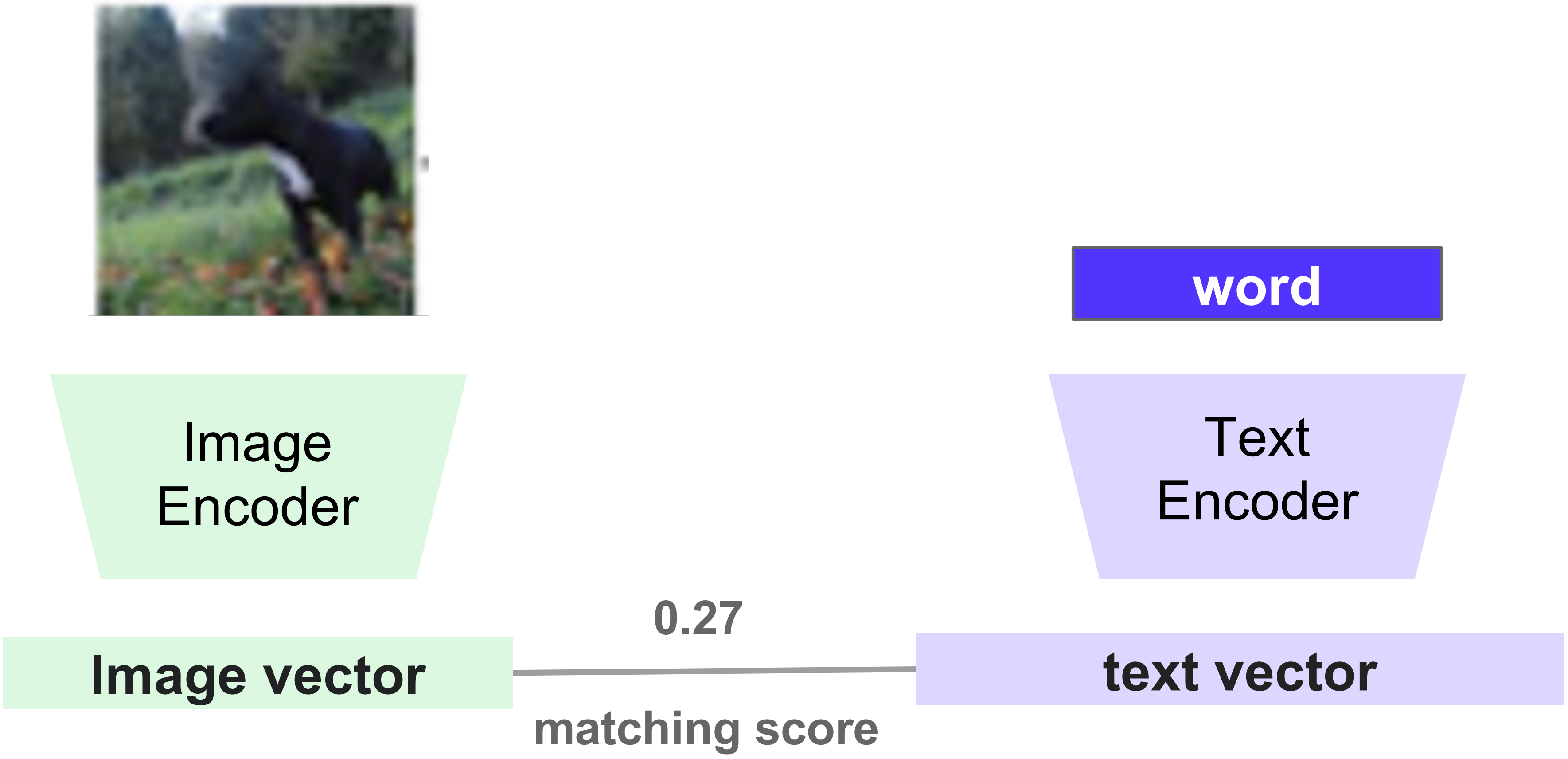
	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

Pre-training tasks:
Contrastive Objective

Step 2: Use the model out of the box in a creative way!

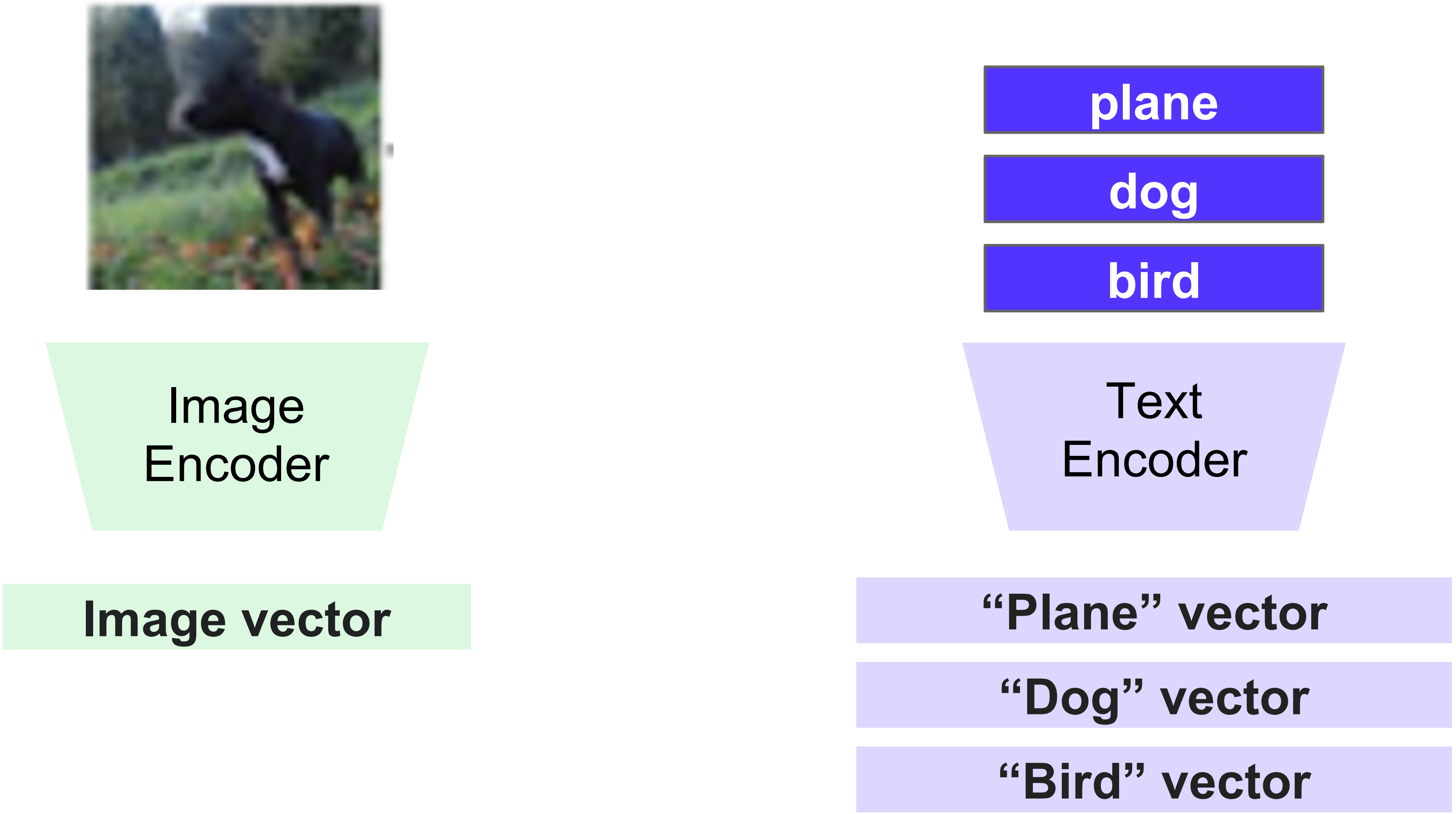
Out of the box classification (No fine-tuning)

Clever trick: we can create a classifier using the text encoder!



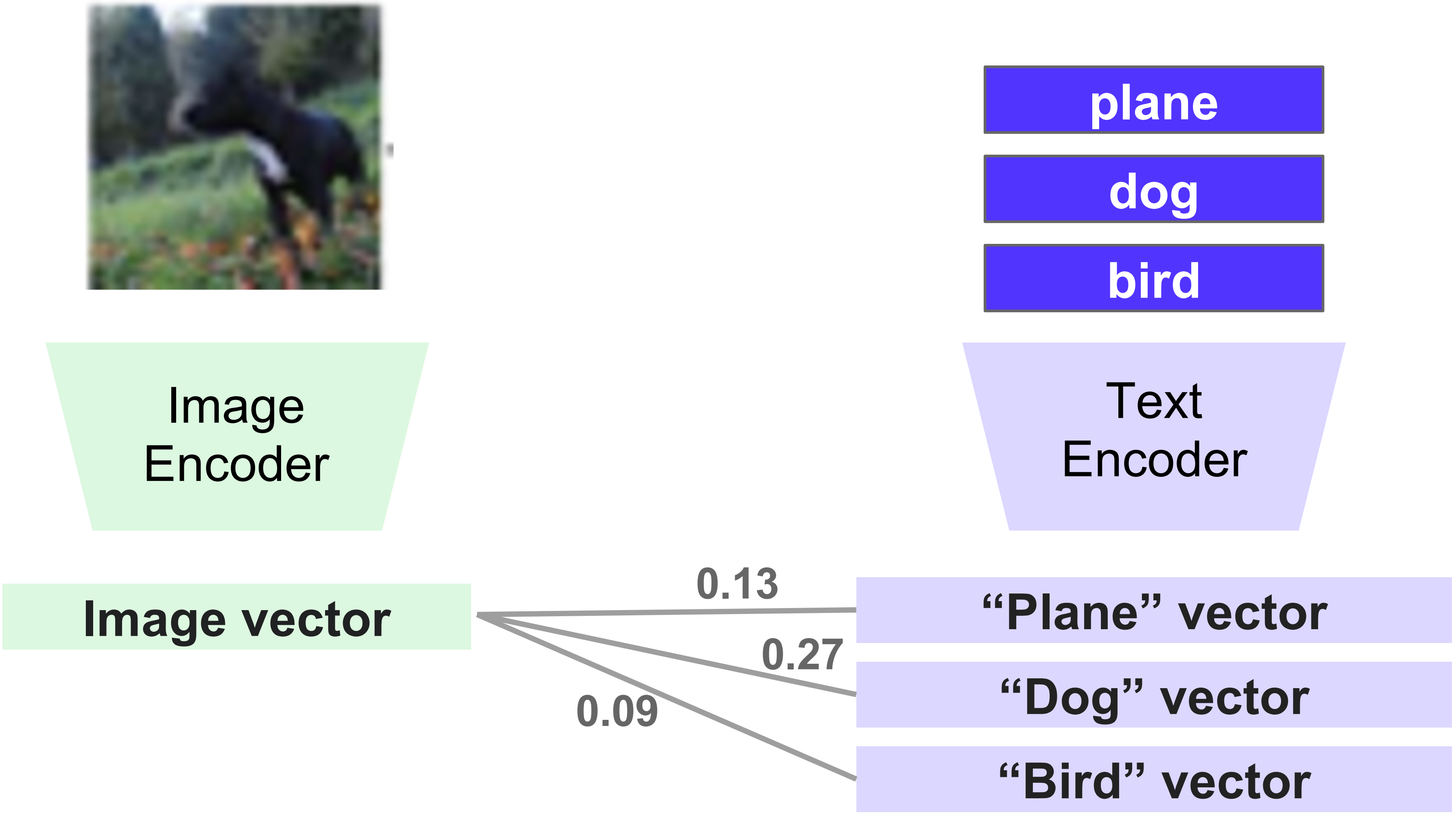
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Create a vector representation for *each* category!



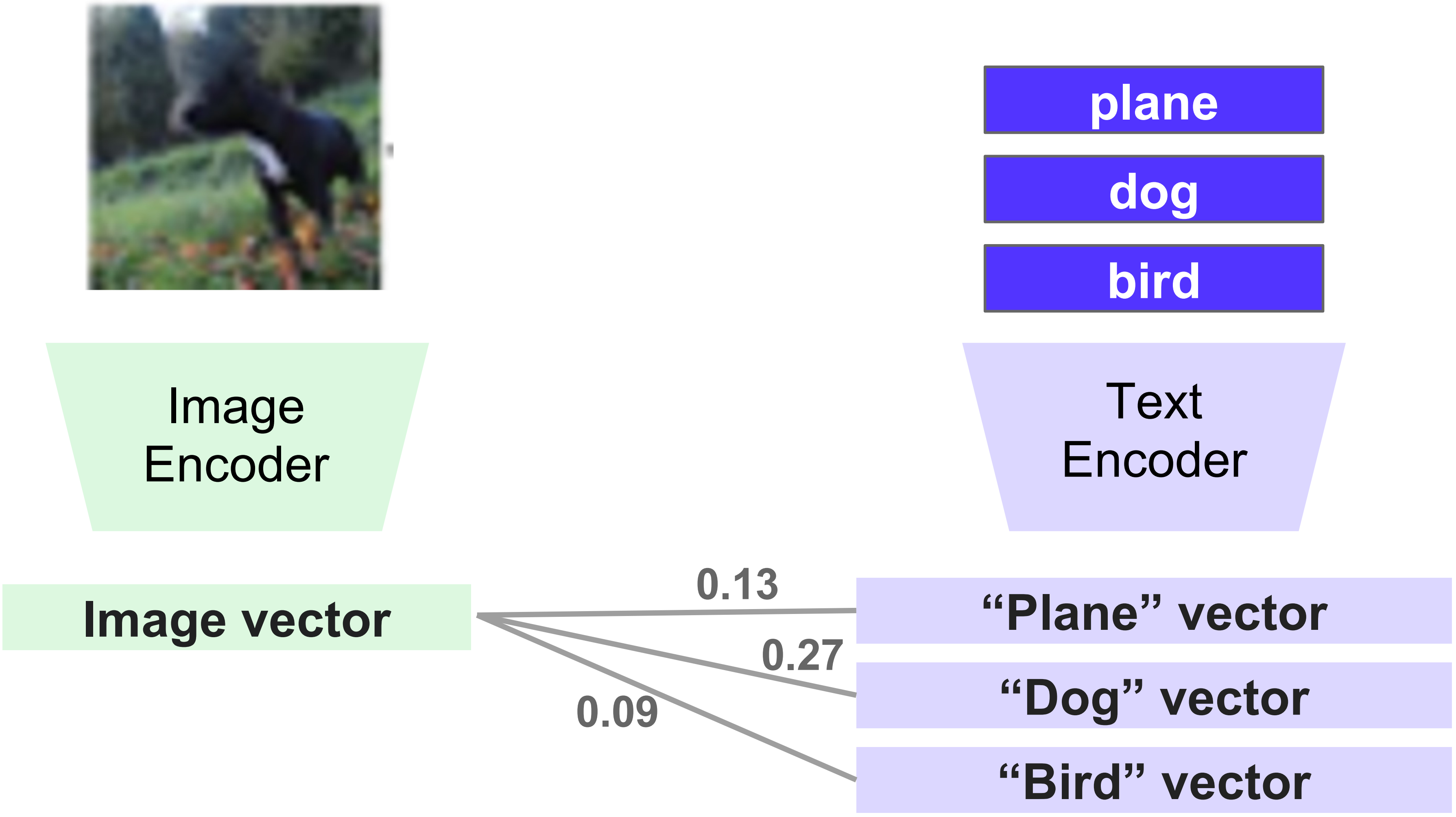
Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

Match a new image to the most similar vector



Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

You can think of this as a 1-NN algorithm with the vectors as the training data



plane

dog

bird

Image Encoder

Text Encoder

Image vector

0.13

0.27

0.09

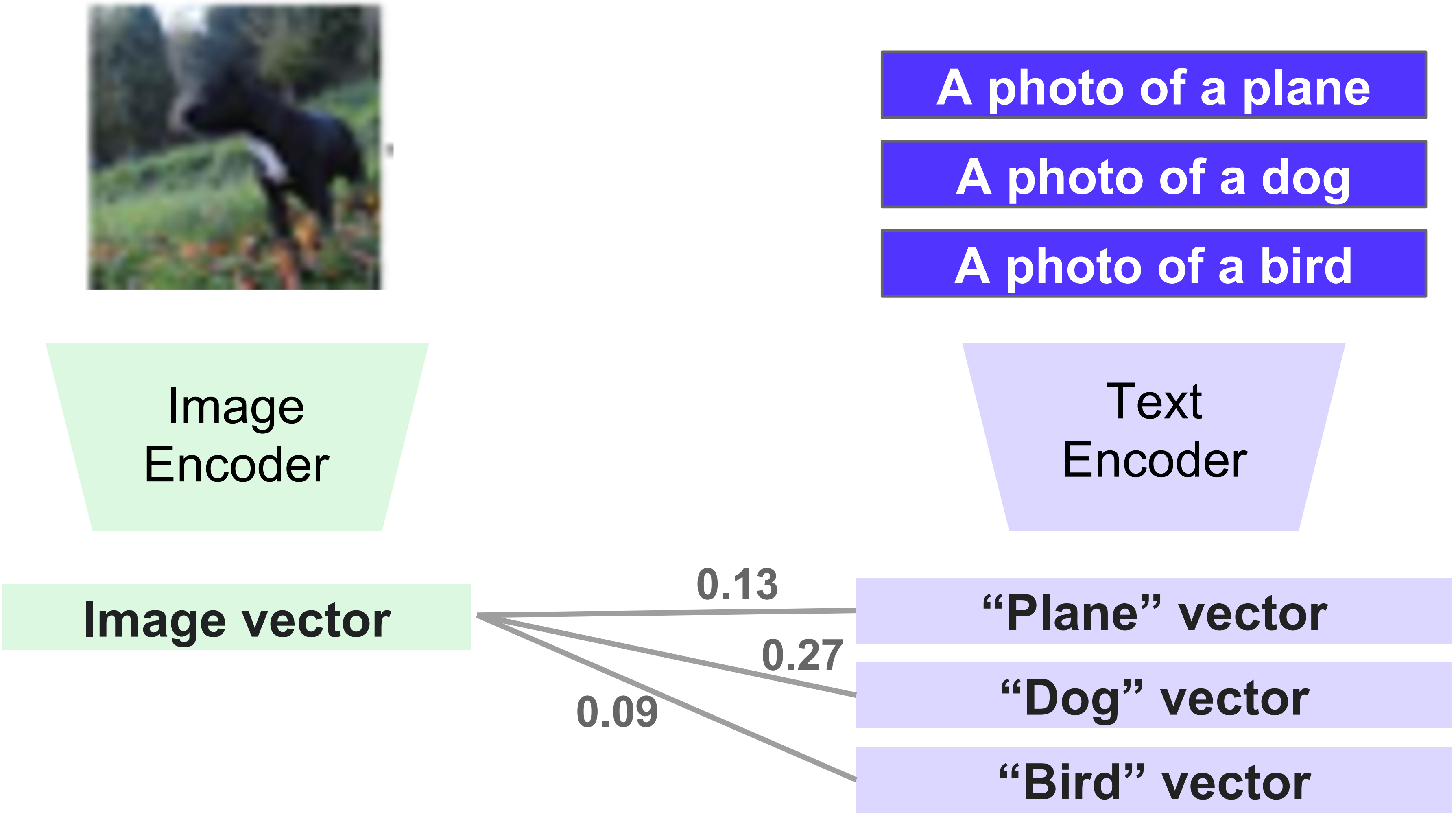
“Plane” vector

“Dog” vector

“Bird” vector

Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Since CLIP was trained with phrases, you can improve performance by using a phrase “A photo of a [category]”



A photo of a plane

A photo of a dog

A photo of a bird

Image Encoder

Text Encoder

Image vector

0.13

0.27

0.09

“Plane” vector

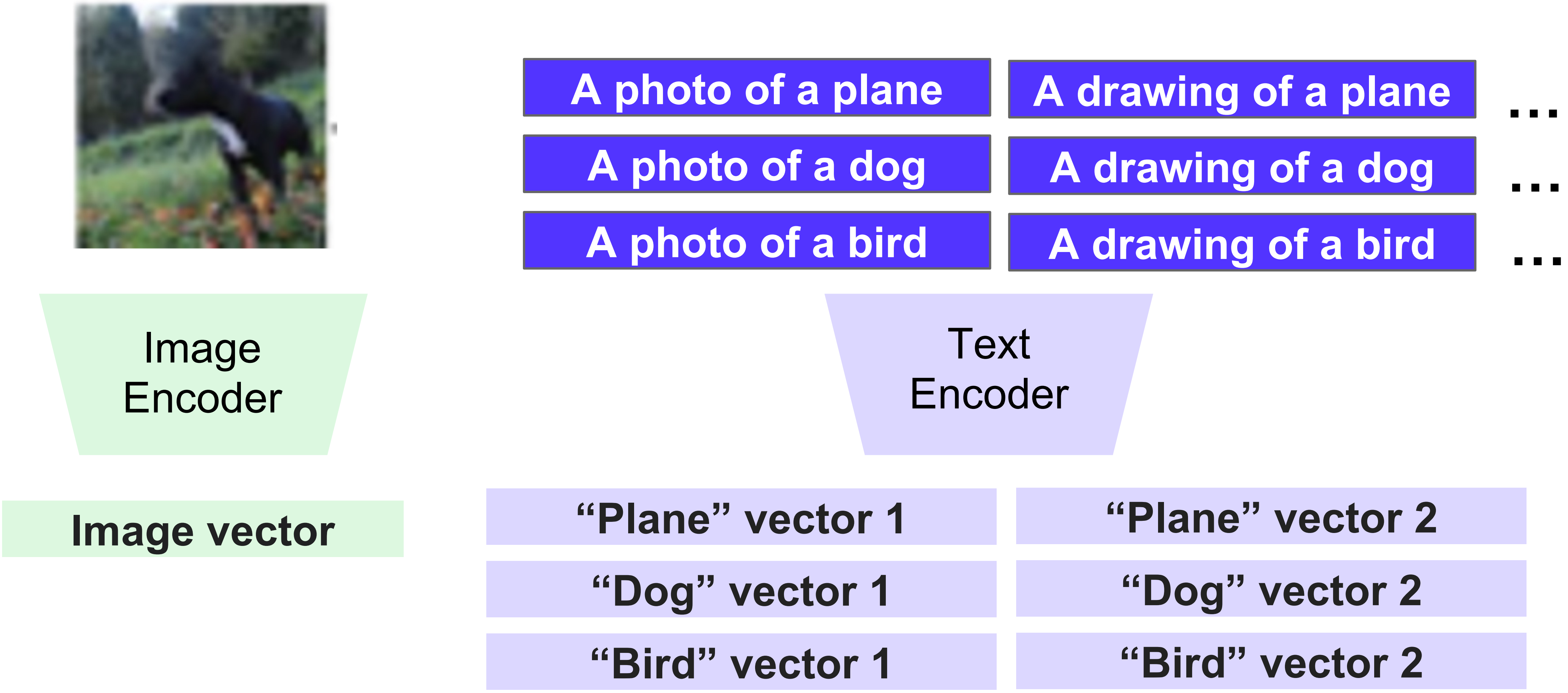
“Dog” vector

“Bird” vector

+1.3% on ImageNet

Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

A single phrase might be too biased.
Solution: Use multiple phrases



A photo of a plane | A drawing of a plane …

A photo of a dog | A drawing of a dog …

A photo of a bird | A drawing of a bird …

Image vector

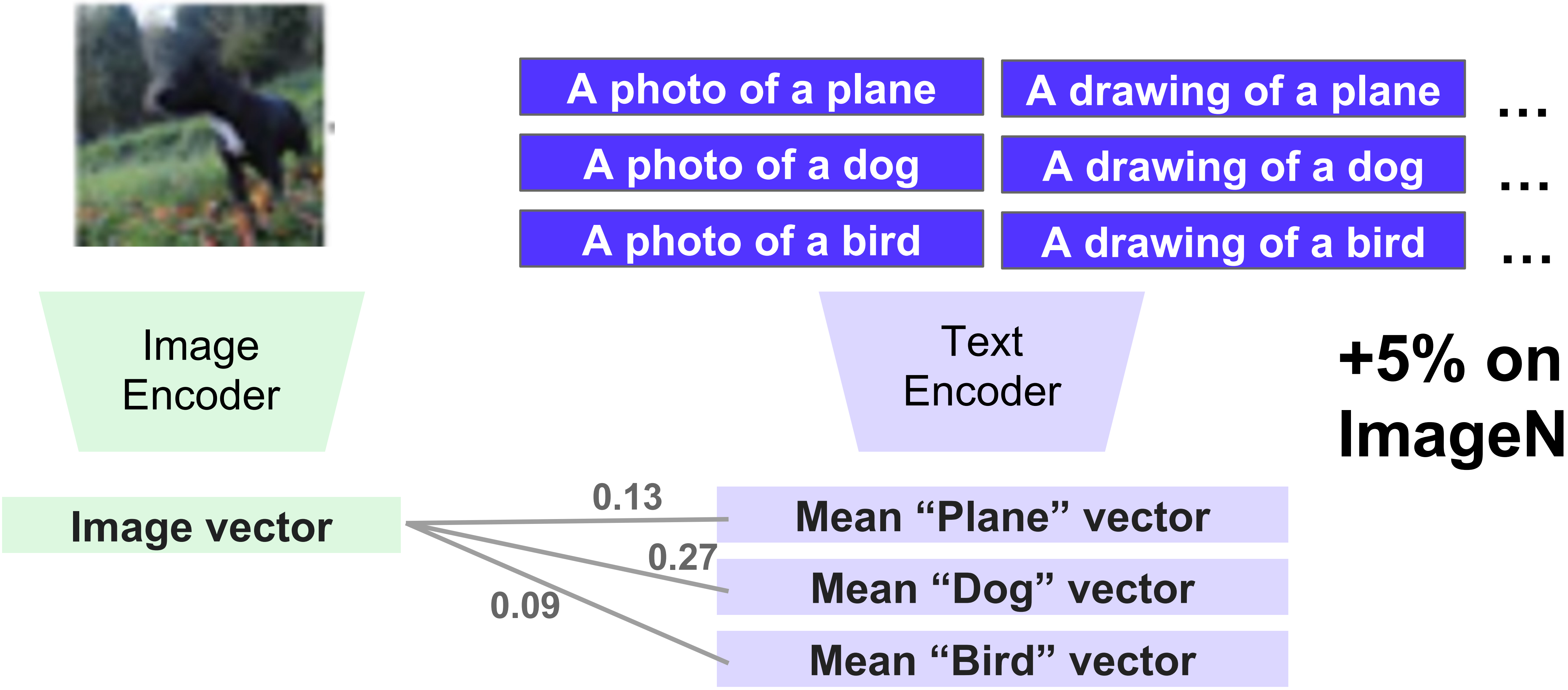
“Plane” vector 1 | “Plane” vector 2

“Dog” vector 1 | “Dog” vector 2

“Bird” vector 1 | “Bird” vector 2

Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Use the average vector across phrases as the representation for each category



A photo of a plane | A drawing of a plane | …

A photo of a dog | A drawing of a dog | …

A photo of a bird | A drawing of a bird | …

Image Encoder

Text Encoder

Image vector

0.13 **Mean “Plane” vector**

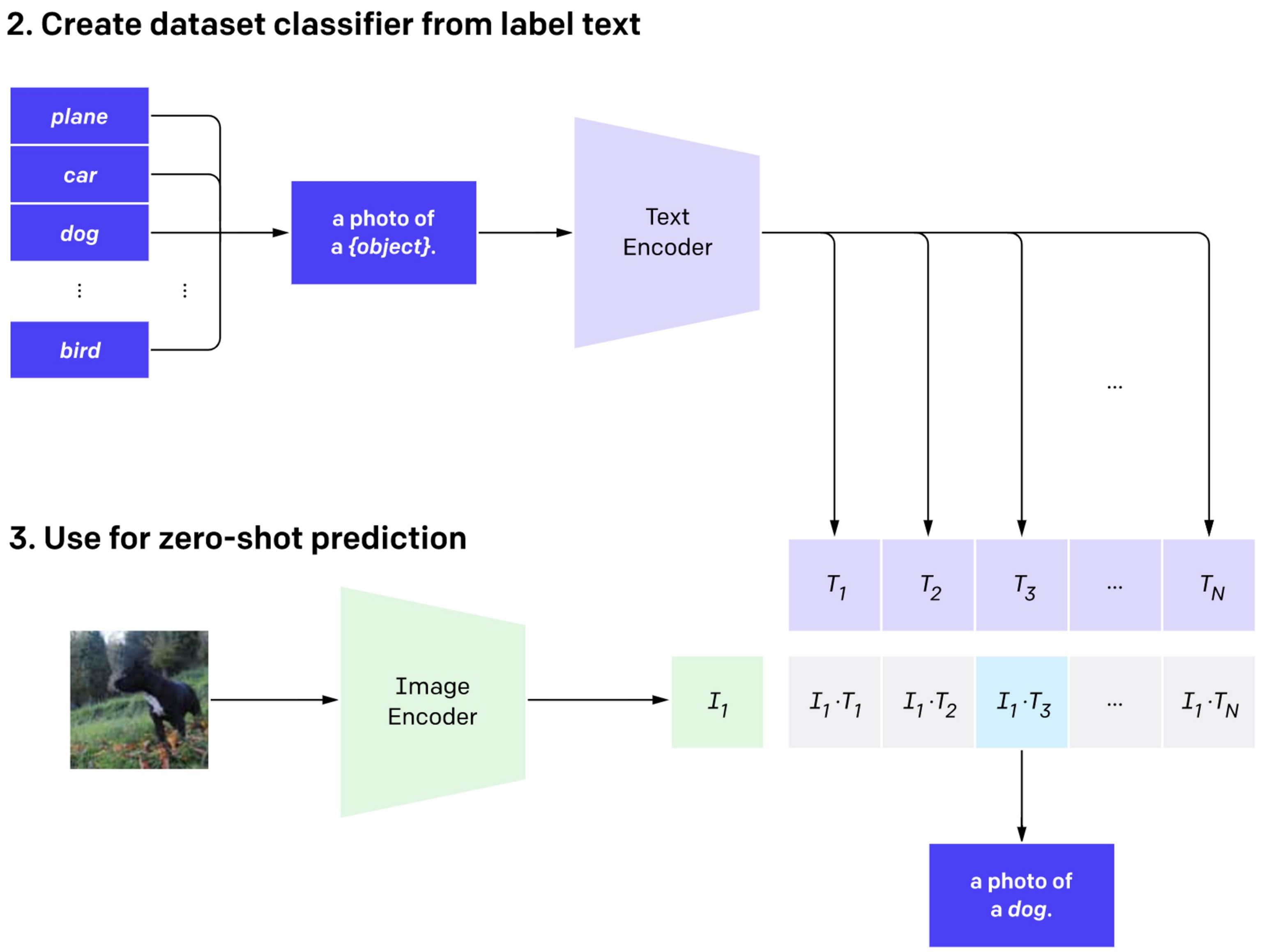
0.27 **Mean “Dog” vector**

0.09 **Mean “Bird” vector**

+5% on ImageNet

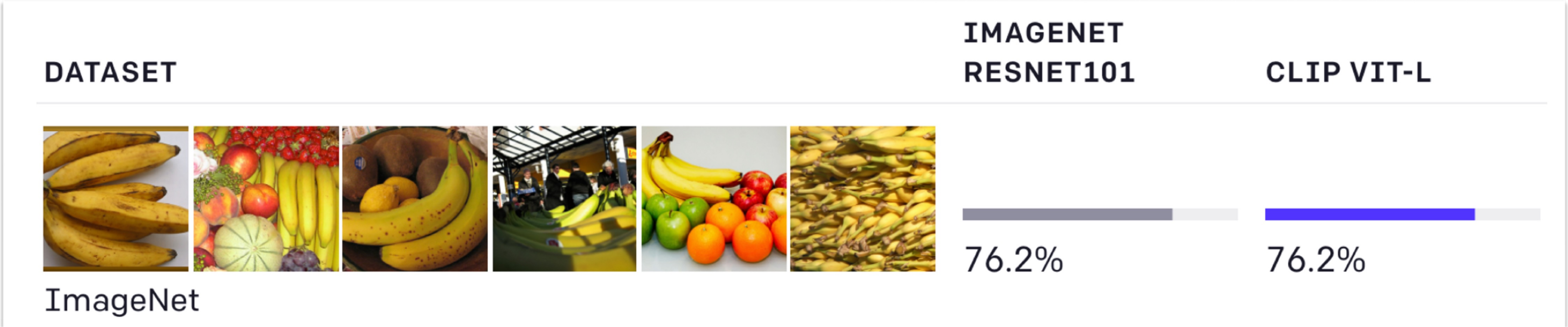
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

That’s it! Now, you can use CLIP as a foundation model for image classification for any dataset



Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Exciting result after training on 400M image-text pairs

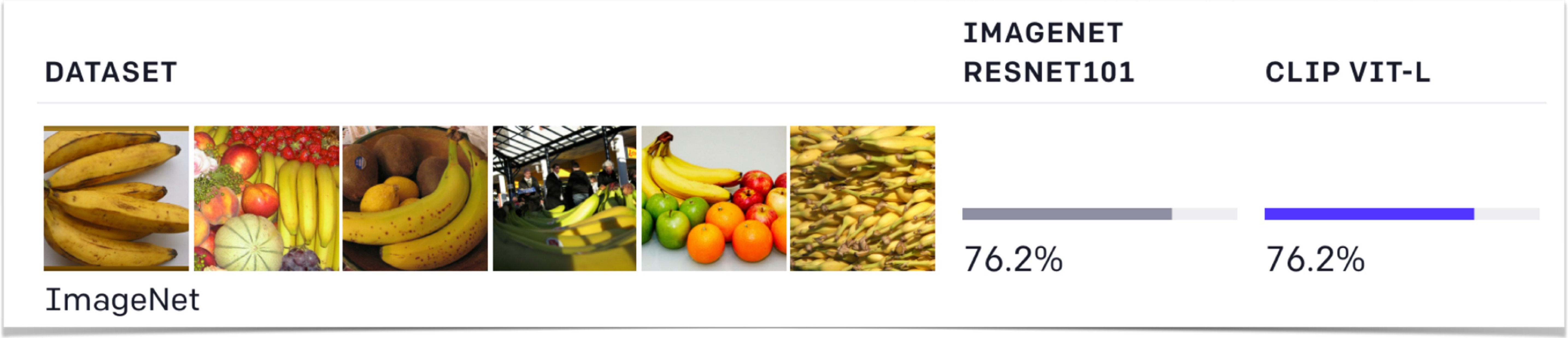


DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%

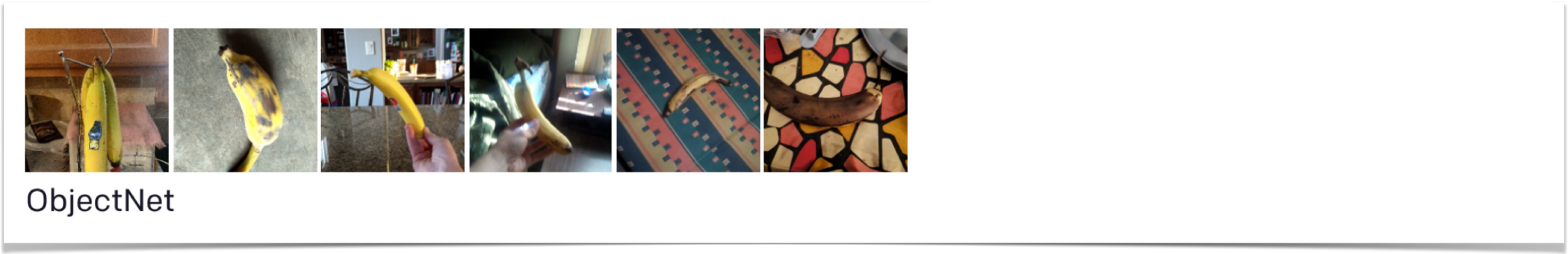
Matches the accuracy of of ResNet 101 that has been trained on ImageNet, except CLIP was trained with no human labels at all!

Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

Here’s where things get even more exciting



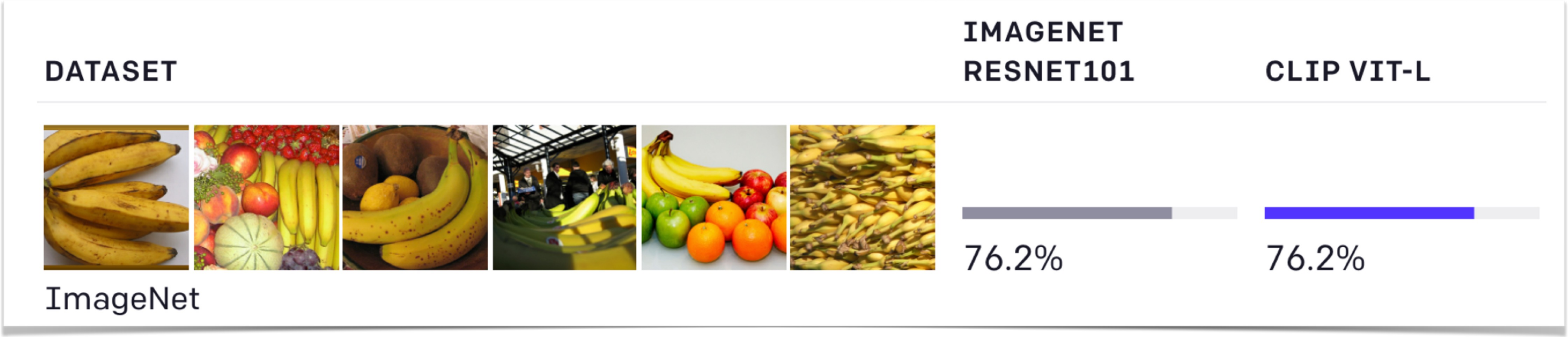
DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%



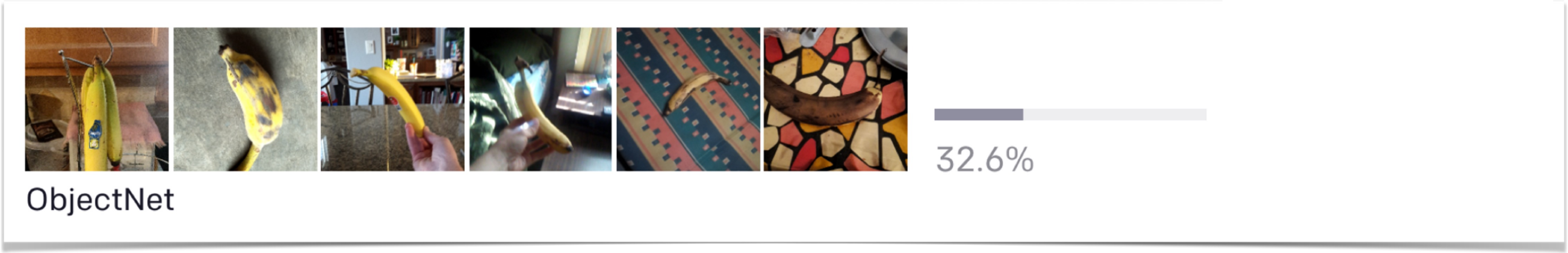
ObjectNet

Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Training on ImageNet doesn’t generalize to other datasets. ObjectNet contains the same categories but in weird viewpoints



DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%

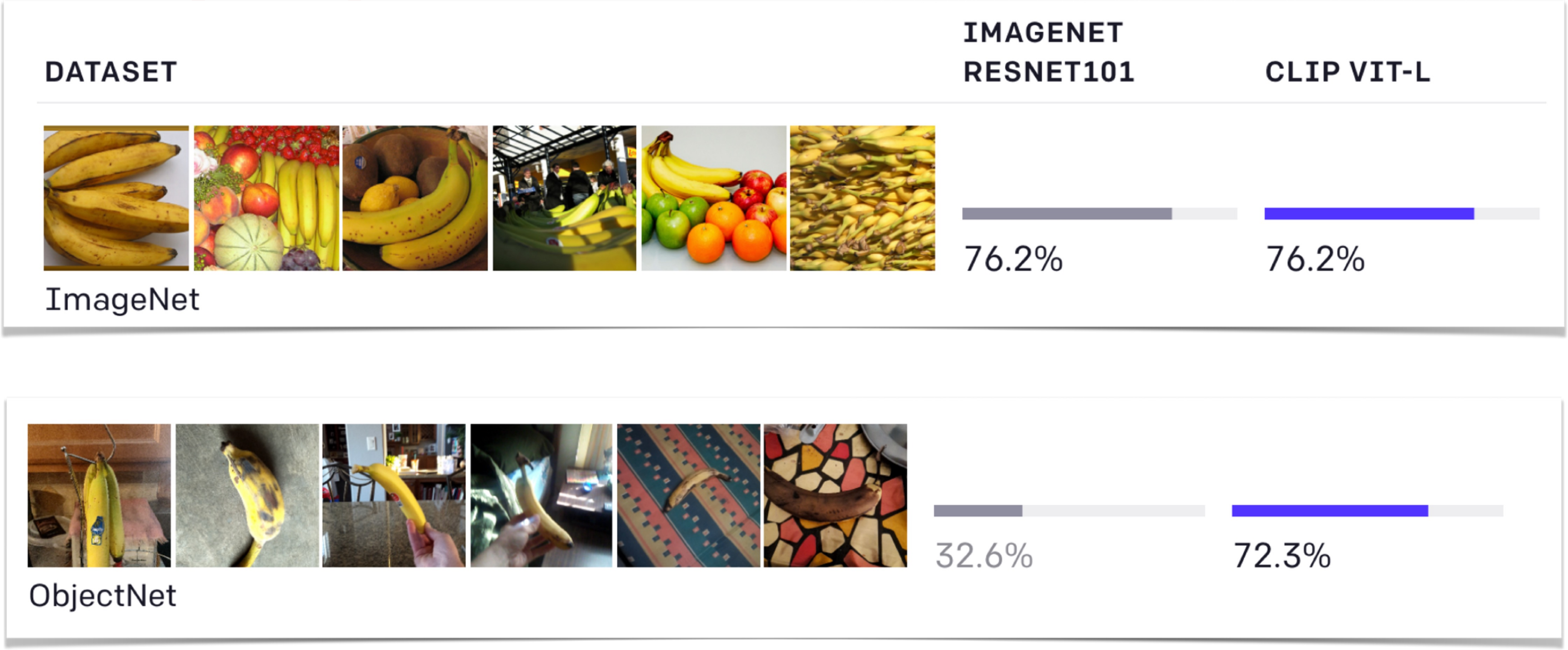


ObjectNet	32.6%

Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

But CLIP zero-shot does so well!

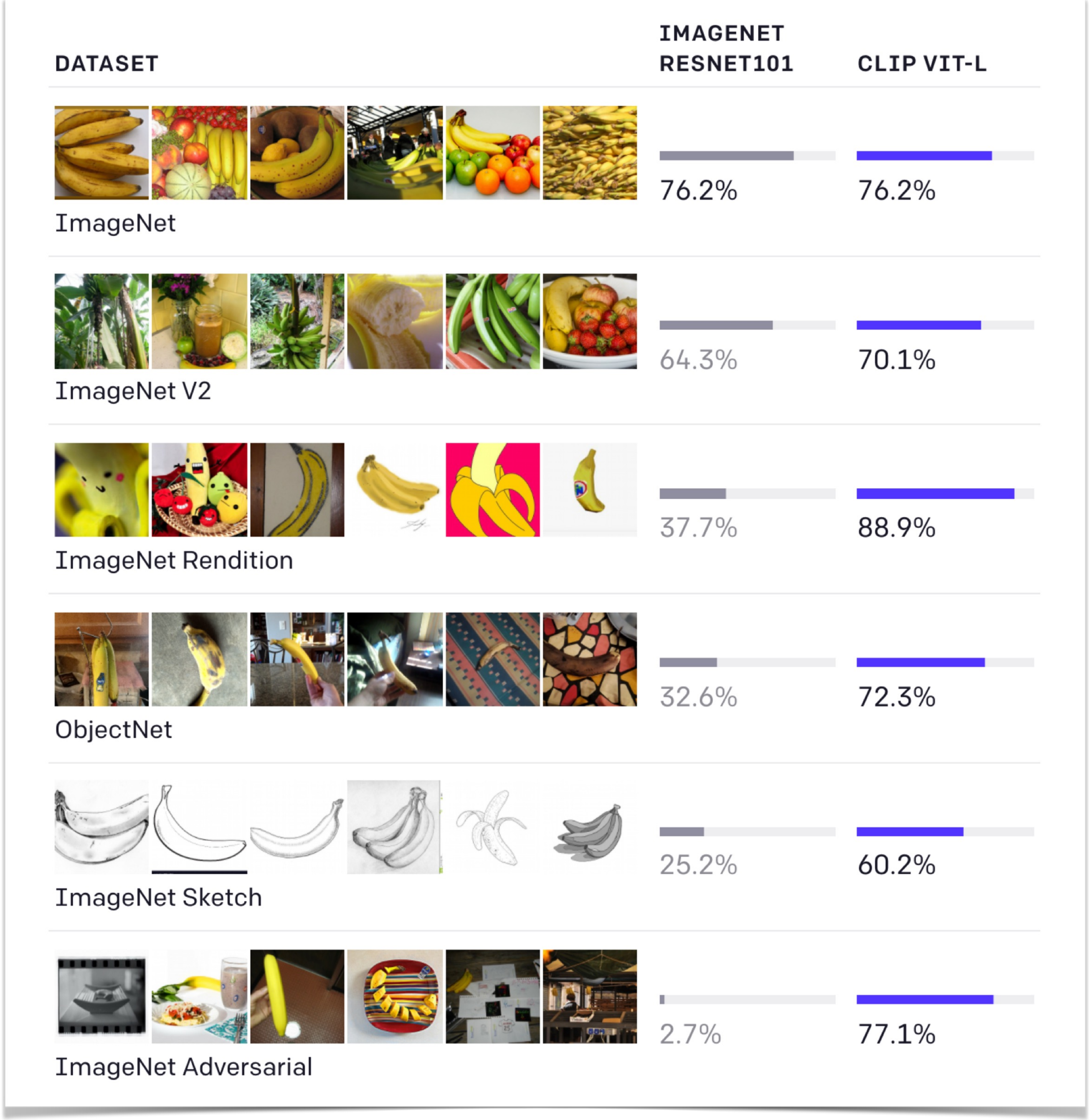
Q. Why do you think that is?



DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%
ObjectNet	32.6%	72.3%

Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

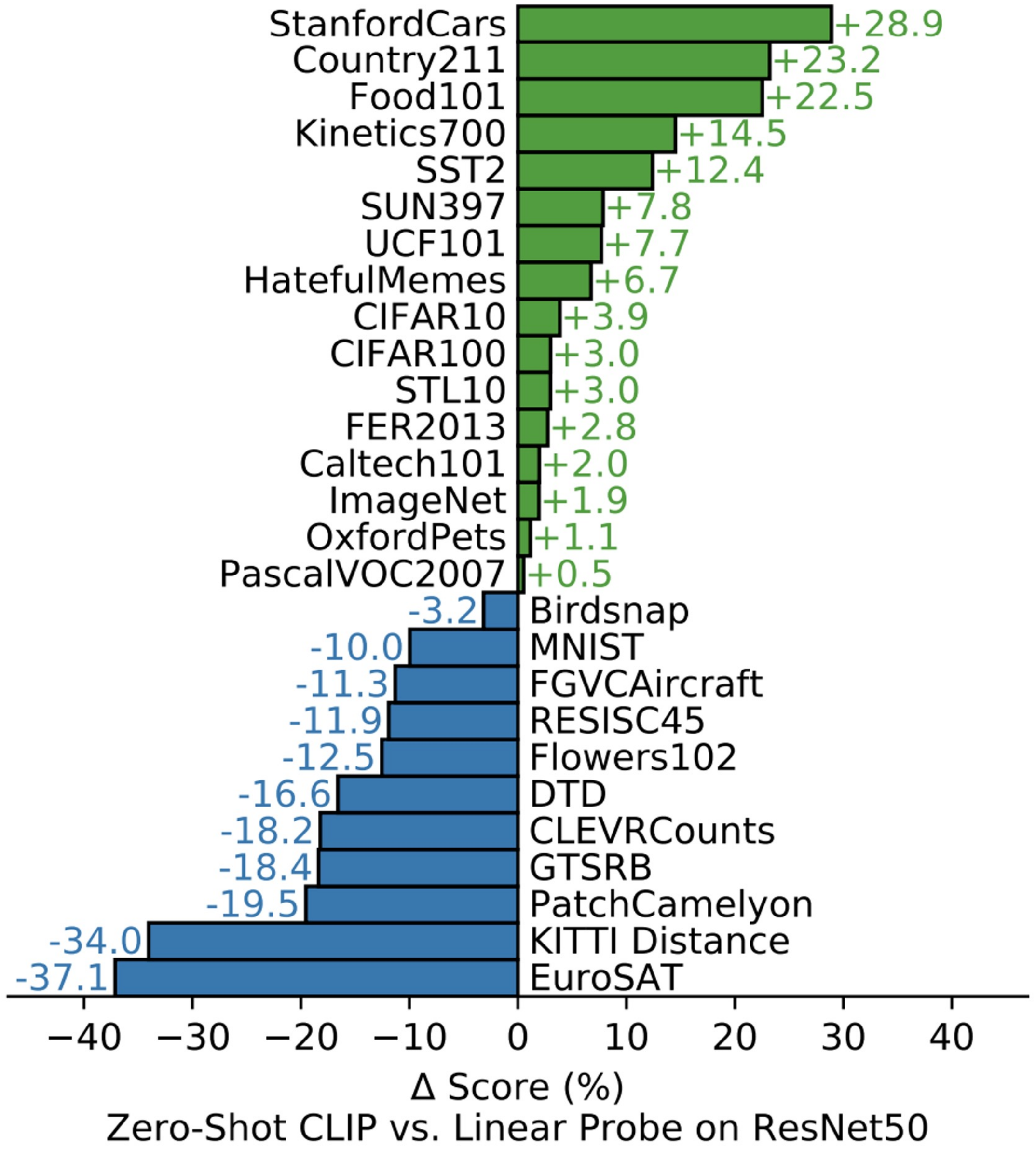
CLIP performance is great also on graphic images , sketches, adversarial datasets,



DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%
ImageNet V2	64.3%	70.1%
ImageNet Rendition	37.7%	88.9%
ObjectNet	32.6%	72.3%
ImageNet Sketch	25.2%	60.2%
ImageNet Adversarial	2.7%	77.1%

Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Difference in performance between linear probe vs zero-shot



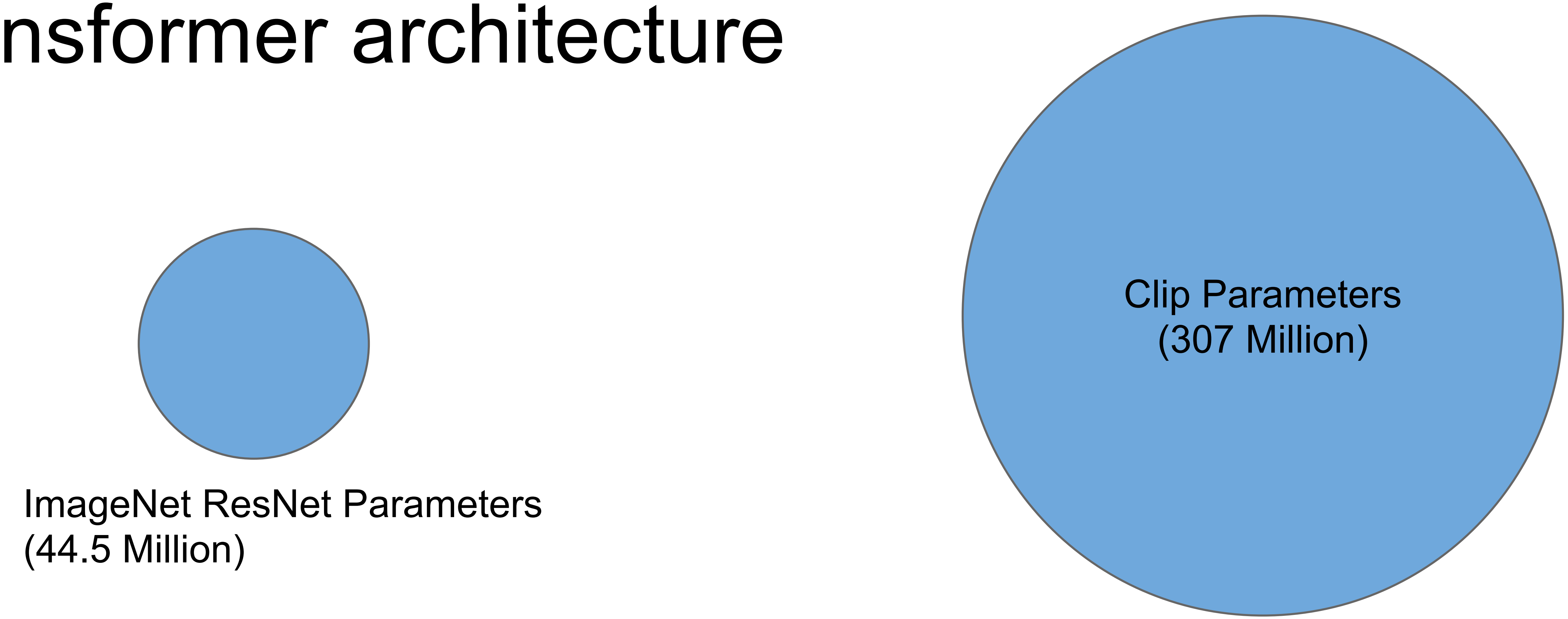
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Why does CLIP perform so well?

How can no labels beat labels??

Scale!

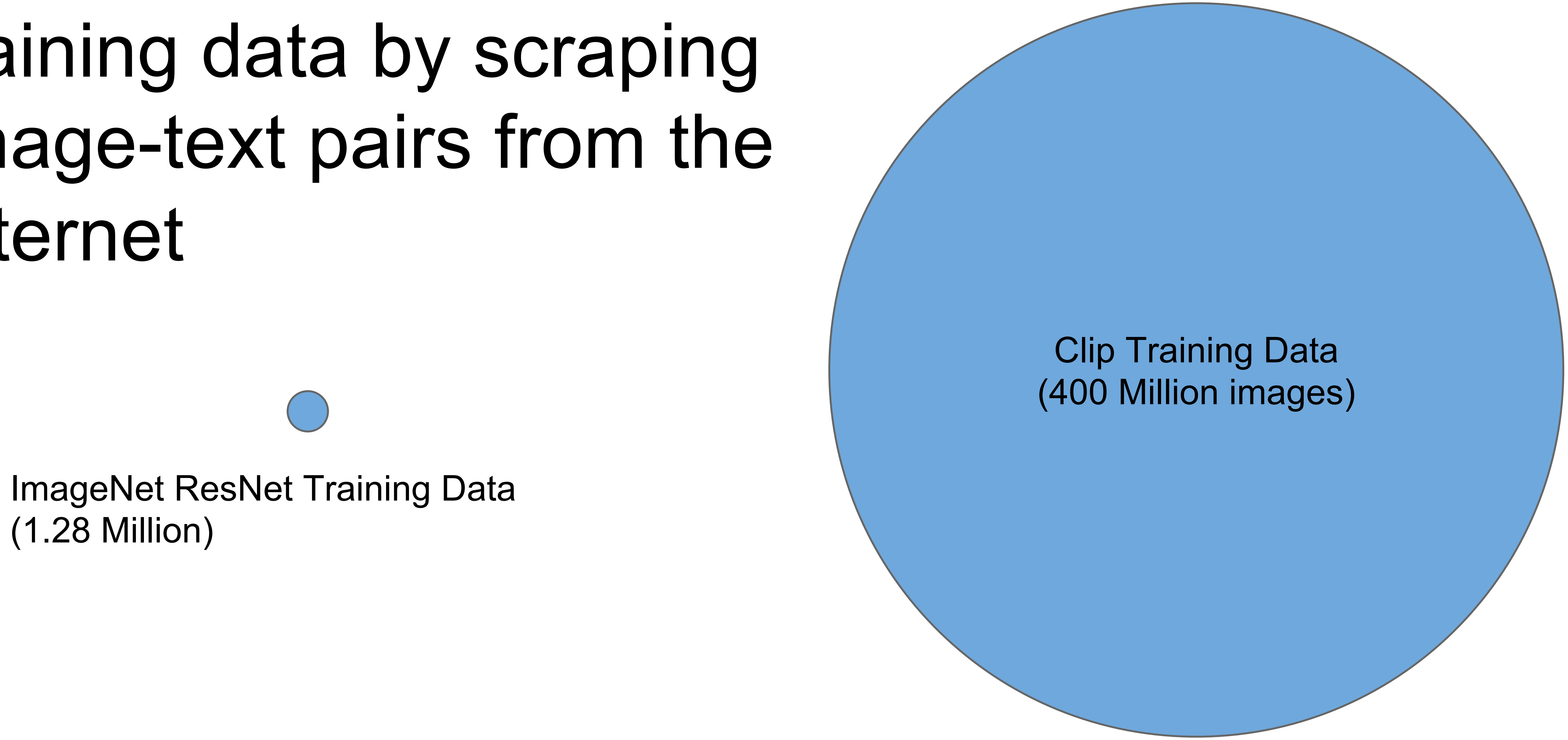
CLIP scaled up the model parameters with the transformer architecture



ImageNet ResNet Parameters
(44.5 Million)

Clip Parameters
(307 Million)

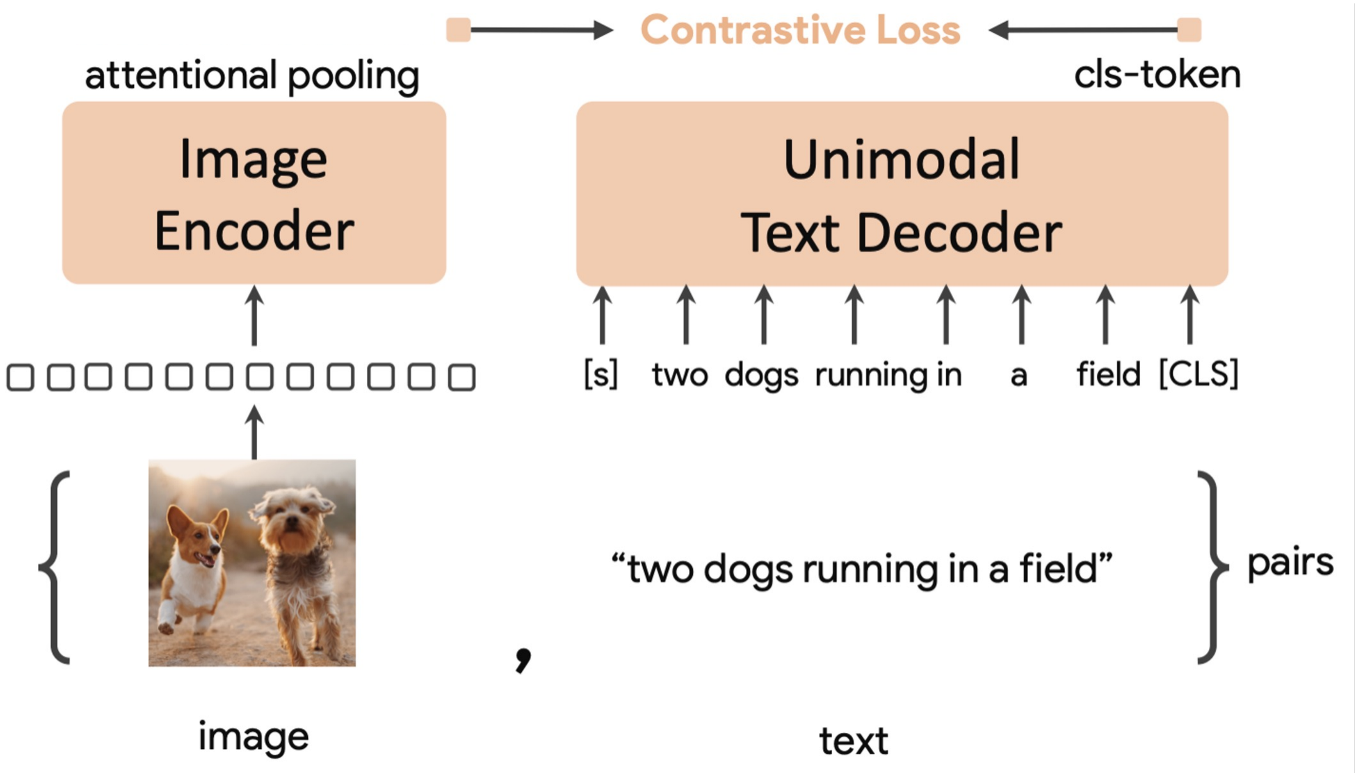
CLIP Scaled up the training data by scraping image-text pairs from the internet



ImageNet ResNet Training Data
(1.28 Million)

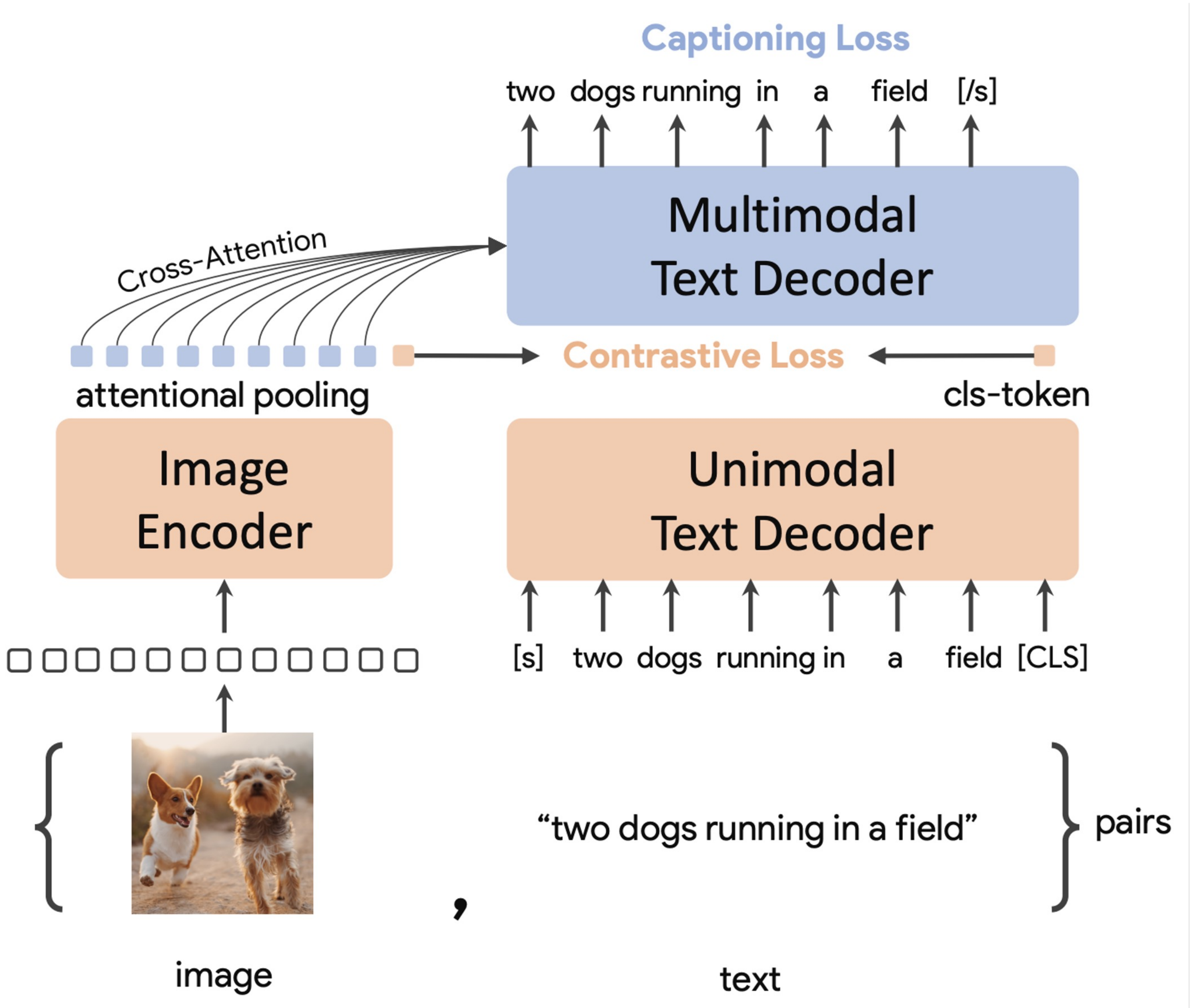
Clip Training Data
(400 Million images)

CoCa improved upon CLIP by adding a generation objective



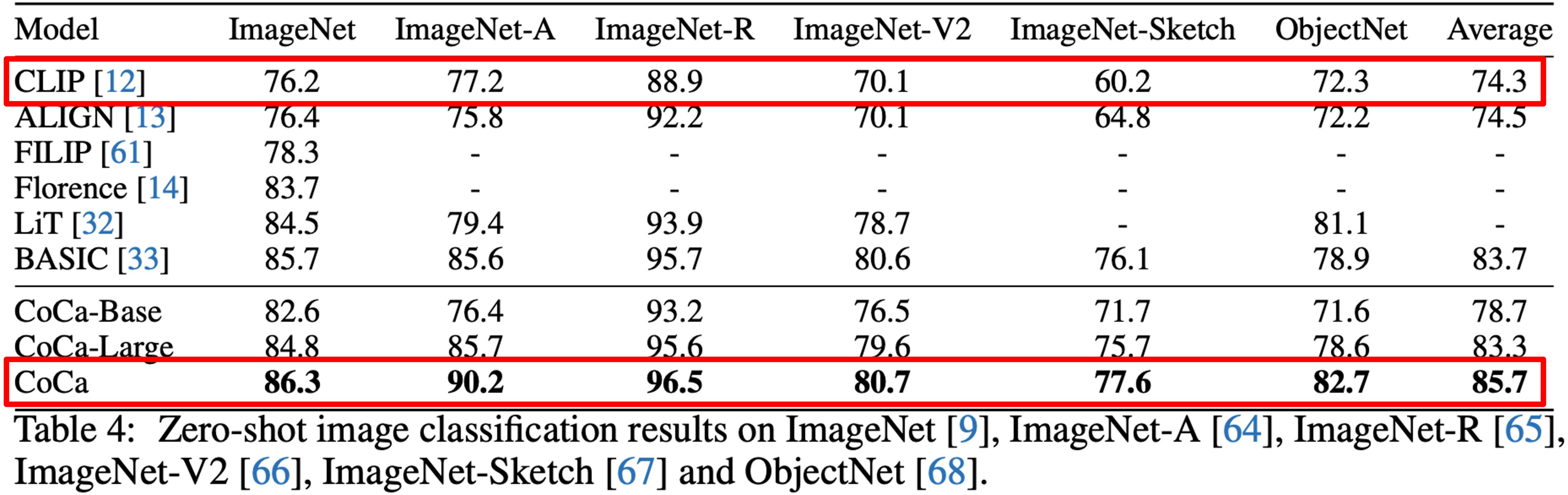
"Contrastive Captioners are Image-Text Foundation Models", 2022

CoCa added a decoder with a captioning loss



"Contrastive Captioners are Image-Text Foundation Models", 2022

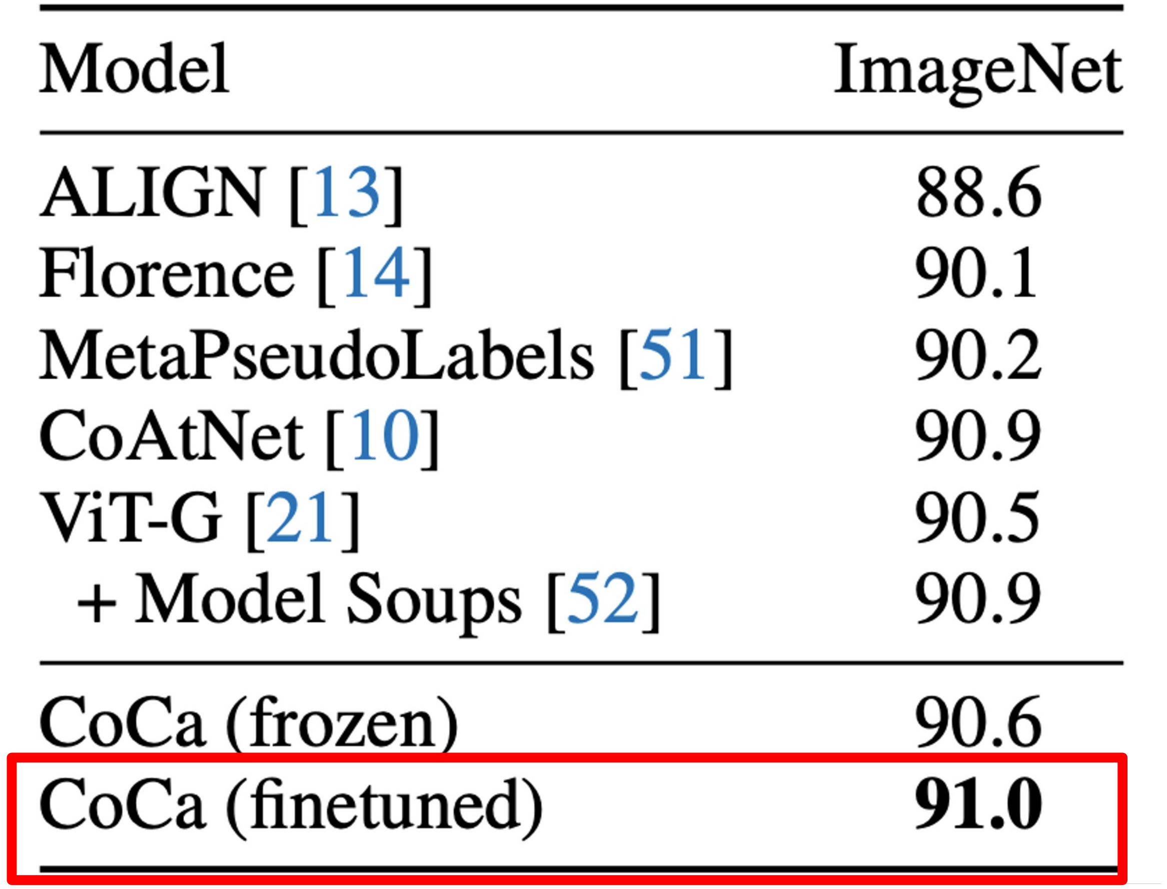
CoCa: Contrastive Captioners are Image-Text Foundation Models



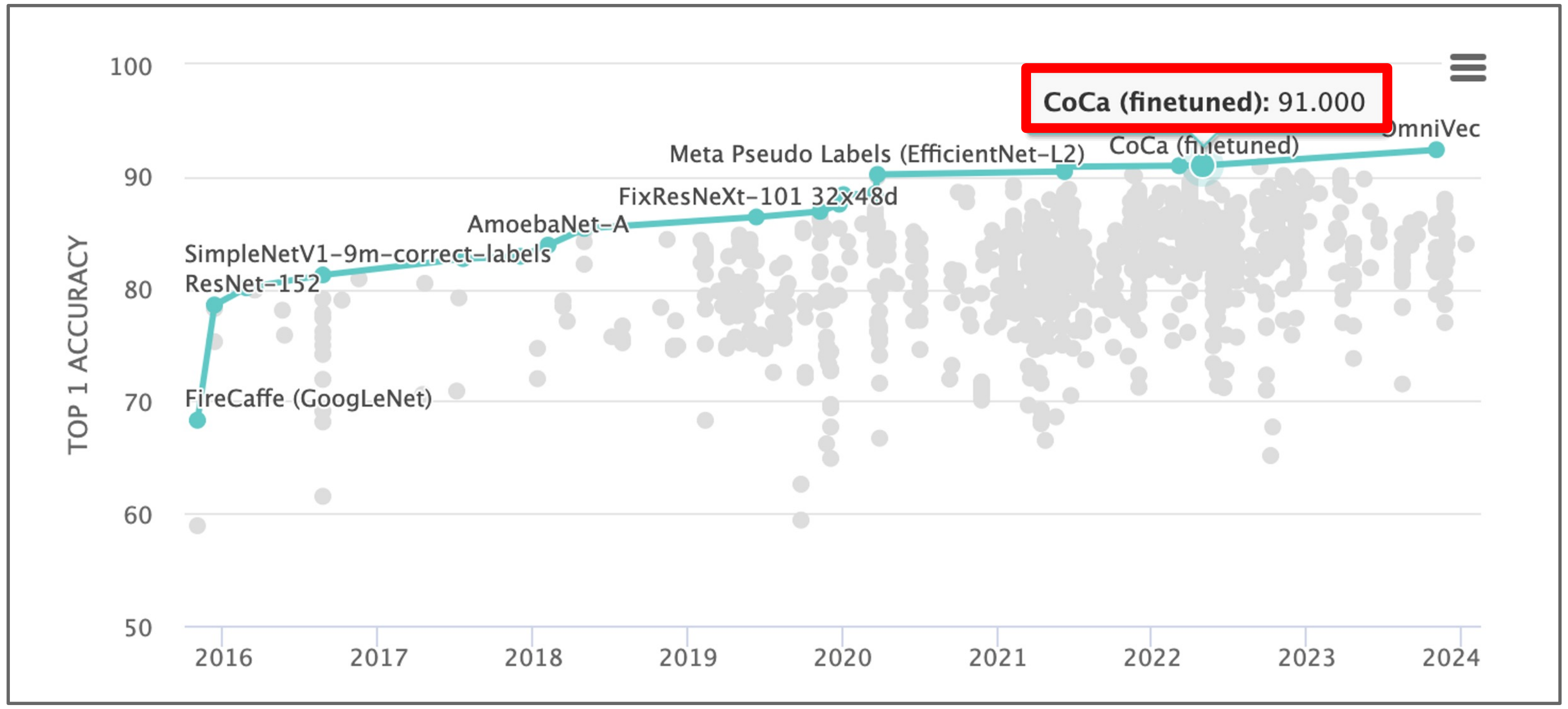
Model	ImageNet	ImageNet-A	ImageNet-R	ImageNet-V2	ImageNet-Sketch	ObjectNet	Average
CLIP [12]	76.2	77.2	88.9	70.1	60.2	72.3	74.3
ALIGN [13]	76.4	75.8	92.2	70.1	64.8	72.2	74.5
FILIP [61]	78.3	-	-	-	-	-	-
Florence [14]	83.7	-	-	-	-	-	-
LiT [32]	84.5	79.4	93.9	78.7	-	81.1	-
BASIC [33]	85.7	85.6	95.7	80.6	76.1	78.9	83.7
CoCa-Base	82.6	76.4	93.2	76.5	71.7	71.6	78.7
CoCa-Large	84.8	85.7	95.6	79.6	75.7	78.6	83.3
CoCa	**86.3**	**90.2**	**96.5**	**80.7**	**77.6**	**82.7**	**85.7**

Table 4: Zero-shot image classification results on ImageNet [9], ImageNet-A [64], ImageNet-R [65], ImageNet-V2 [66], ImageNet-Sketch [67] and ObjectNet [68].

Classifier foundation models now beat all other models on ImageNet



Model	ImageNet
ALIGN [13]	88.6
Florence [14]	90.1
MetaPseudoLabels [51]	90.2
CoAtNet [10]	90.9
ViT-G [21]	90.5
+ Model Soups [52]	90.9
CoCa (frozen)	90.6
CoCa (finetuned)	**91.0**



Advantages of CLIP-style models

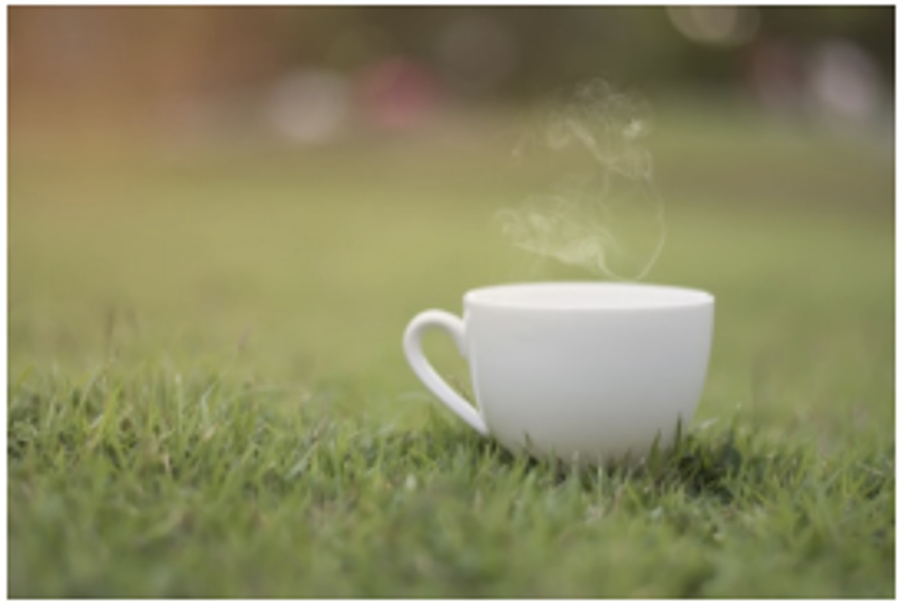
1. Dot product is super efficient
   a. Easy to train (enables scaling)
   b. Fast inference, e.g., retrieval over 5B images

2. Open-vocabulary (zero-shot generalization)

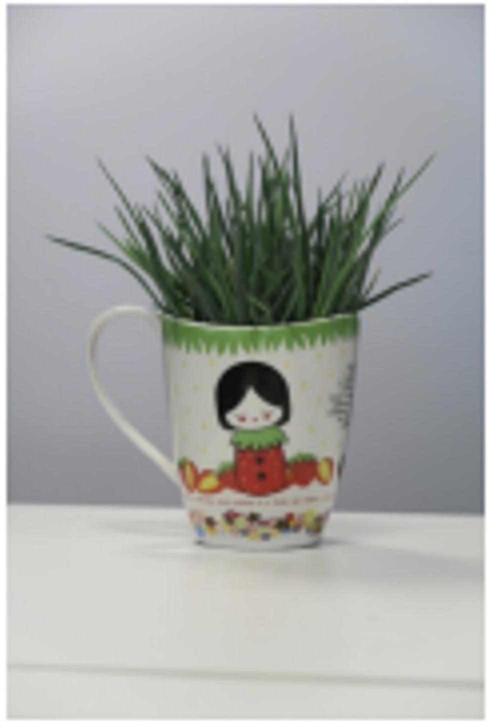
3. Can be chained with other models (CuPL)
   [we will discuss this later today]

April 2022, Tristan Thrush et al:

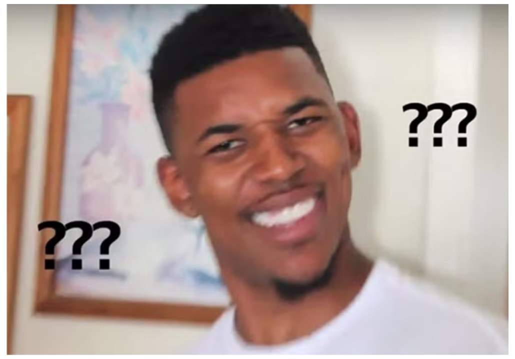
CLIP can’t distinguish between:



there is a mug in some grass



there is some grass in a mug



Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

Increasing batch size helps you understand fine-grained concepts

Batch size: 4	“animal”
Batch size: 100	“dog”
Batch size: **32000**	“Welsh Corgi”

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

Increasing batch size helps you understand fine-grained concepts

But there’s a limit to how fine-grained you can get this way

Even in a batch of 32K, it’s unlikely you see both “a mug in some grass” and “some grass in a mug”

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

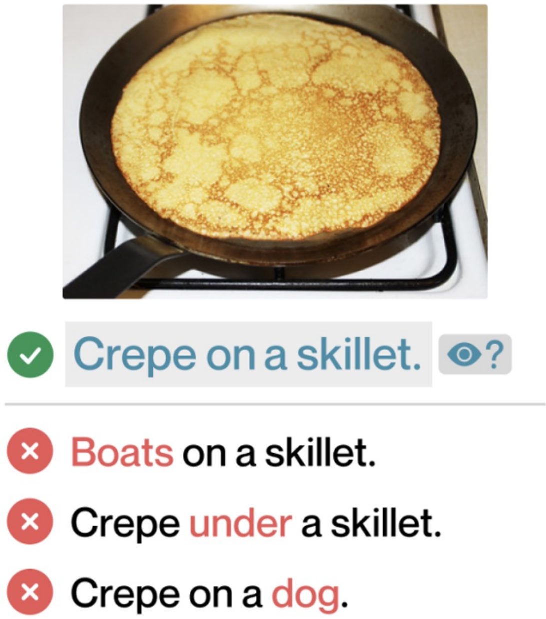
Winoground

there is a mug in some grass

there is some grass in a mug

“compositionality”

CREPE



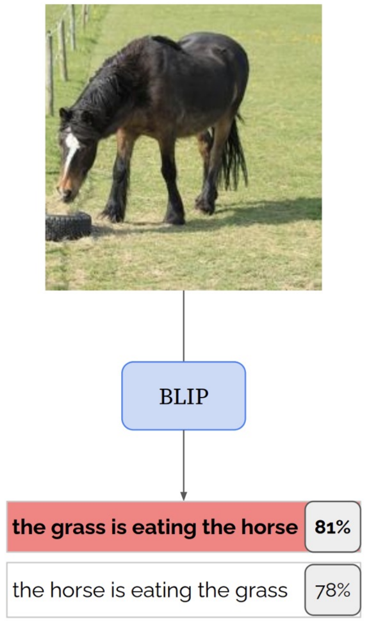
✓ Crepe on a skillet. ?

✗ Boats on a skillet.

✗ Crepe under a skillet.

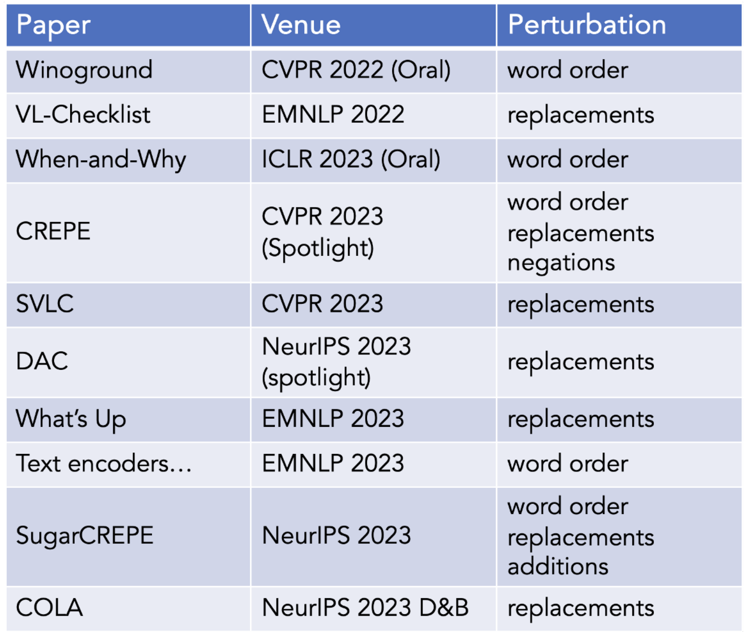
✗ Crepe on a dog.

ARO



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

…

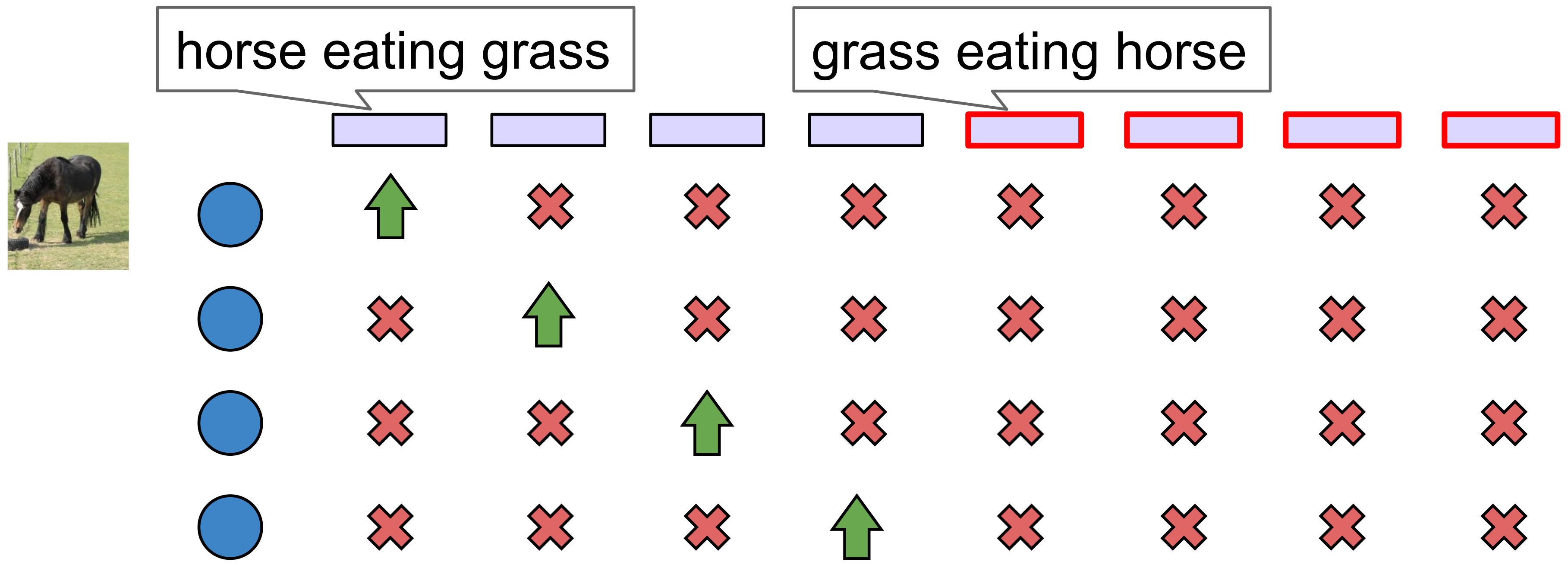


Paper	Venue	Perturbation
Winoground	CVPR 2022 (Oral)	word order
VL-Checklist	EMNLP 2022	replacements
When-and-Why	ICLR 2023 (Oral)	word order
CREPE	CVPR 2023 (Spotlight)	word order replacements negations
SVLC	CVPR 2023	replacements
DAC	NeurIPS 2023 (spotlight)	replacements
What's Up	EMNLP 2023	replacements
Text encoders…	EMNLP 2023	word order
SugarCREPE	NeurIPS 2023	word order replacements additions
COLA	NeurIPS 2023 D&B	replacements

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

Solution? Hard Negative Fine-Tuning



TODO: Get NegCLIP scores for these captions now

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

But training with hard negatives has its own problems…

A black cat and a brown dog
✓

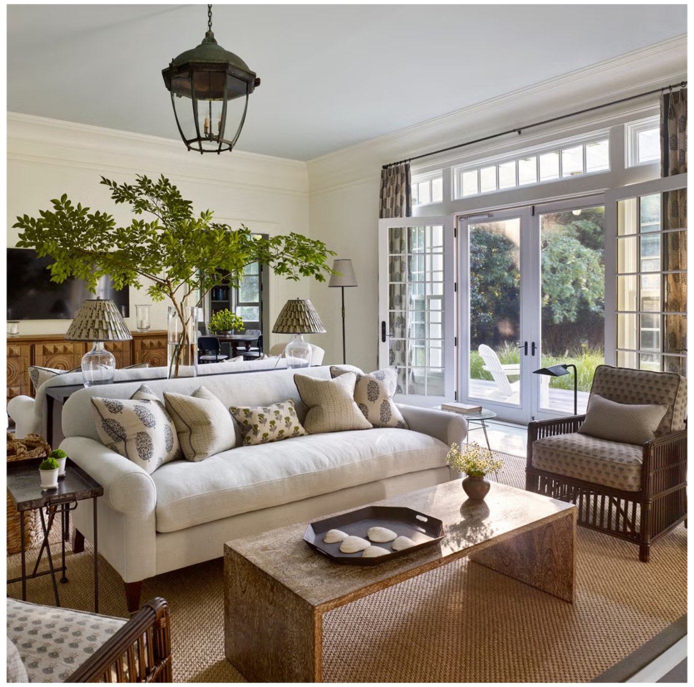
A brown cat and a black dog
✗

A brown dog and a black cat
✗

“hard positives”

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts
2. Image-level captions are insufficient supervision

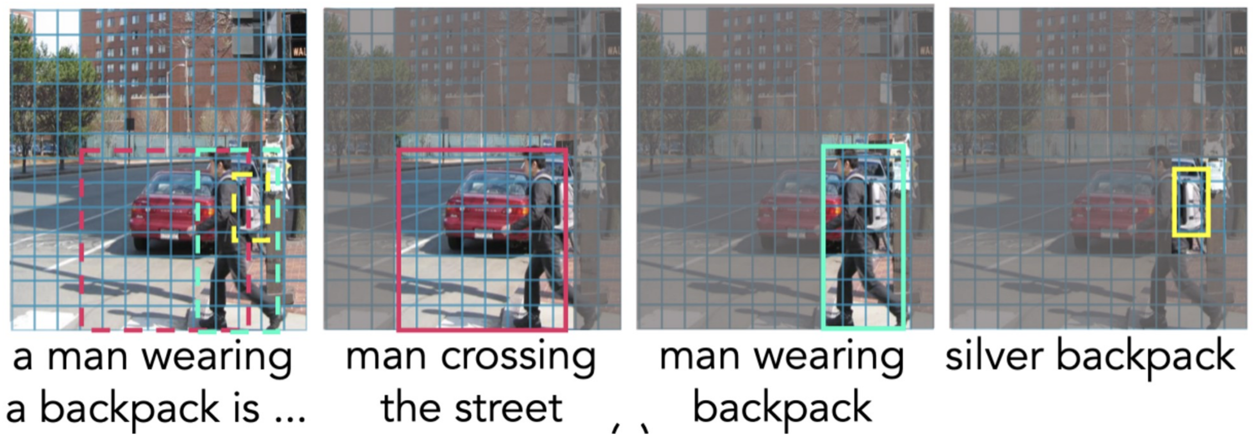


“living room”	✓
“house plants”	✗
“couch”	✗

Disadvantages of CLIP-style models

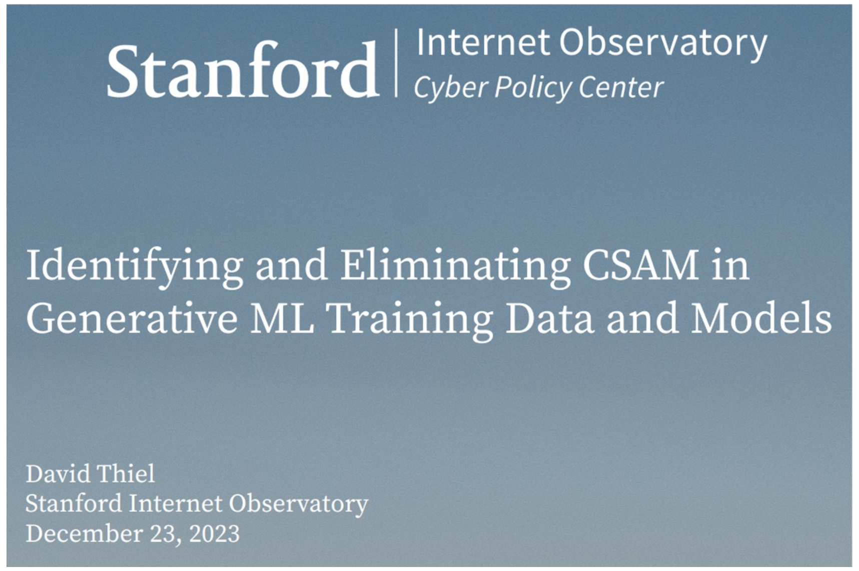
1. Rely too heavily on batch size to learn concepts
2. Image-level captions are insufficient supervision

Also train on region captions
with bounding box coordinates



Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts
2. Image-level captions are insufficient supervision
3. You can’t know everything in your 5B dataset



Stanford | Internet Observatory
Cyber Policy Center

Identifying and Eliminating CSAM in Generative ML Training Data and Models

David Thiel
Stanford Internet Observatory
December 23, 2023

It’s extremely important to be intentional about data collection and filtering

Foundation Models

Language

ELMo
BERT
GPT
T5

Classification

CLIP
CoCa

LM + Vision

LLaVA
Flamingo
GPT-4V
Gemini
Molmo

And More!

Segment Anything
Whisper
Dalle
Stable Diffusion
Imagen

Chaining

LMs + CLIP
Visual Programming

LLaVA

Motivation: Language models which do next token prediction can be applied to a wide variety of tasks at inference (Math, sentiment analysis, symbolic reasoning)

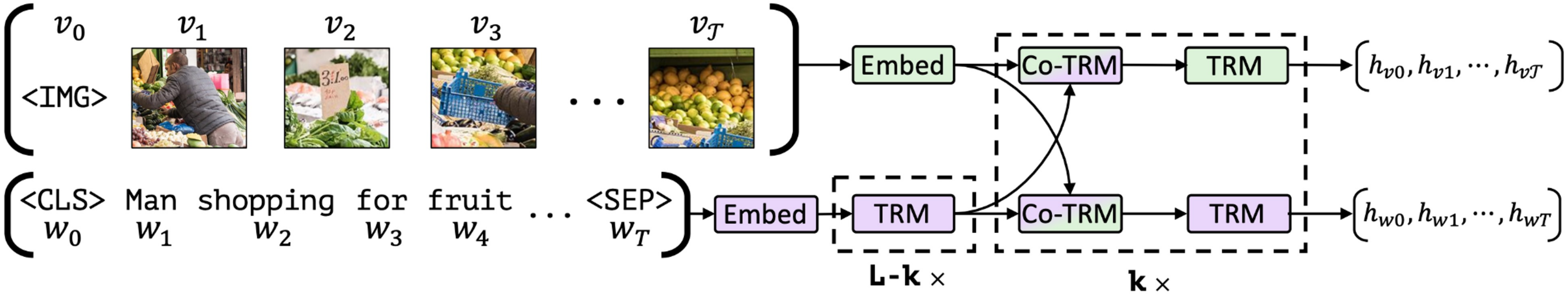
Can we build a model that can accept images and text as input, and then output text?

→ **Vision-Language Models**

First, some historical context

Vision-Language Models didn't start with LLaVA!

They go as far back as 2019 → ViLBERT



Historical context

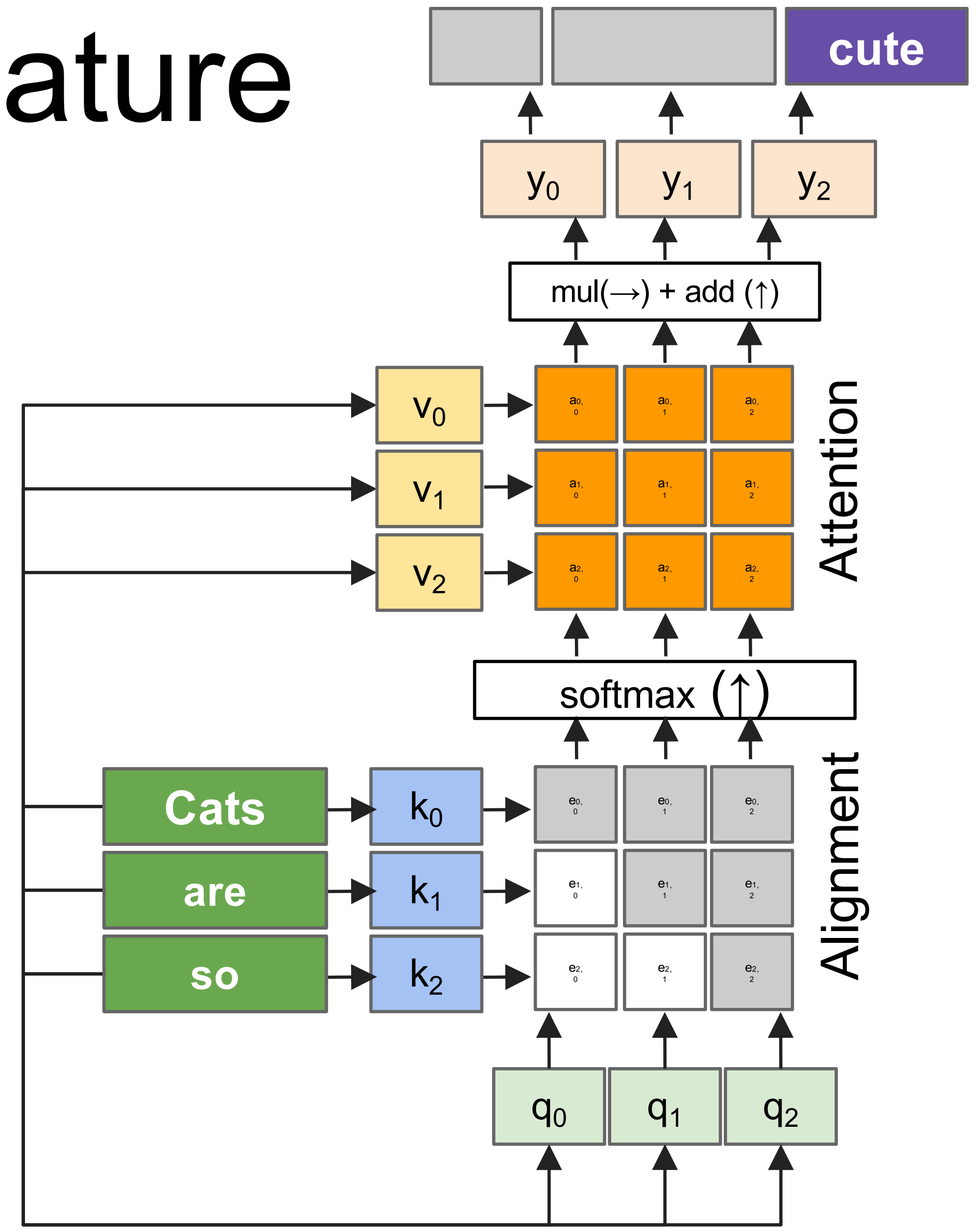
Vision-Language Models didn’t start with LLaVA!

They go as far back as 2019 → ViLBERT

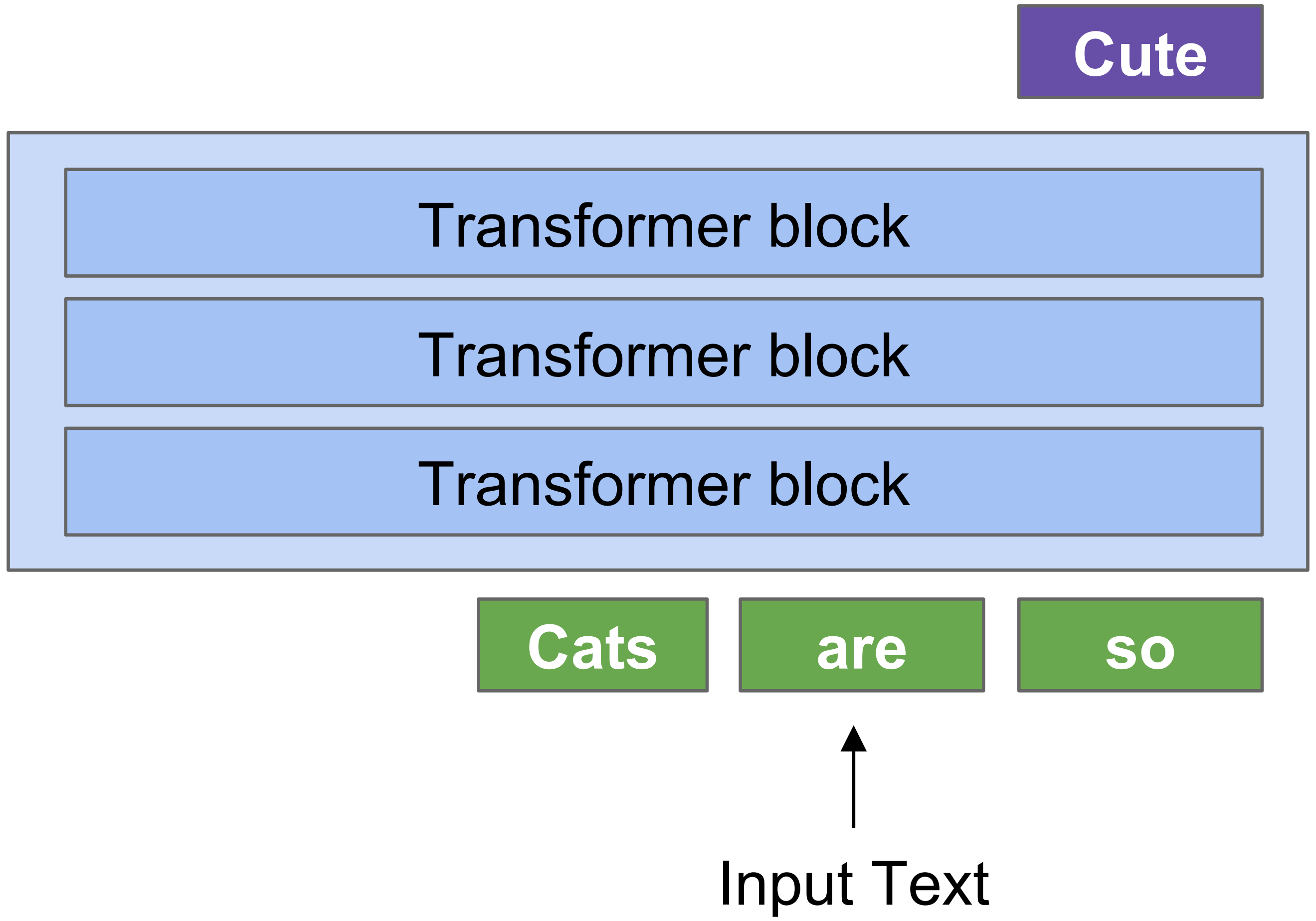
BUT, they had to finetune for each task separately, with non-trivial task-specific methods (e.g., Mask-RCNN bounding box re-ranking for RefCOCO)

→ Same paradigm as we discussed right at the beginning of this lecture:
very task-specific

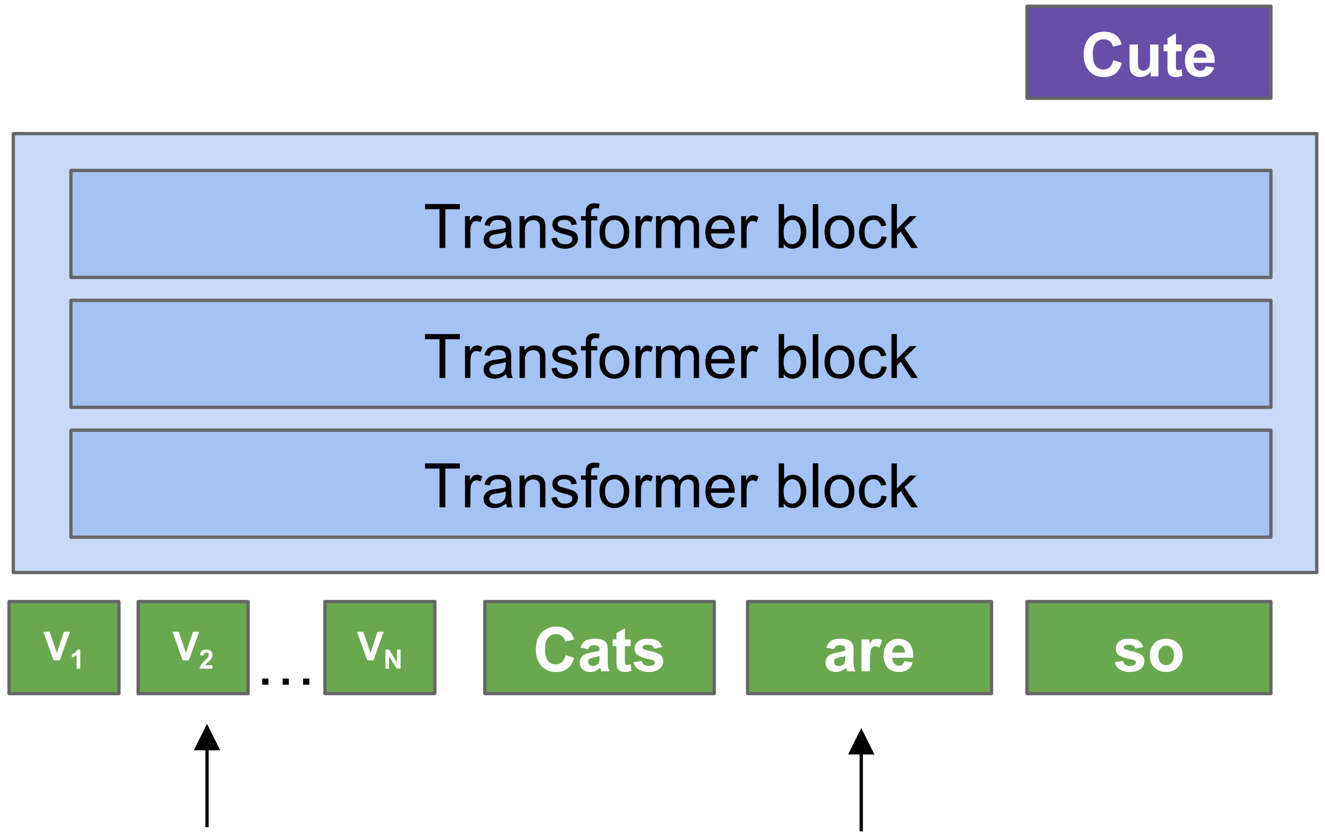
LLaVA uses the autoregressive nature of LLMs



Recall how transformers decode language



Key idea behind LLaVA – add visual information to the LLM

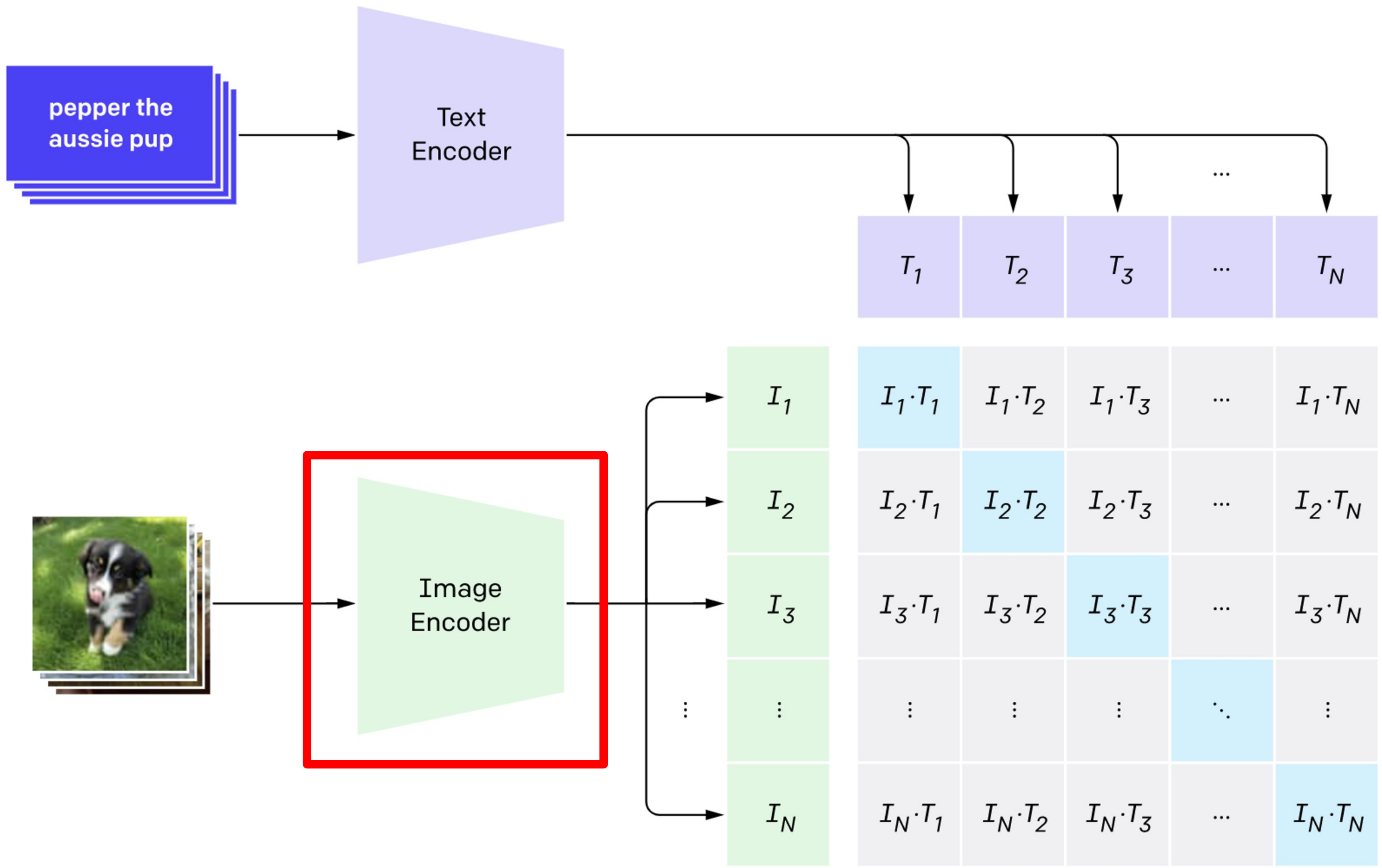


Input image tokens

Input Text

Which image tokens work best here?

The CLIP encoder is a good option!



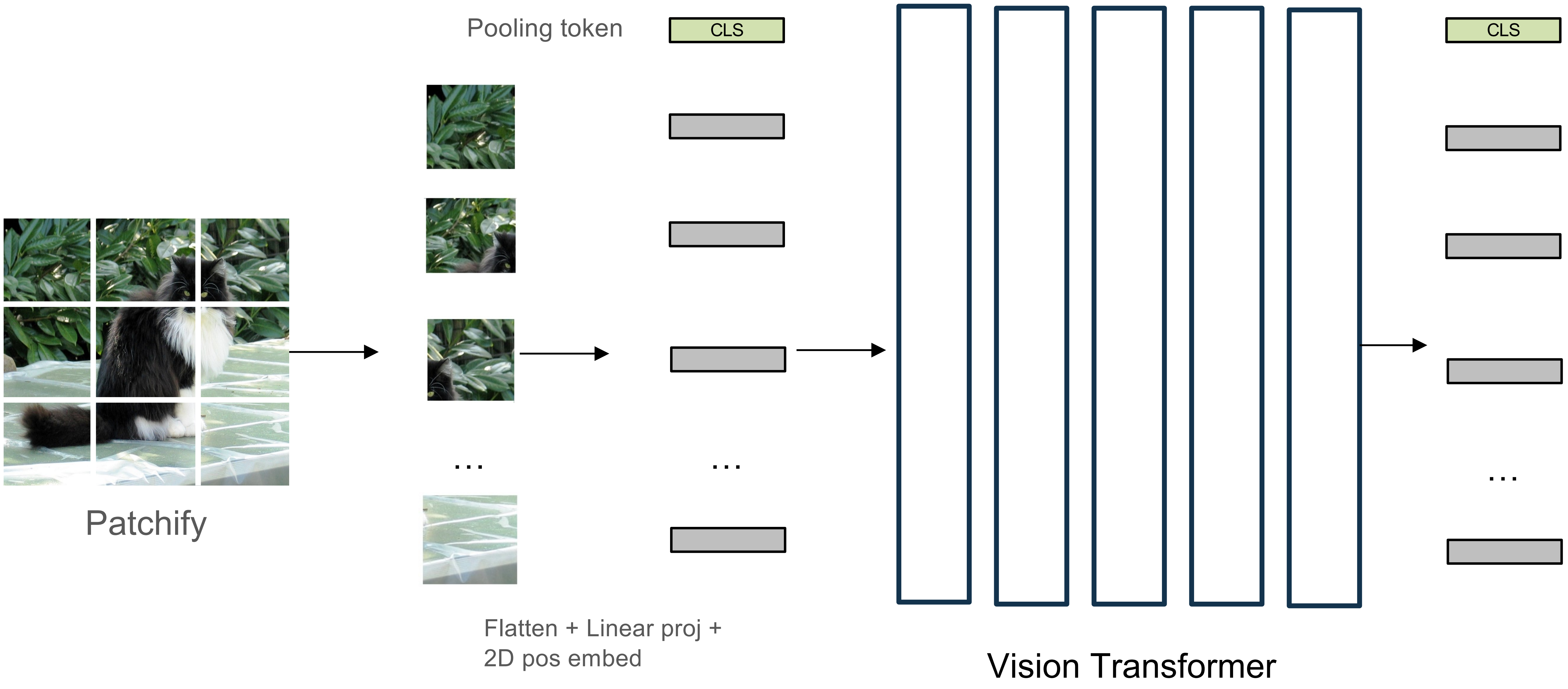
1. Contrastive pre-training

	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

At the end of training, you have a model that will give you a similarity score between an image and a text

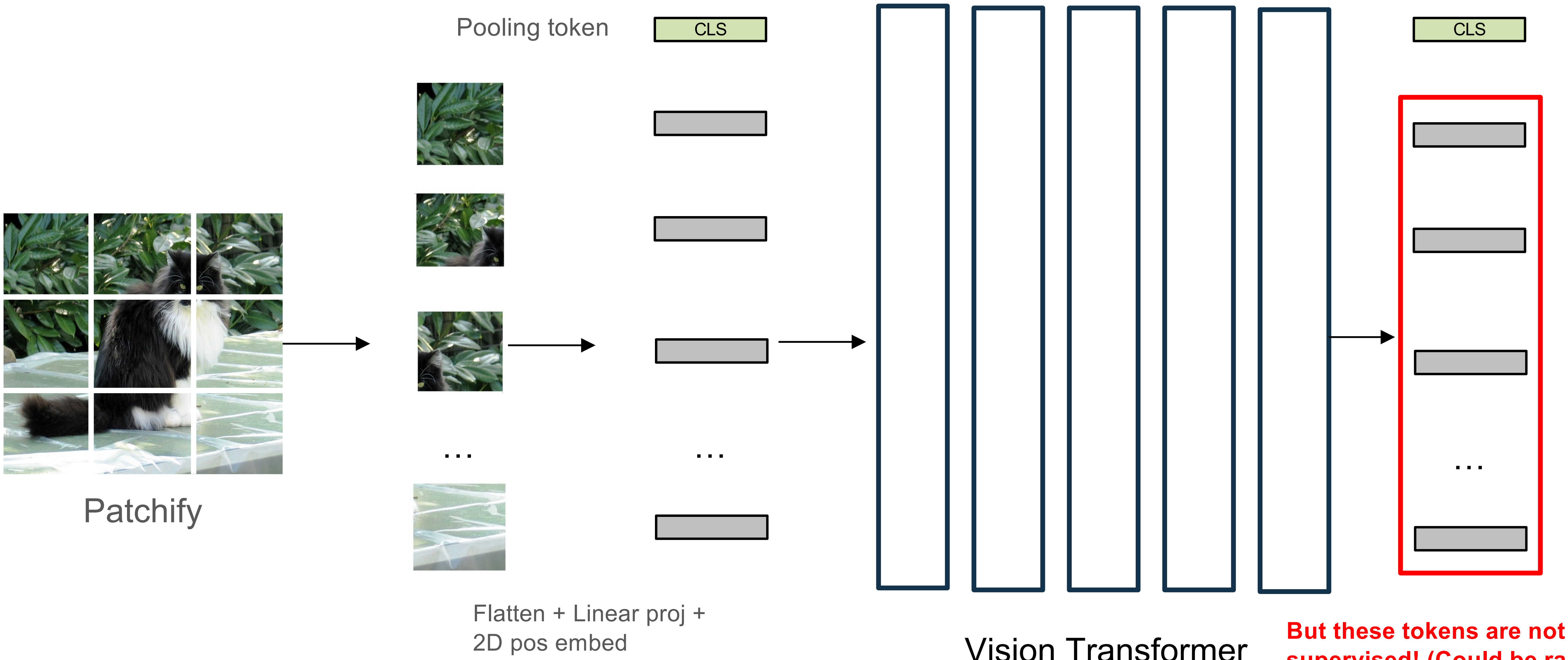
What features should we use from CLIP?

Extract Image Features from CLS token for contrastive learning



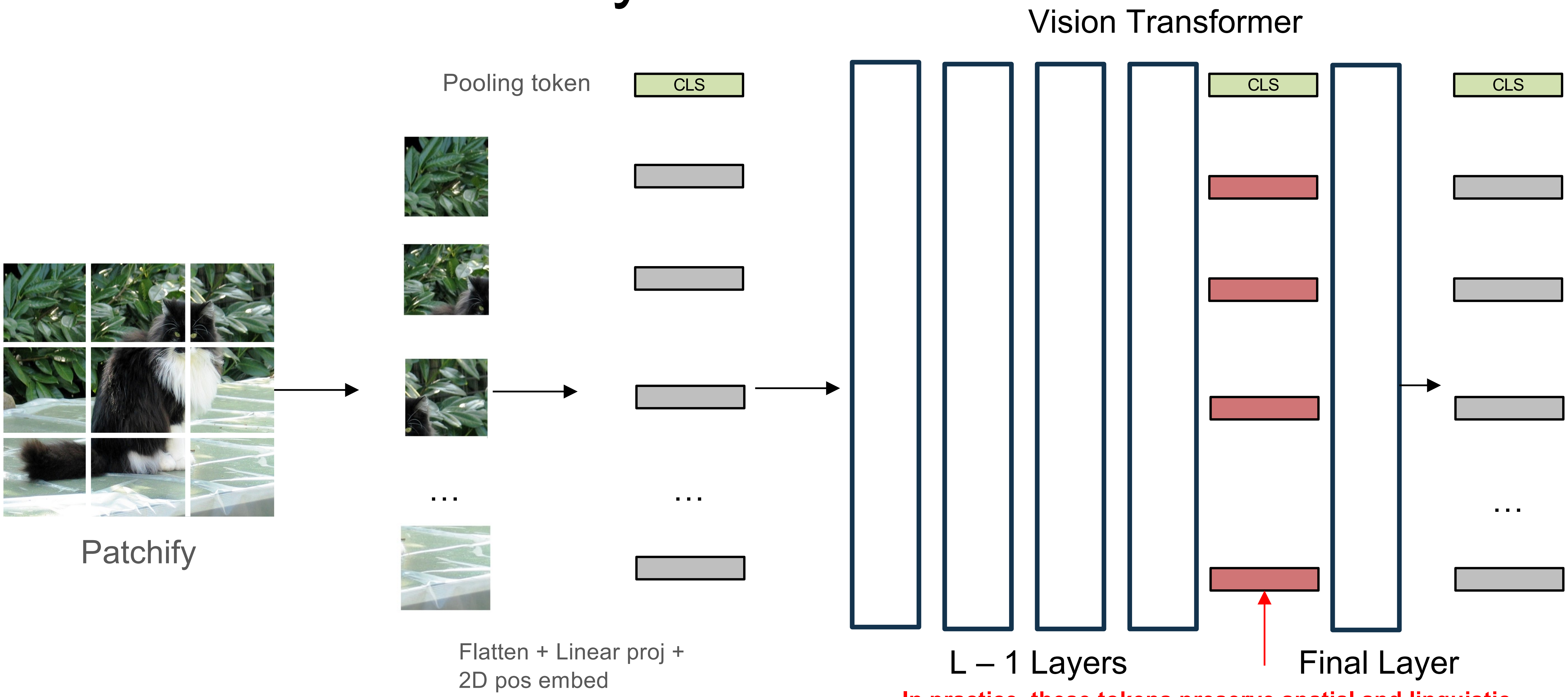
[Image source]

What features should we use from CLIP?



But these tokens are not supervised! (Could be random and loss will not change)

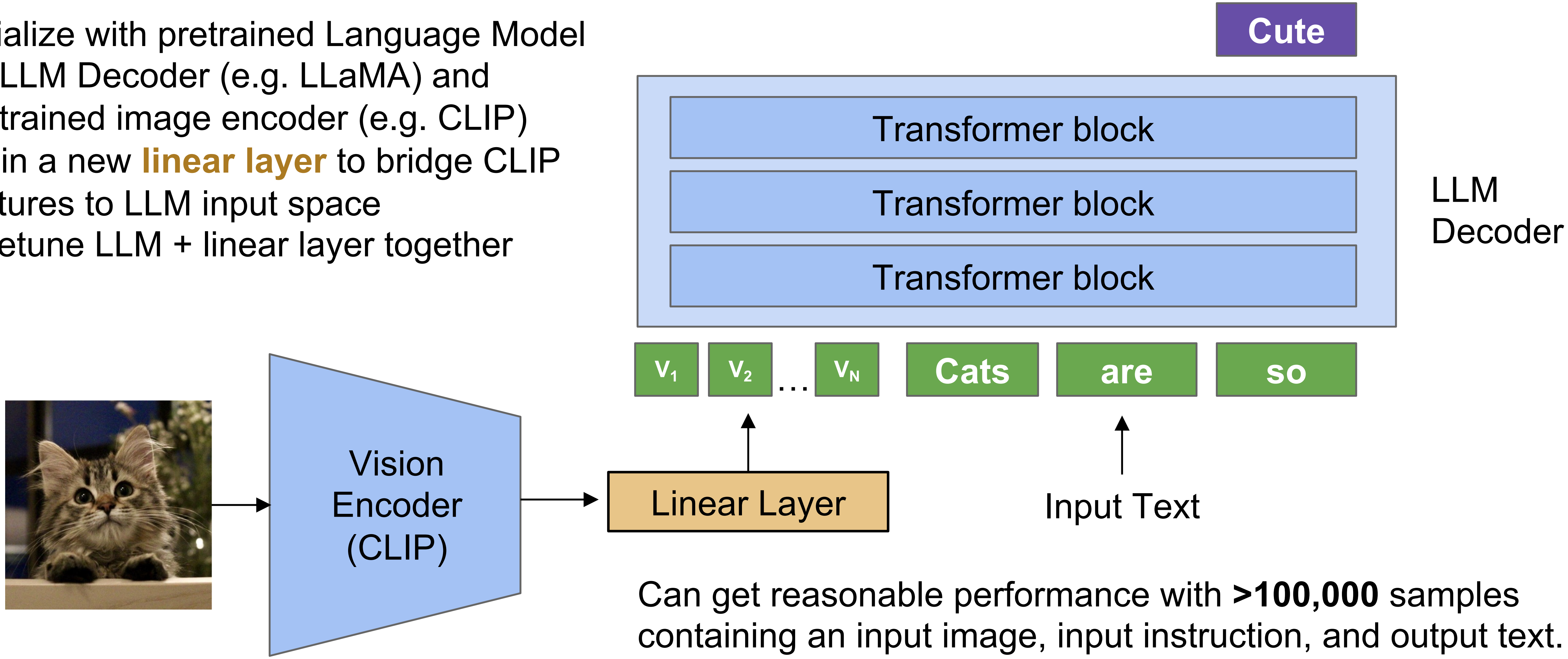
Use Penultimate Layer!



In practice, these tokens preserve spatial and linguistic information best for LLMs. Can drop CLS for slight gains.

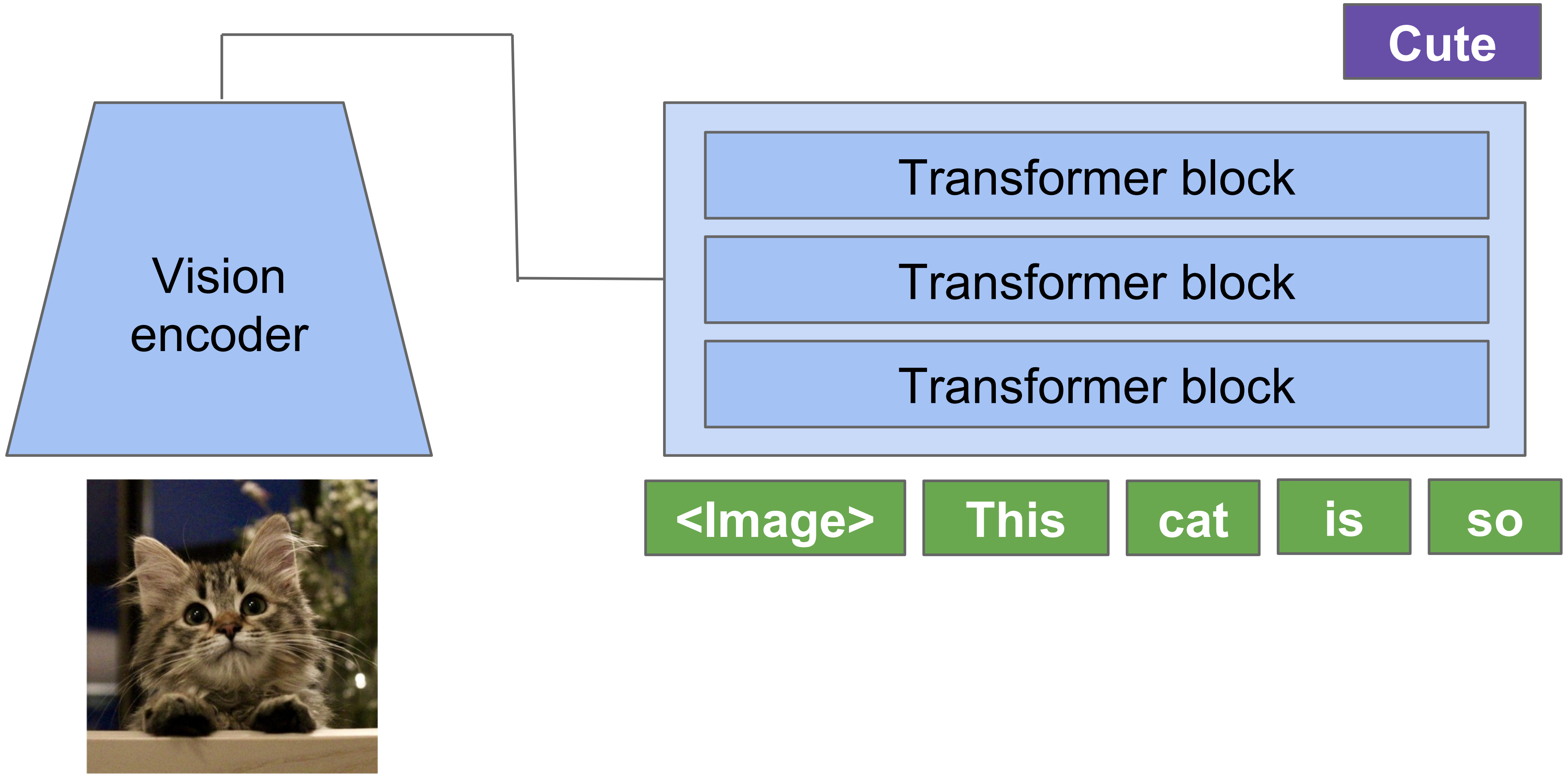
LLaVA – Overall Architecture + Training Recipe

1. Initialize with pretrained Language Model for LLM Decoder (e.g. LLaMA) and pretrained image encoder (e.g. CLIP)
2. Train a new **linear layer** to bridge CLIP features to LLM input space
3. Finetune LLM + linear layer together

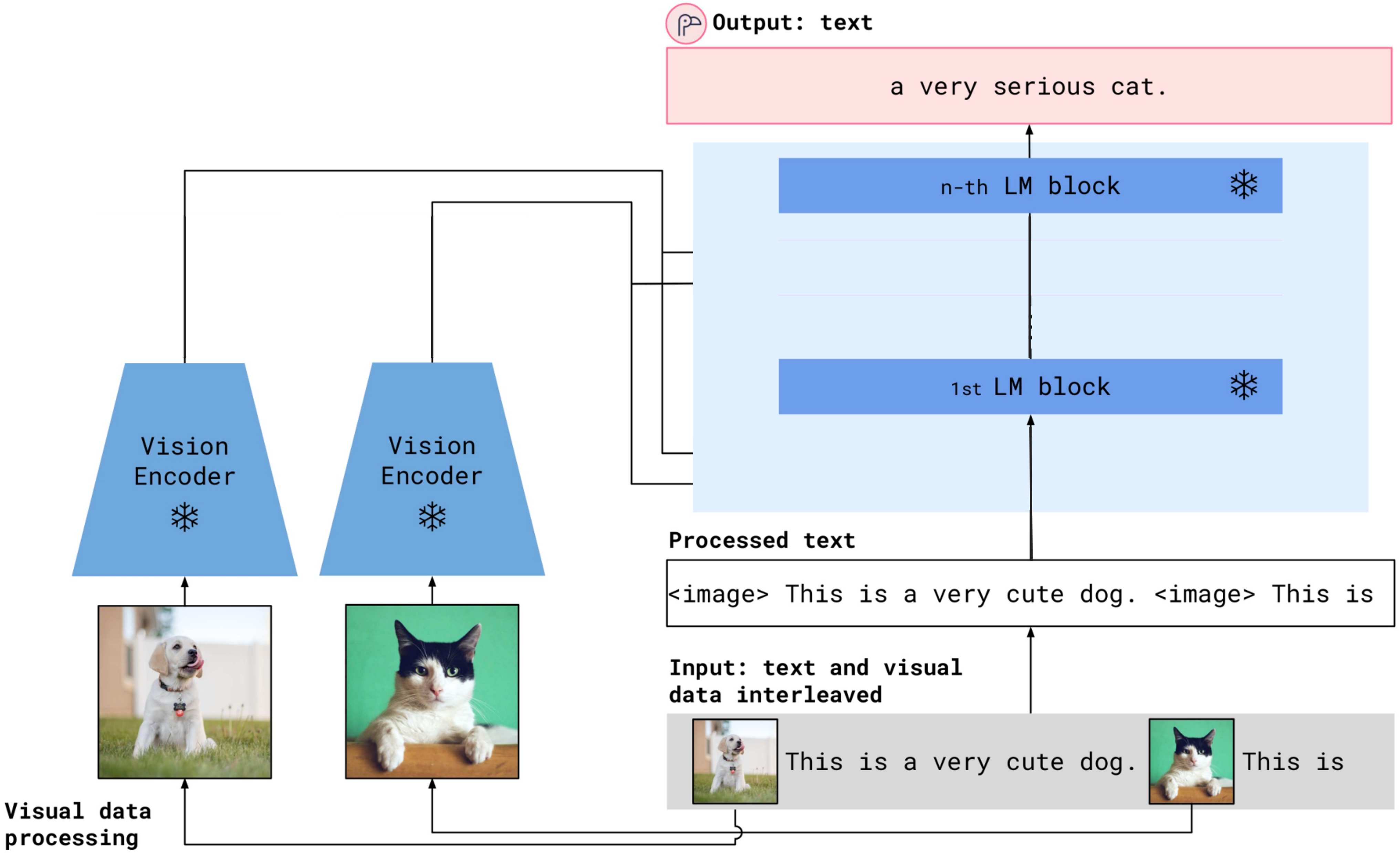


Can get reasonable performance with **>100,000** samples containing an input image, input instruction, and output text.

Flamingo followed up with a new way to fuse visual features

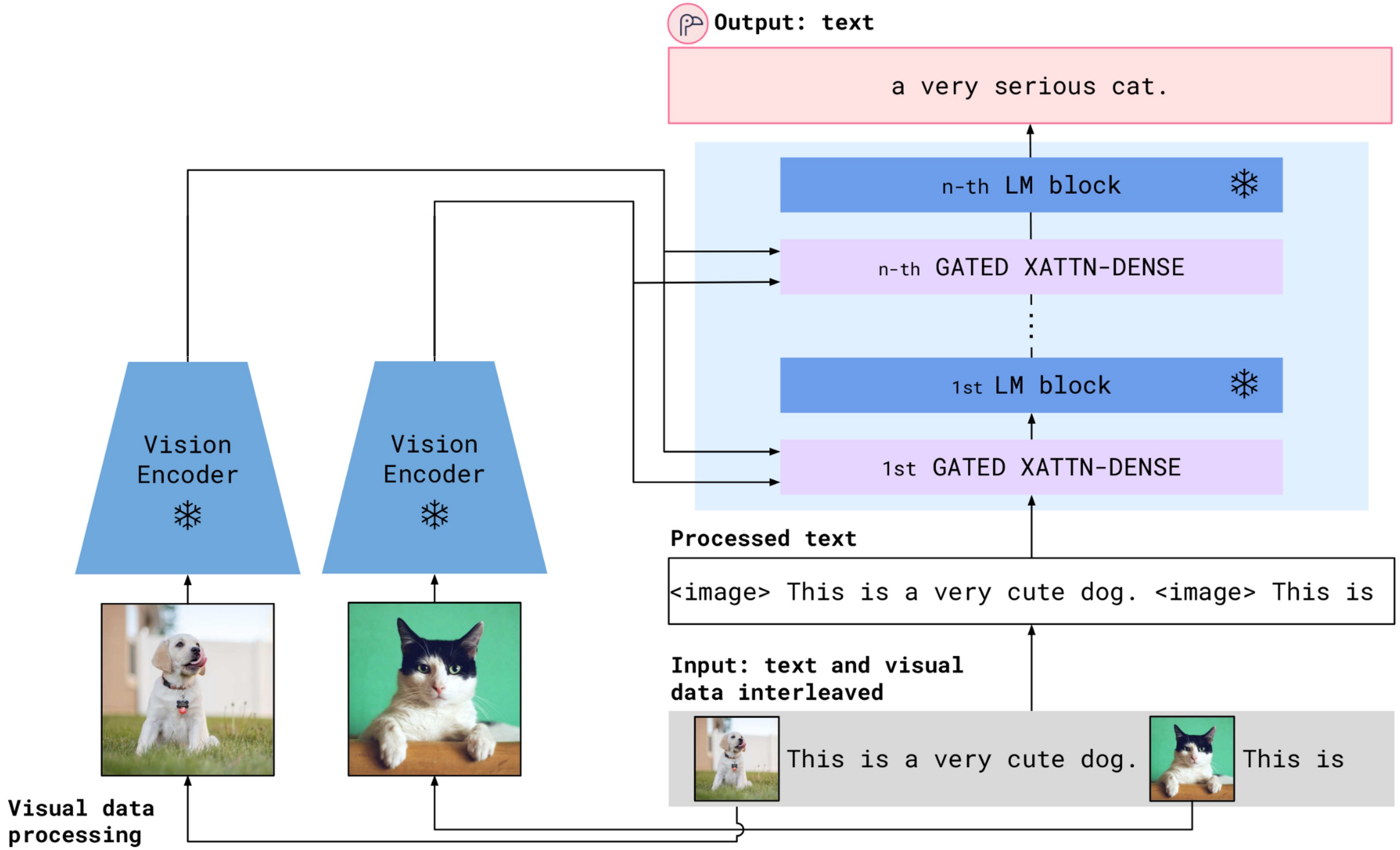


Pre-trained parts of Flamingo



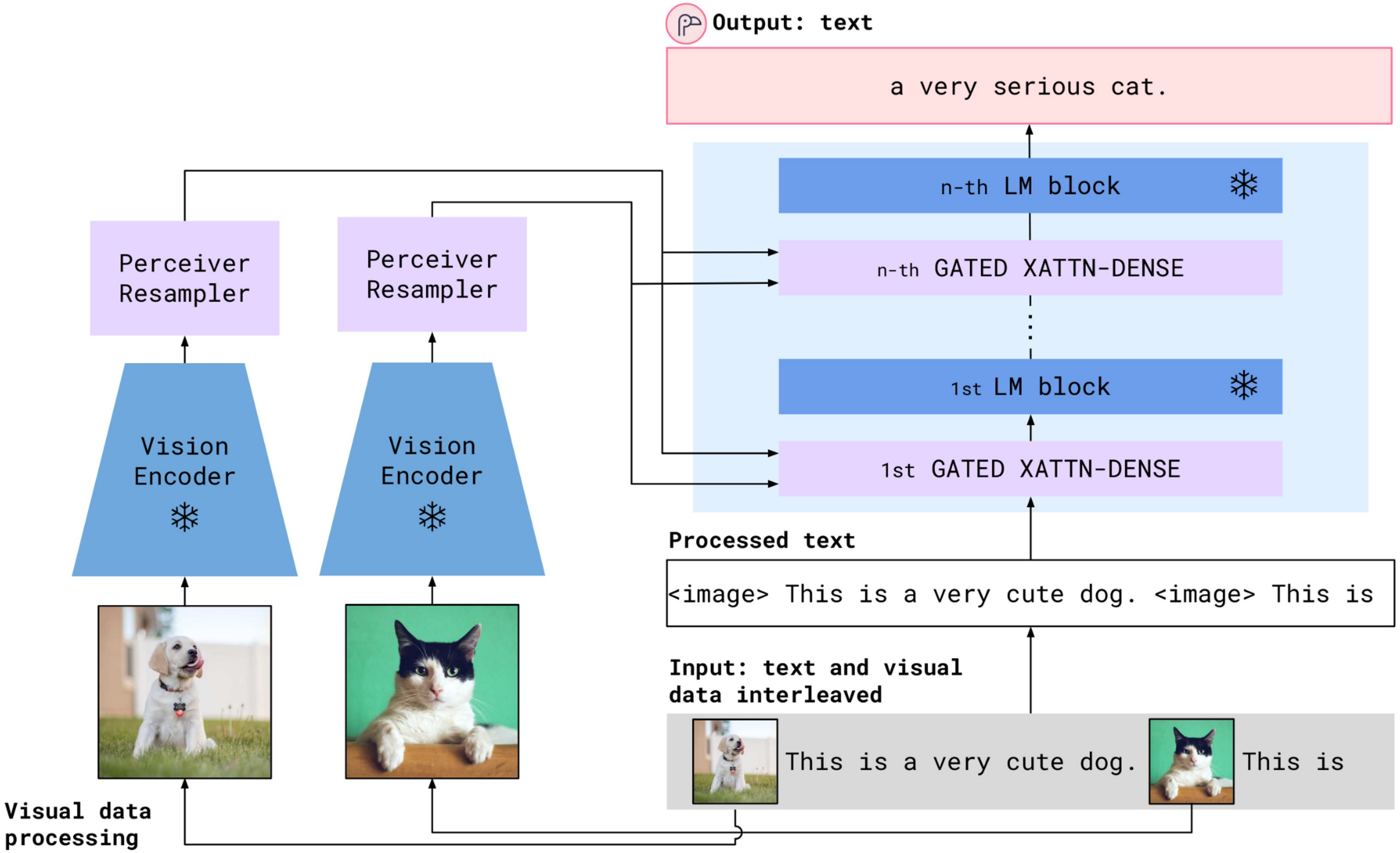
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

There are 2 learned parts in Flamingo



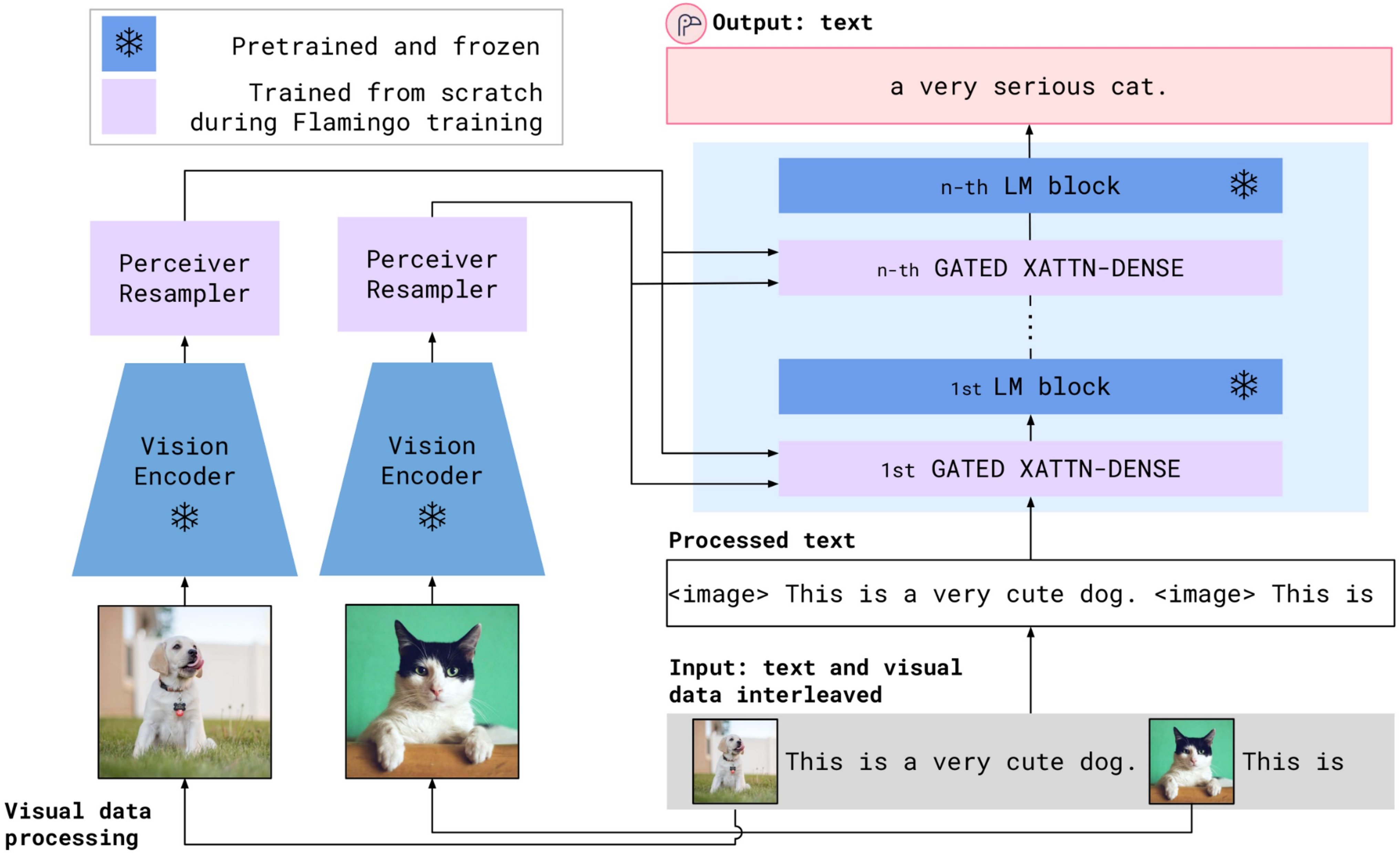
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Perceiver sampler converts variable sized image tokens to fixed sized ones



Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

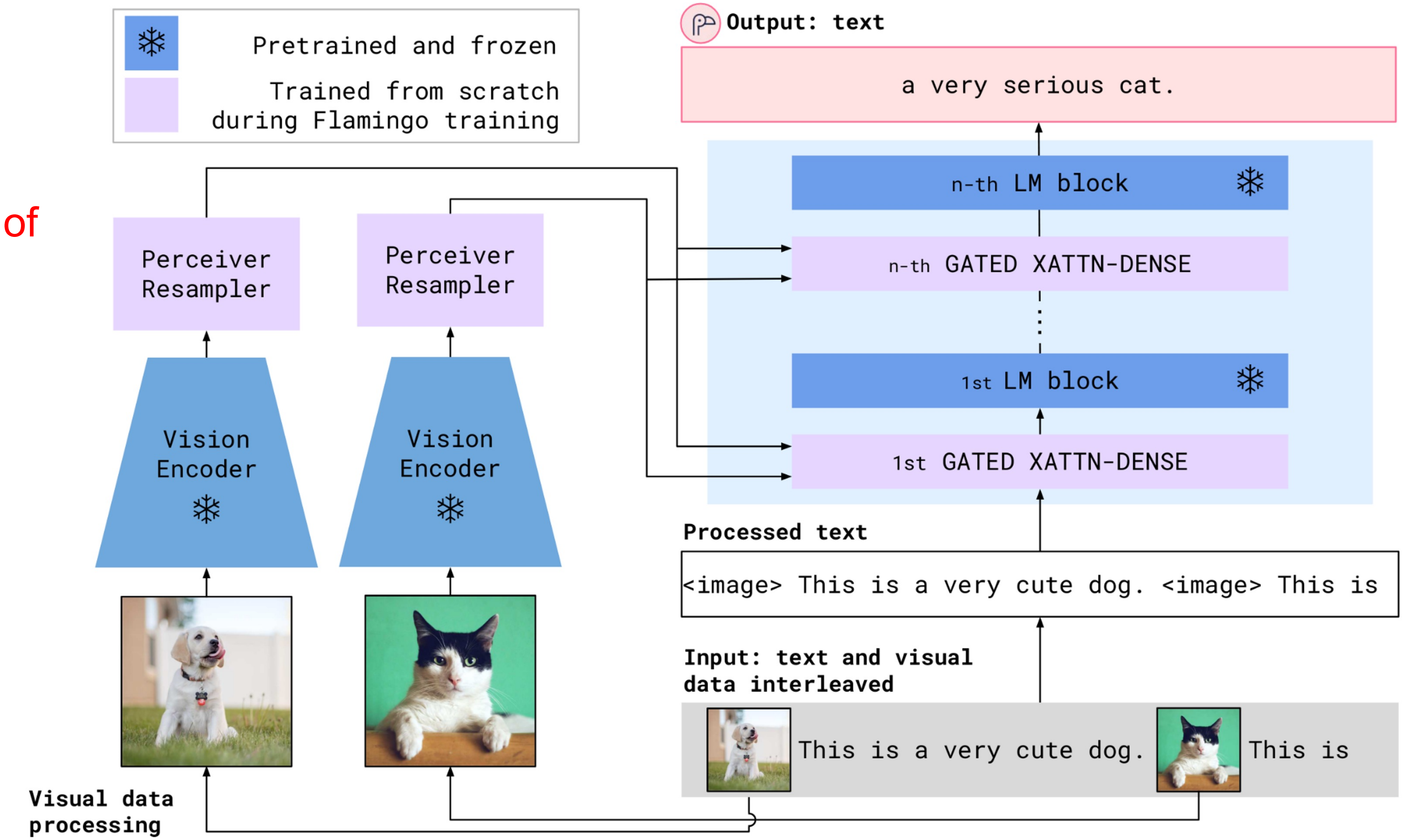
Flamingo full architecture



Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo full architecture

Learned method of down-sampling image/video representations



❄ Pretrained and frozen

▨ Trained from scratch during Flamingo training

Output: text

a very serious cat.

n-th LM block ❄

n-th GATED XATTN-DENSE

1st LM block ❄

1st GATED XATTN-DENSE

Perceiver Resampler

Perceiver Resampler

Vision Encoder ❄

Vision Encoder ❄

Processed text

<image> This is a very cute dog. <image> This is

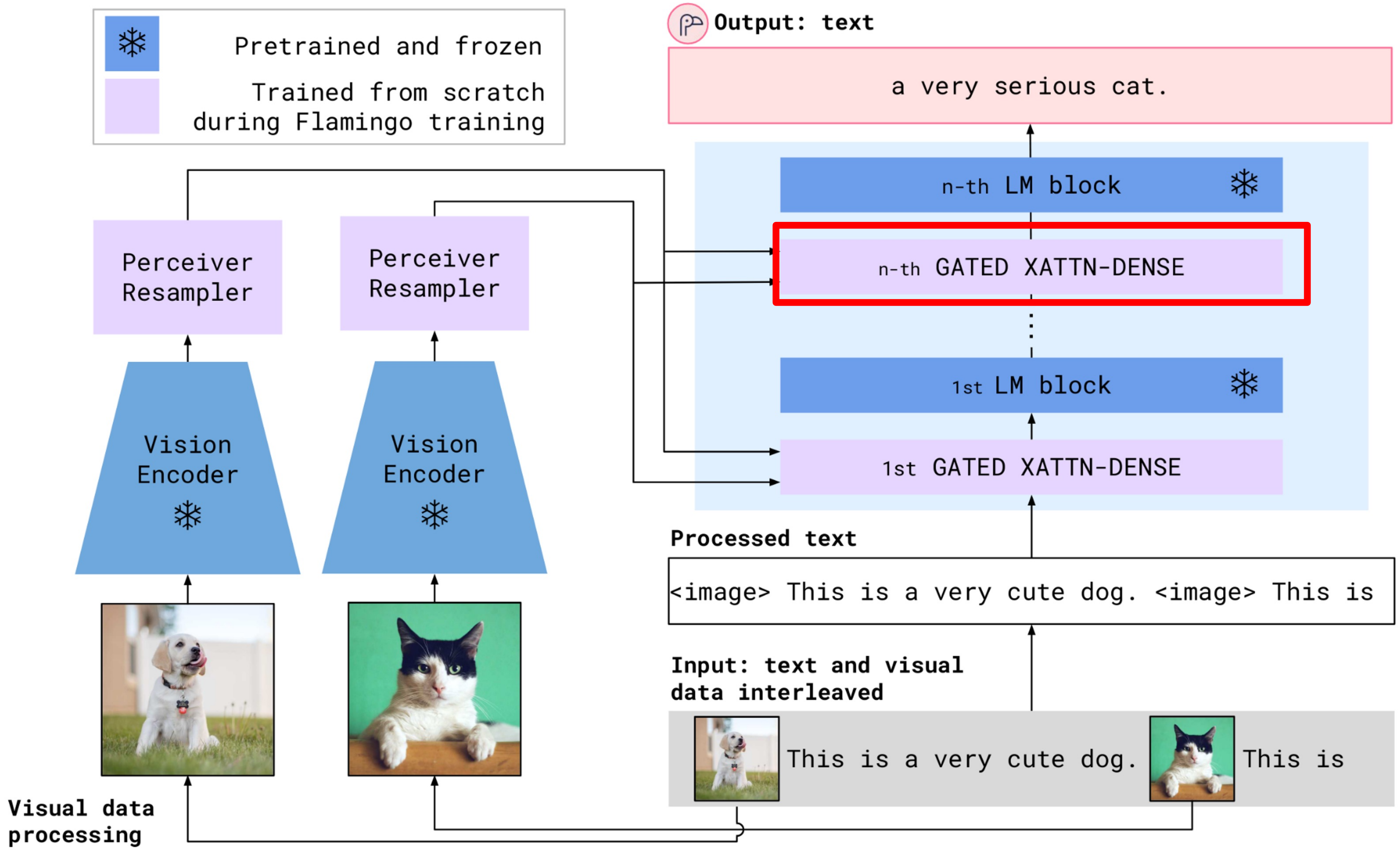
Input: text and visual data interleaved

This is a very cute dog. This is

Visual data processing

Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo full architecture



❄ Pretrained and frozen

Trained from scratch during Flamingo training

Output: text

a very serious cat.

n-th LM block ❄

n-th GATED XATTN-DENSE

1st LM block ❄

1st GATED XATTN-DENSE

Perceiver Resampler

Perceiver Resampler

Vision Encoder ❄

Vision Encoder ❄

Processed text

<image> This is a very cute dog. <image> This is

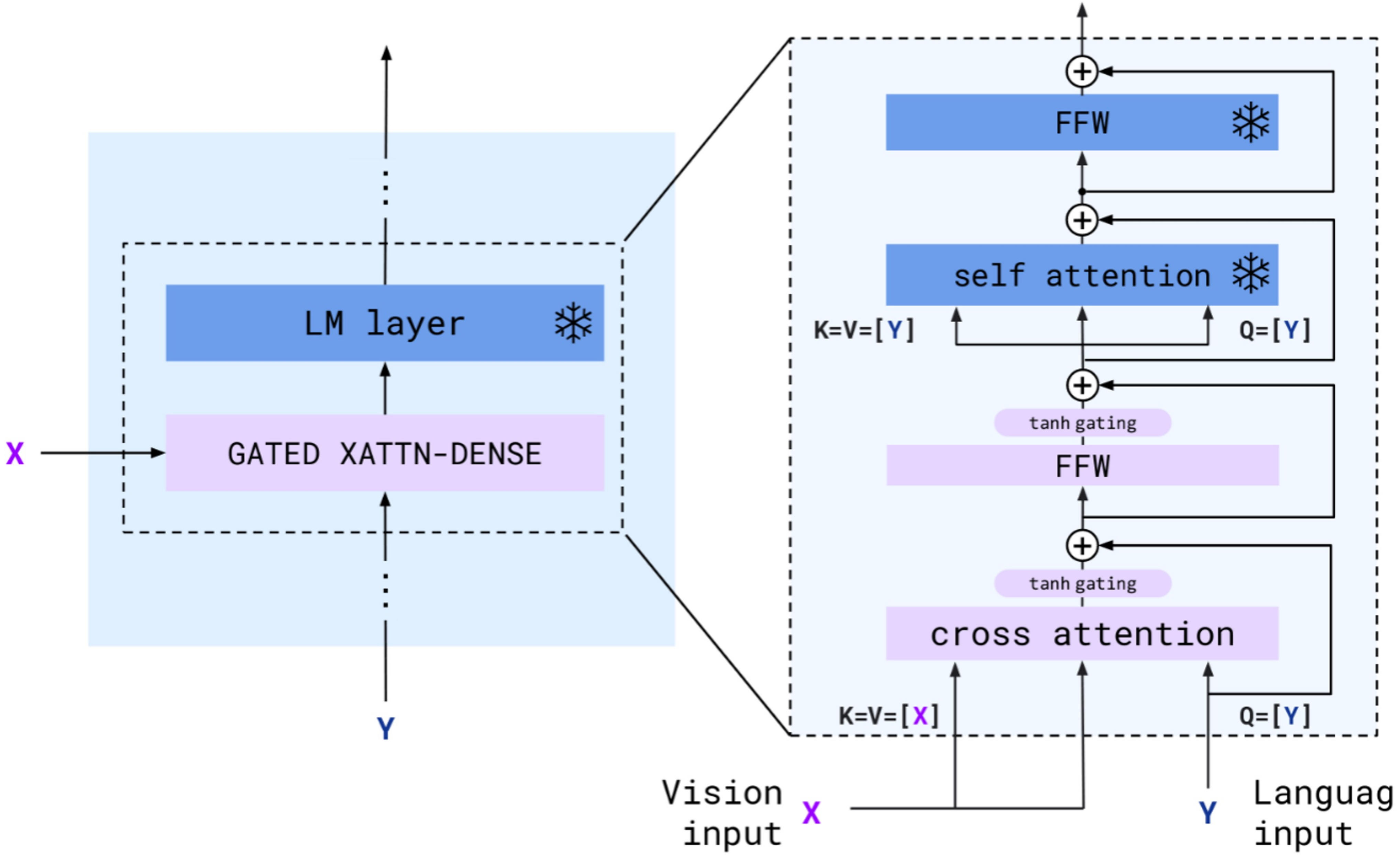
Input: text and visual data interleaved

This is a very cute dog. This is

Visual data processing

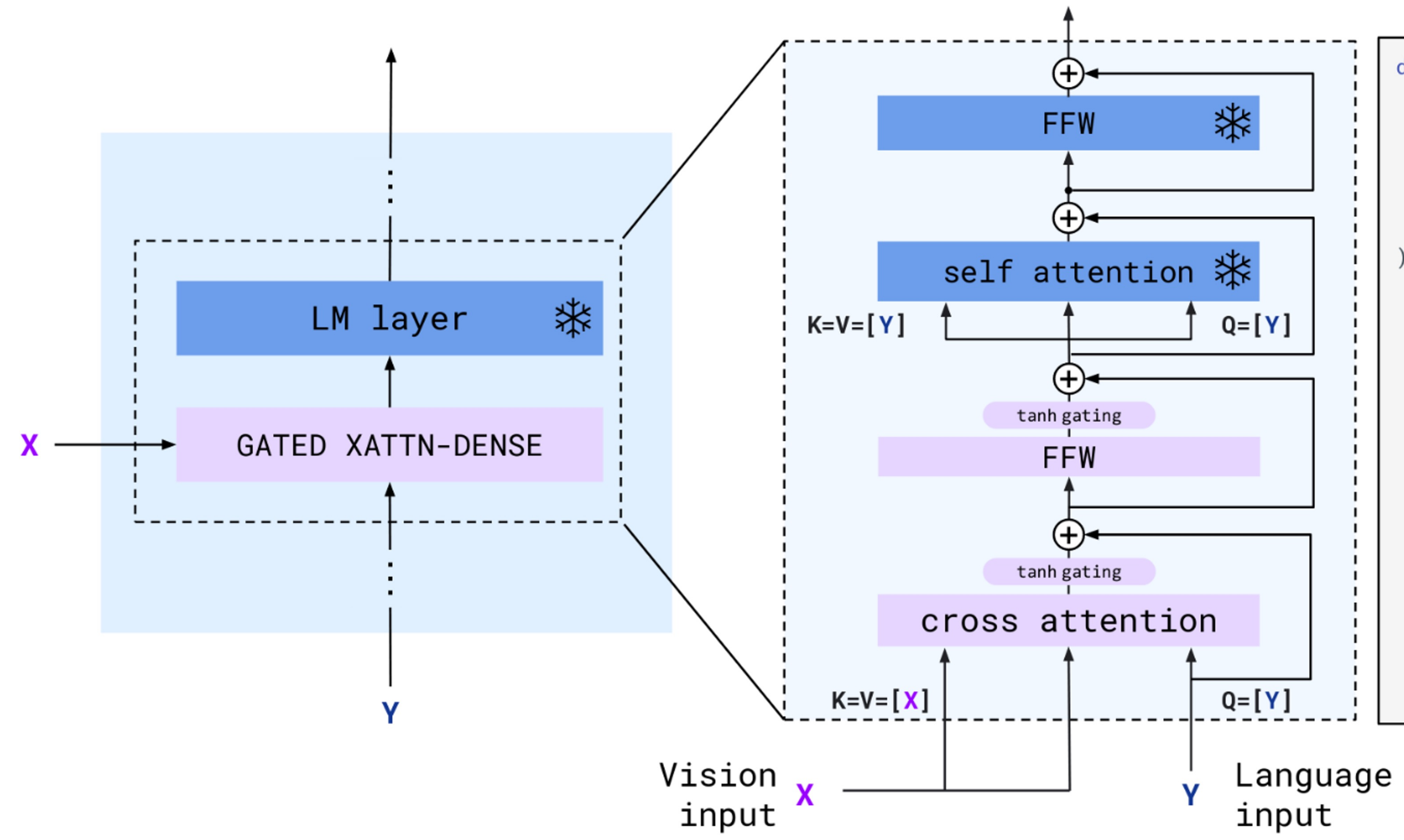
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo gated cross-attention



Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo gated cross-attention

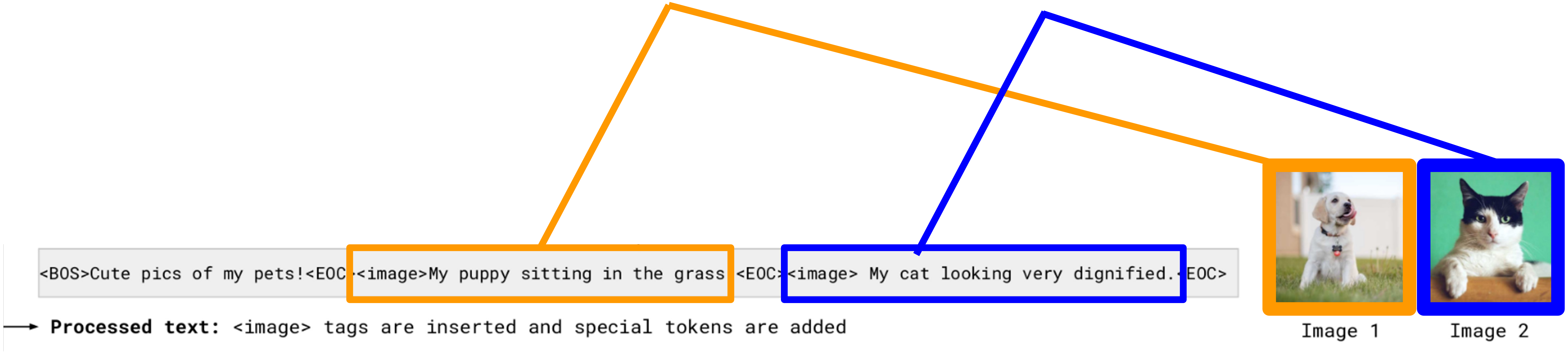


```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter – init at 0.
    alpha_dense, # ffw gating parameter – init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 1. Gated Cross Attention
  y = y + tanh(alpha_xattn) * attention(q=y, kv=x)
  # 2. Gated Feed Forward (dense) Layer
  y = y + tanh(alpha_dense) * ffw(y)

  # Regular self-attention + FFW on language
  y = y + frozen_attention(q=y, kv=y)
  y = y + frozen_ffw(y)
  return y  # output visually informed language features
```

Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo arranges its training data similar to language modeling,

with special tags <image>, <eos> to indicate when a new image shows up or the text ends.

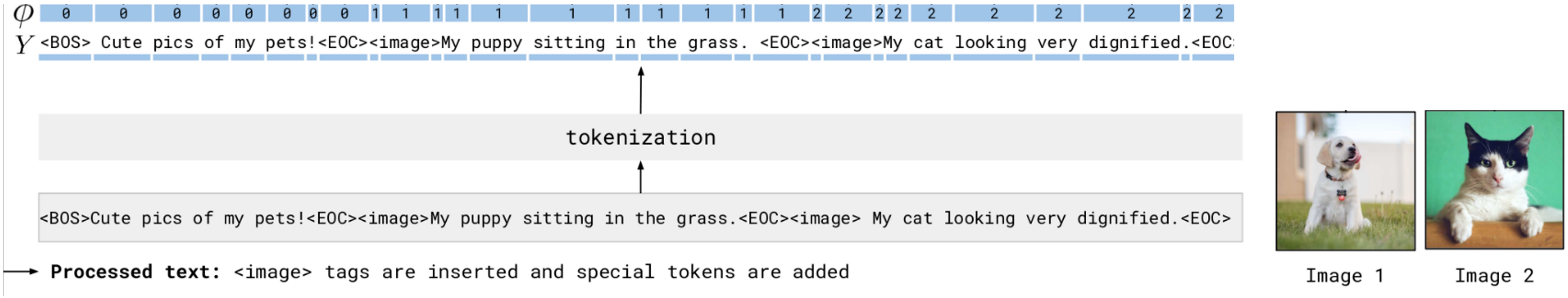


<BOS>Cute pics of my pets!<EOC><image>My puppy sitting in the grass<EOC><image> My cat looking very dignified.<EOC>

→ **Processed text:** <image> tags are inserted and special tokens are added

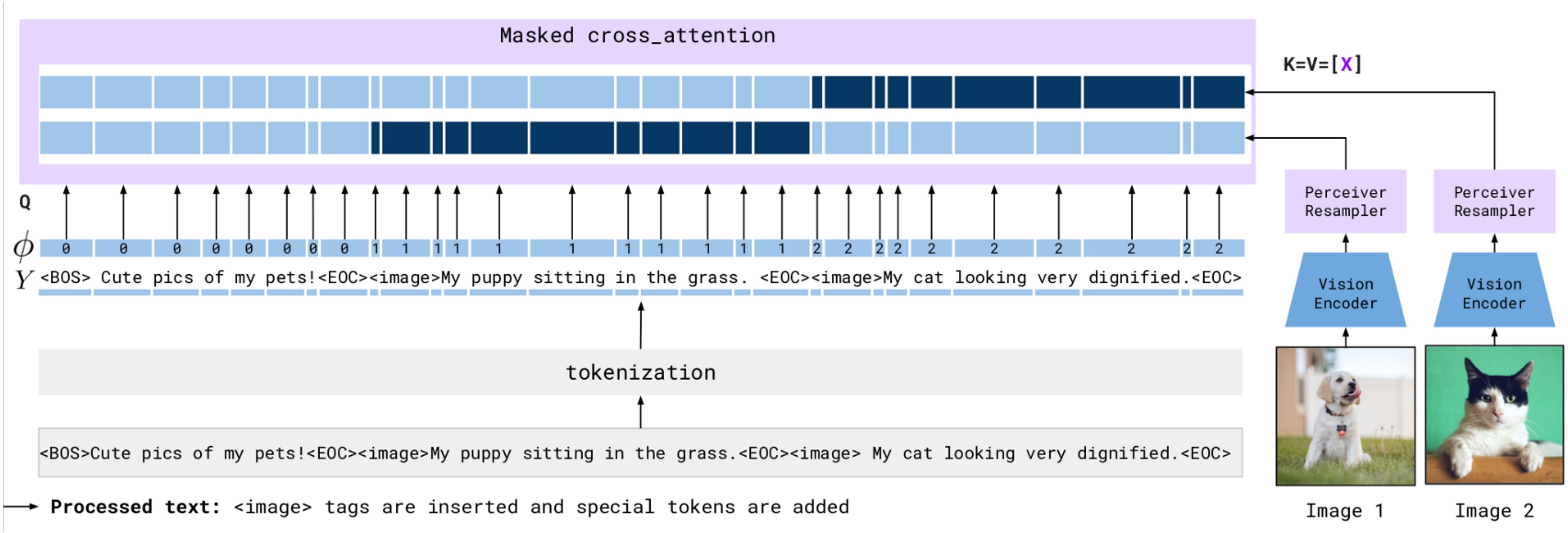
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo masked attention


ϕ
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
Y
<BOS> Cute pics of my pets!<EOC><image>My puppy sitting in the grass. <EOC><image>My cat looking very dignified.<EOC
tokenization
<BOS>Cute pics of my pets!<EOC><image>My puppy sitting in the grass.<EOC><image> My cat looking very dignified.<EOC>
Processed text: <image> tags are inserted and special tokens are added
Image 1
Image 2

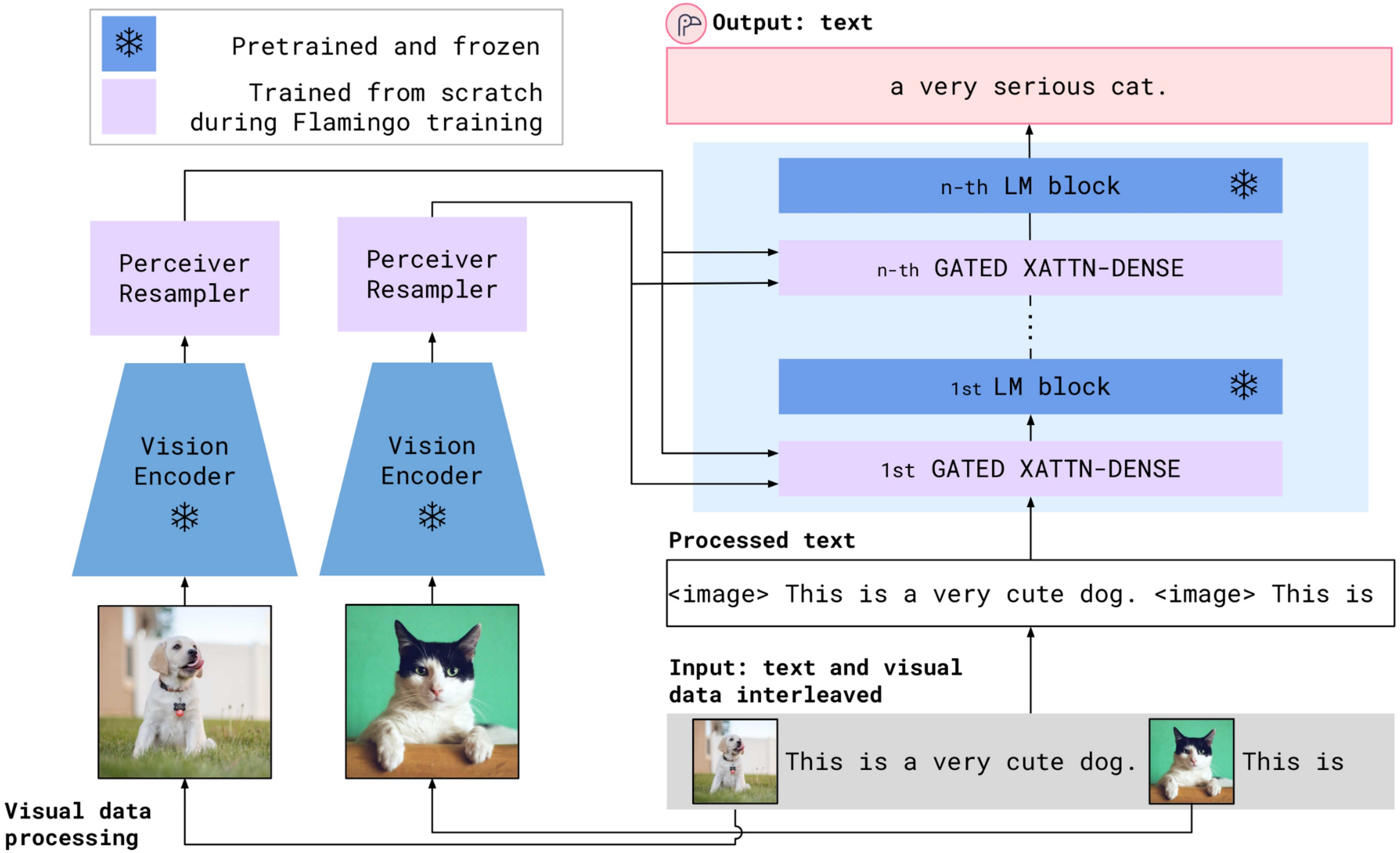
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo masked attention


Masked cross_attention
K=V=[X]
Q
ϕ
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
Y
<BOS> Cute pics of my pets!<EOC><image>My puppy sitting in the grass. <EOC><image>My cat looking very dignified.<EOC>
tokenization
<BOS>Cute pics of my pets!<EOC><image>My puppy sitting in the grass.<EOC><image> My cat looking very dignified.<EOC>
Processed text: <image> tags are inserted and special tokens are added
Perceiver Resampler
Perceiver Resampler
Vision Encoder
Vision Encoder
Image 1
Image 2

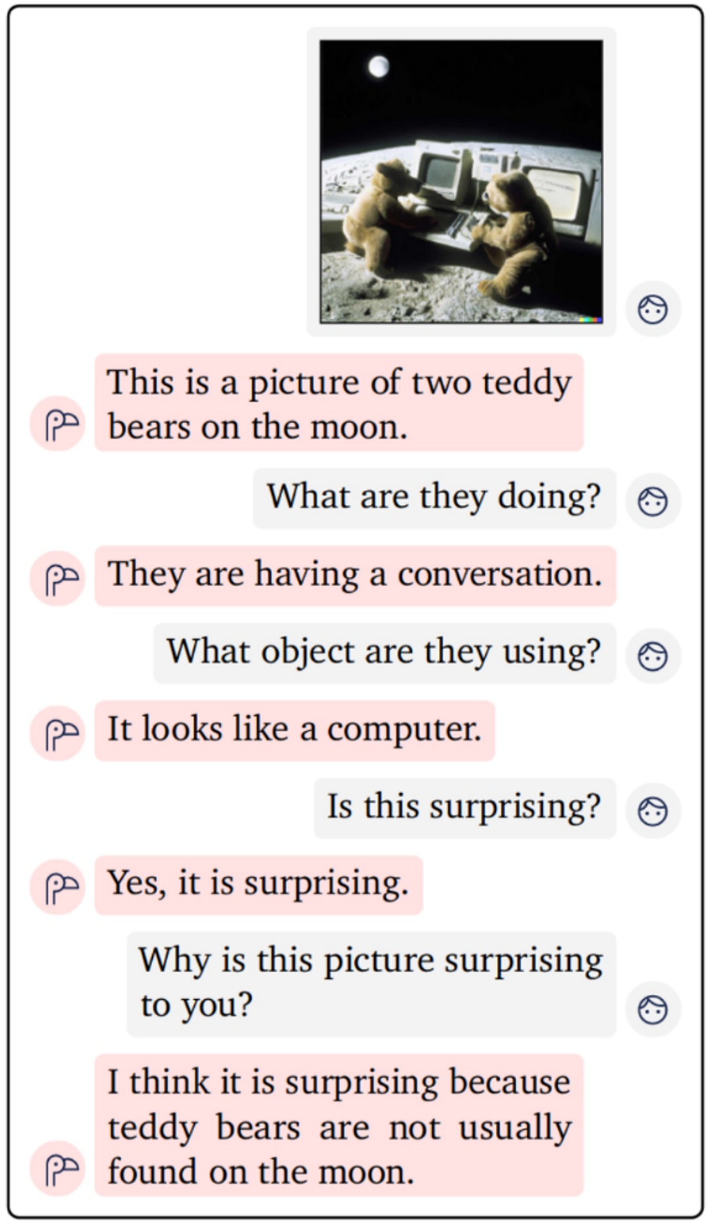
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo full architecture



Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo results



This is a picture of two teddy bears on the moon.

What are they doing?

They are having a conversation.

What object are they using?

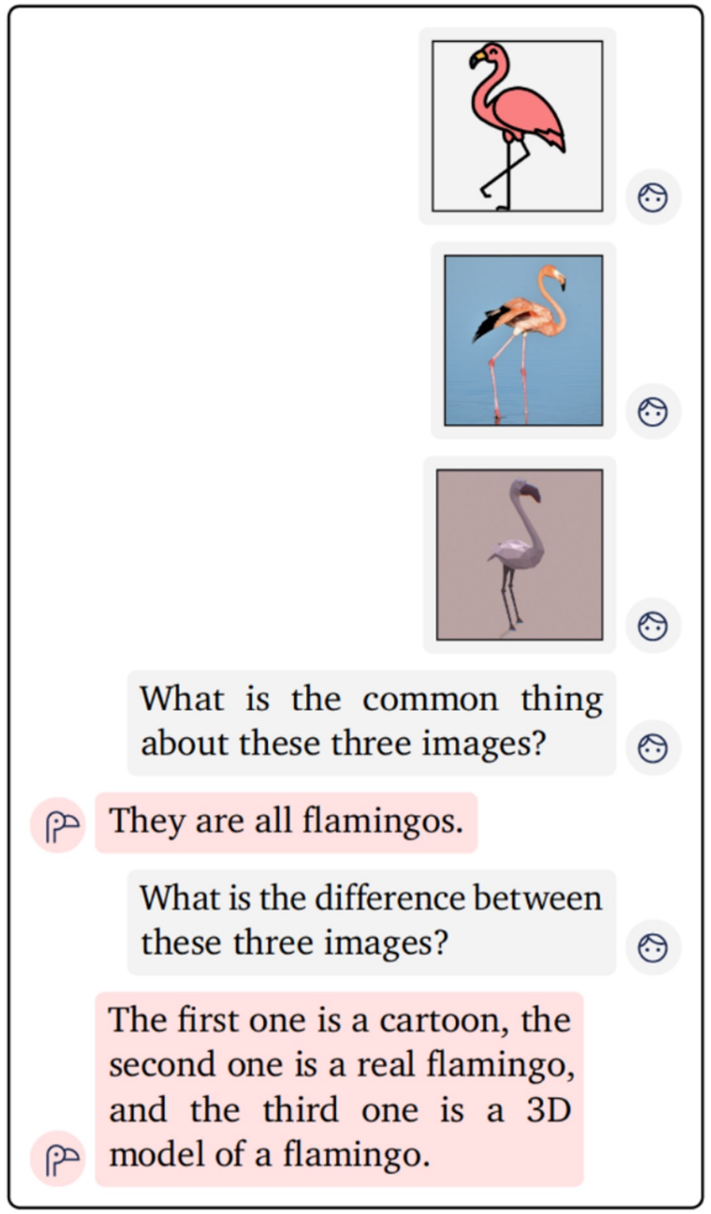
It looks like a computer.

Is this surprising?

Yes, it is surprising.

Why is this picture surprising to you?

I think it is surprising because teddy bears are not usually found on the moon.

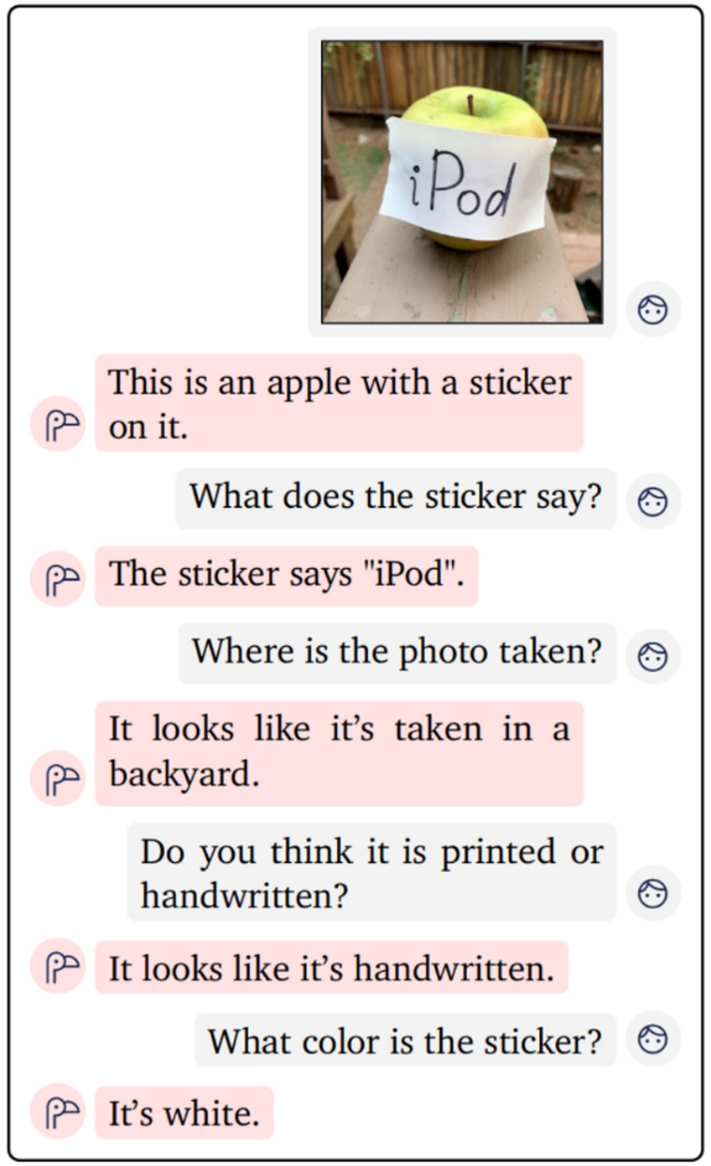


What is the common thing about these three images?

They are all flamingos.

What is the difference between these three images?

The first one is a cartoon, the second one is a real flamingo, and the third one is a 3D model of a flamingo.



This is an apple with a sticker on it.

What does the sticker say?

The sticker says "iPod".

Where is the photo taken?

It looks like it's taken in a backyard.

Do you think it is printed or handwritten?

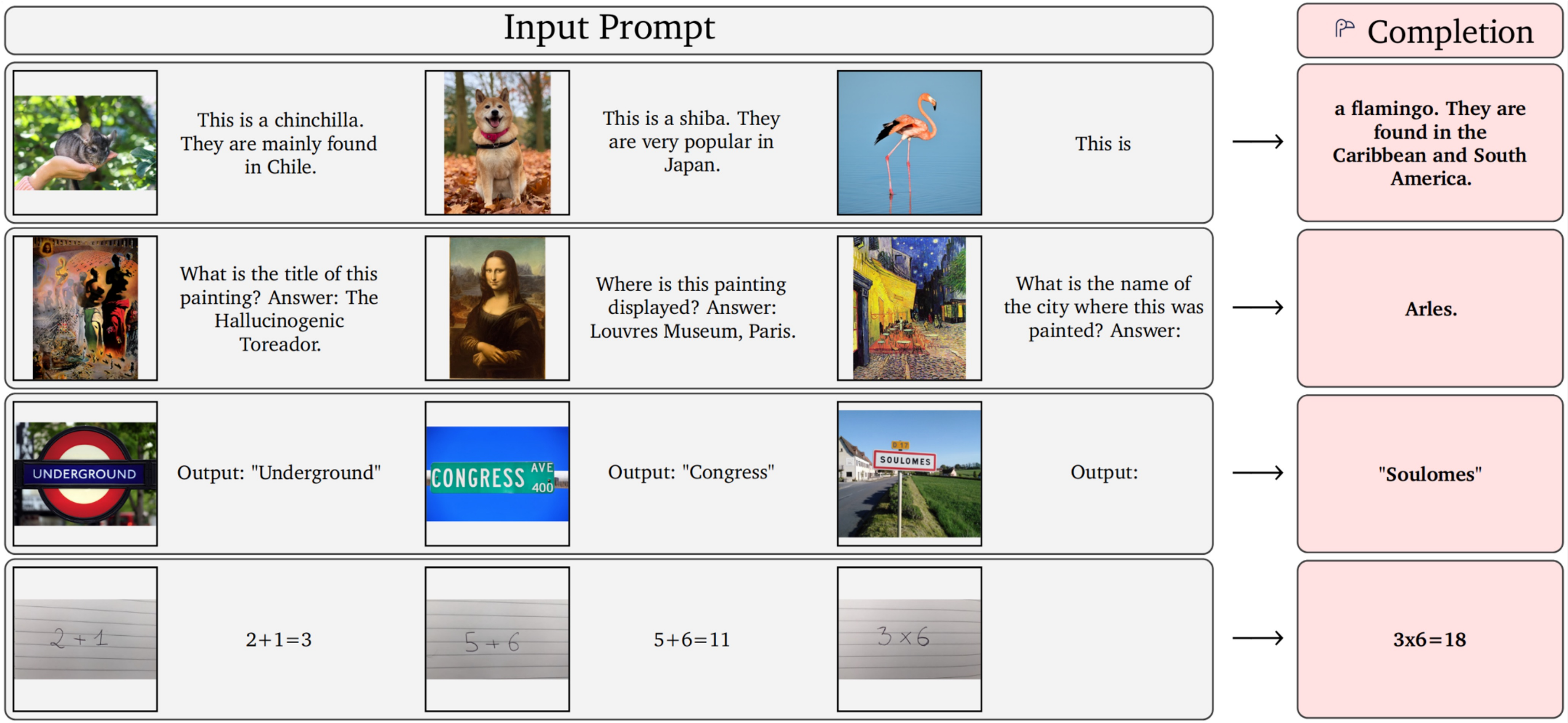
It looks like it's handwritten.

What color is the sticker?

It's white.

Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

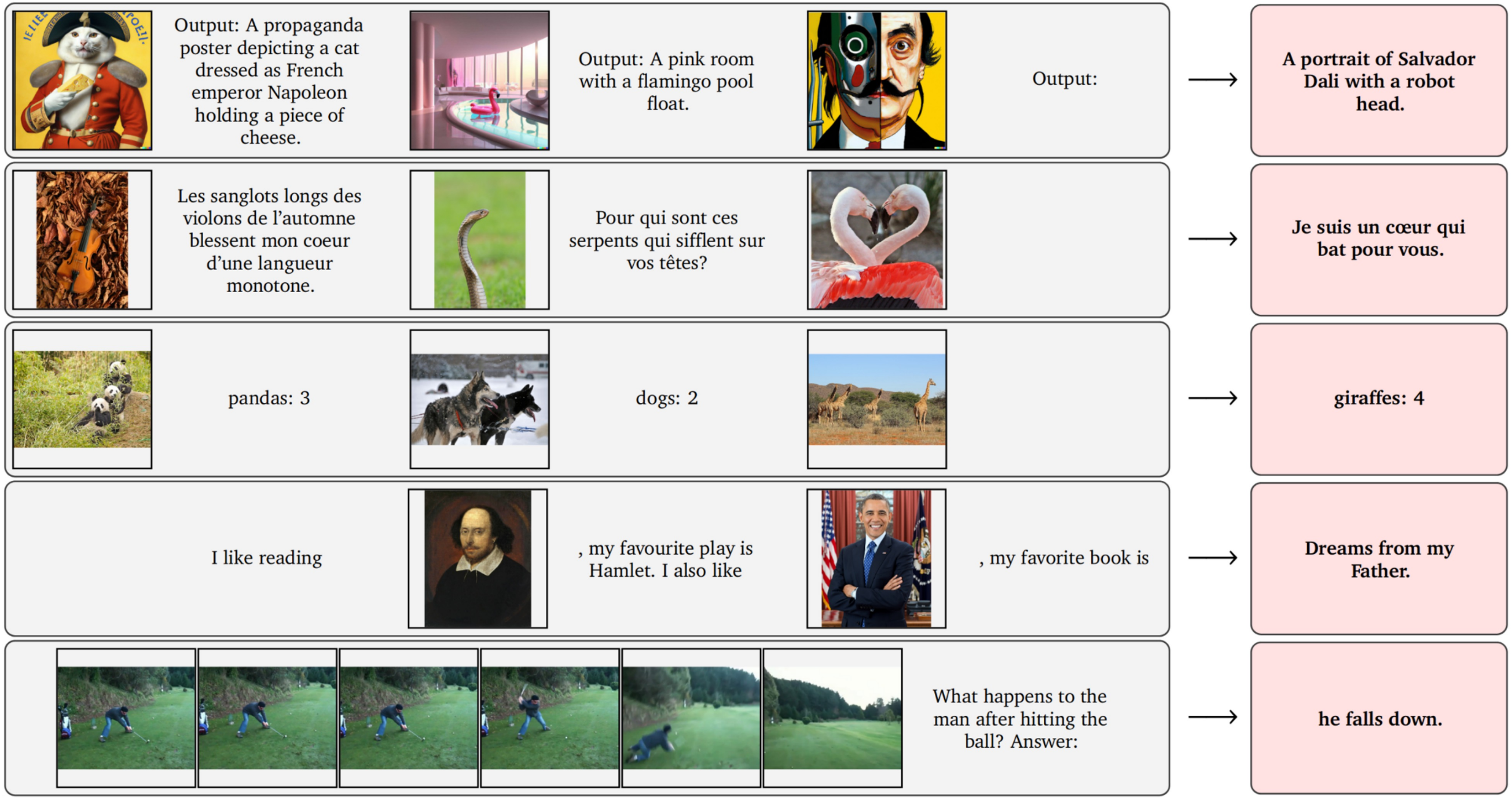
Flaming enables in-context learning



Input Prompt				Completion
This is a chinchilla. They are mainly found in Chile.	This is a shiba. They are very popular in Japan.	This is	→	a flamingo. They are found in the Caribbean and South America.
What is the title of this painting? Answer: The Hallucinogenic Toreador.	Where is this painting displayed? Answer: Louvres Museum, Paris.	What is the name of the city where this was painted? Answer:	→	Arles.
Output: "Underground"	Output: "Congress"	Output:	→	"Soulomes"
2+1=3	5+6=11		→	3x6=18

Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo results



Output: A propaganda poster depicting a cat dressed as French emperor Napoleon holding a piece of cheese. Output: A pink room with a flamingo pool float. Output: → **A portrait of Salvador Dali with a robot head.**

Les sanglots longs des violons de l'automne blessent mon coeur d'une langueur monotone. Pour qui sont ces serpents qui sifflent sur vos têtes? → **Je suis un cœur qui bat pour vous.**

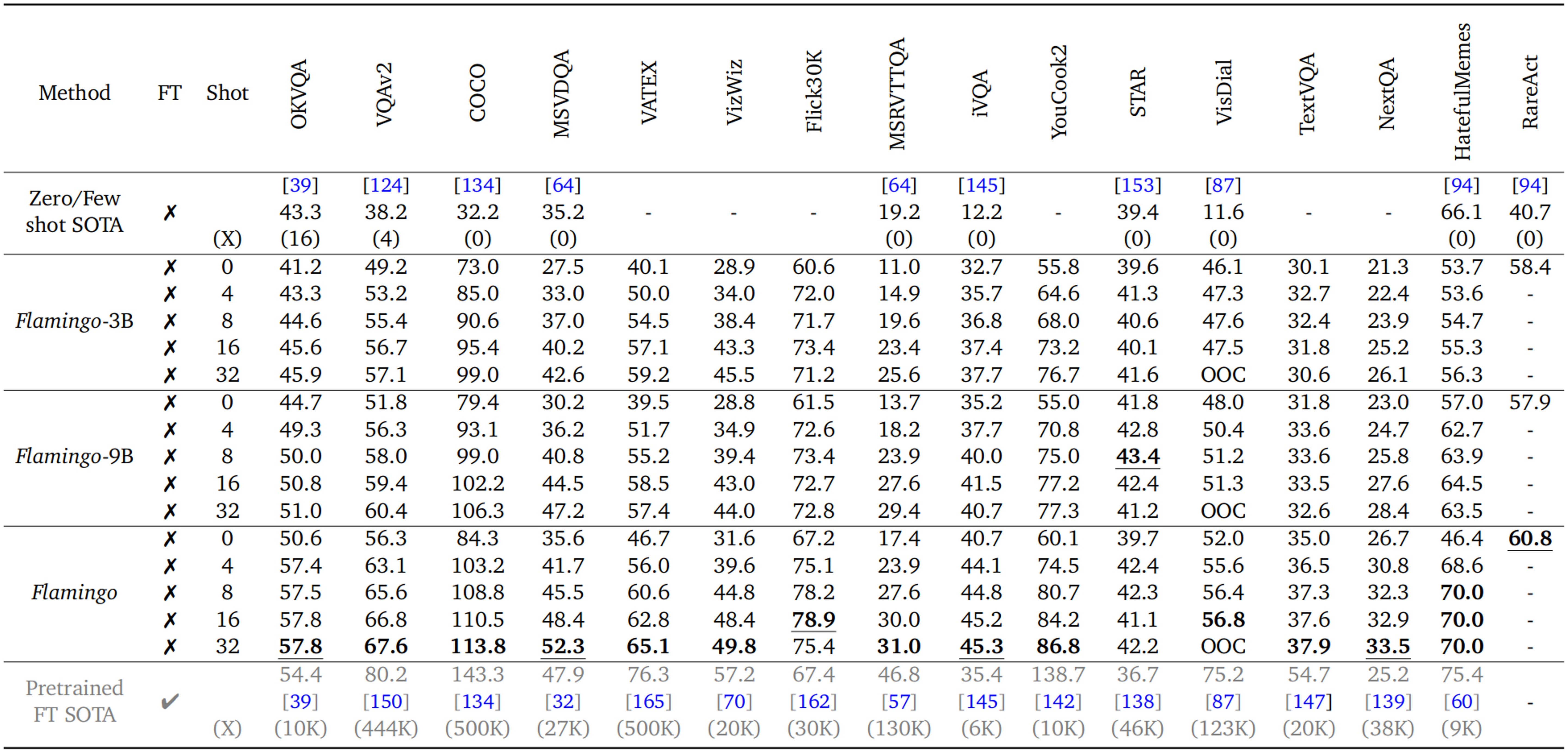
pandas: 3 dogs: 2 → **giraffes: 4**

I like reading , my favourite play is Hamlet. I also like , my favorite book is → **Dreams from my Father.**

What happens to the man after hitting the ball? Answer: → **he falls down.**

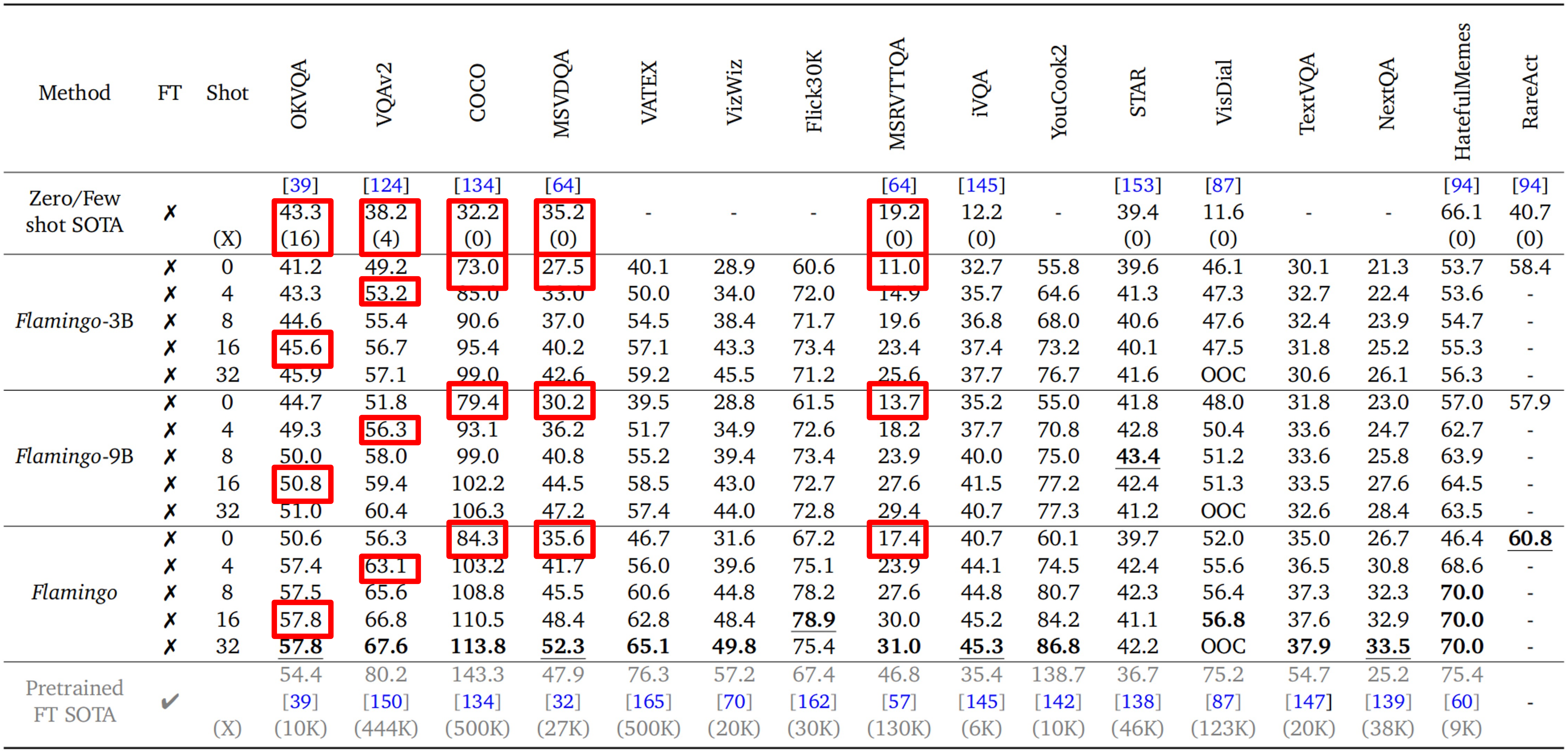
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Results: zero & few shot



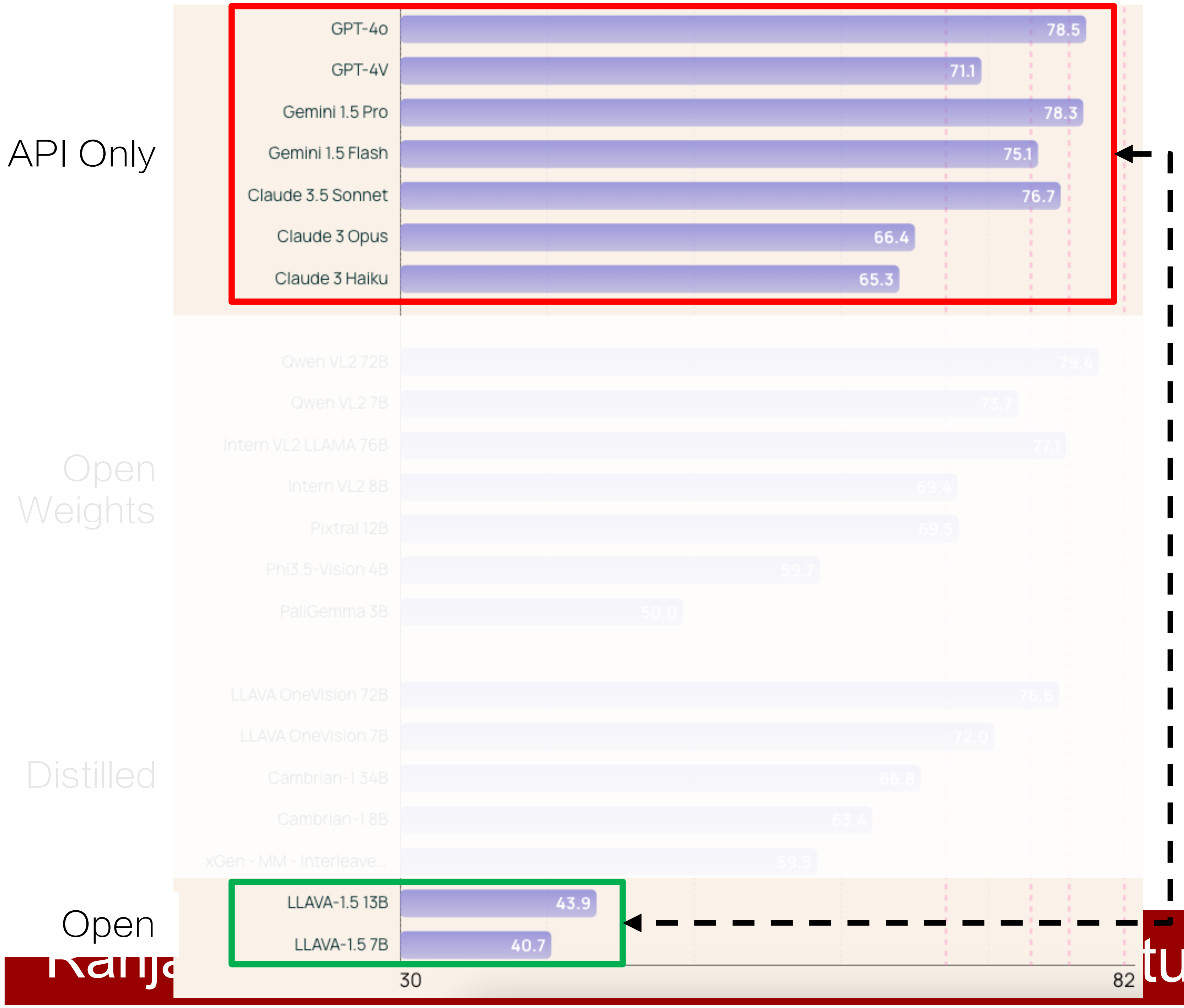
Method	FT	Shot	OKVQA	VQAv2	COCO	MSVDQA	VATEX	VizWiz	Flick30K	MSRVTTQA	iVQA	YouCook2	STAR	VisDial	TextVQA	NextQA	HatefulMemes	RareAct
Zero/Few shot SOTA	✗	(X)	[39] 43.3 (16)	[124] 38.2 (4)	[134] 32.2 (0)	[64] 35.2 (0)	-	-	-	[64] 19.2 (0)	[145] 12.2 (0)	-	[153] 39.4 (0)	[87] 11.6 (0)	-	-	[94] 66.1 (0)	[94] 40.7 (0)
Flamingo-3B	✗	0	41.2	49.2	73.0	27.5	40.1	28.9	60.6	11.0	32.7	55.8	39.6	46.1	30.1	21.3	53.7	58.4
	✗	4	43.3	53.2	85.0	33.0	50.0	34.0	72.0	14.9	35.7	64.6	41.3	47.3	32.7	22.4	53.6	-
	✗	8	44.6	55.4	90.6	37.0	54.5	38.4	71.7	19.6	36.8	68.0	40.6	47.6	32.4	23.9	54.7	-
	✗	16	45.6	56.7	95.4	40.2	57.1	43.3	73.4	23.4	37.4	73.2	40.1	47.5	31.8	25.2	55.3	-
	✗	32	45.9	57.1	99.0	42.6	59.2	45.5	71.2	25.6	37.7	76.7	41.6	OOC	30.6	26.1	56.3	-
Flamingo-9B	✗	0	44.7	51.8	79.4	30.2	39.5	28.8	61.5	13.7	35.2	55.0	41.8	48.0	31.8	23.0	57.0	57.9
	✗	4	49.3	56.3	93.1	36.2	51.7	34.9	72.6	18.2	37.7	70.8	42.8	50.4	33.6	24.7	62.7	-
	✗	8	50.0	58.0	99.0	40.8	55.2	39.4	73.4	23.9	40.0	75.0	**43.4**	51.2	33.6	25.8	63.9	-
	✗	16	50.8	59.4	102.2	44.5	58.5	43.0	72.7	27.6	41.5	77.2	42.4	51.3	33.5	27.6	64.5	-
	✗	32	51.0	60.4	106.3	47.2	57.4	44.0	72.8	29.4	40.7	77.3	41.2	OOC	32.6	28.4	63.5	-
Flamingo	✗	0	50.6	56.3	84.3	35.6	46.7	31.6	67.2	17.4	40.7	60.1	39.7	52.0	35.0	26.7	46.4	**60.8**
	✗	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	55.6	36.5	30.8	68.6	-
	✗	8	57.5	65.6	108.8	45.5	60.6	44.8	78.2	27.6	44.8	80.7	42.3	56.4	37.3	32.3	**70.0**	-
	✗	16	57.8	66.8	110.5	48.4	62.8	48.4	**78.9**	30.0	45.2	84.2	41.1	**56.8**	37.6	32.9	**70.0**	-
	✗	32	**57.8**	**67.6**	**113.8**	**52.3**	**65.1**	**49.8**	75.4	**31.0**	**45.3**	**86.8**	42.2	OOC	**37.9**	**33.5**	**70.0**	-
Pretrained FT SOTA	✔	(X)	54.4 [39] (10K)	80.2 [150] (444K)	143.3 [134] (500K)	47.9 [32] (27K)	76.3 [165] (500K)	57.2 [70] (20K)	67.4 [162] (30K)	46.8 [57] (130K)	35.4 [145] (6K)	138.7 [142] (10K)	36.7 [138] (46K)	75.2 [87] (123K)	54.7 [147] (20K)	25.2 [139] (38K)	75.4 [60] (9K)	-

Results: zero & few shot



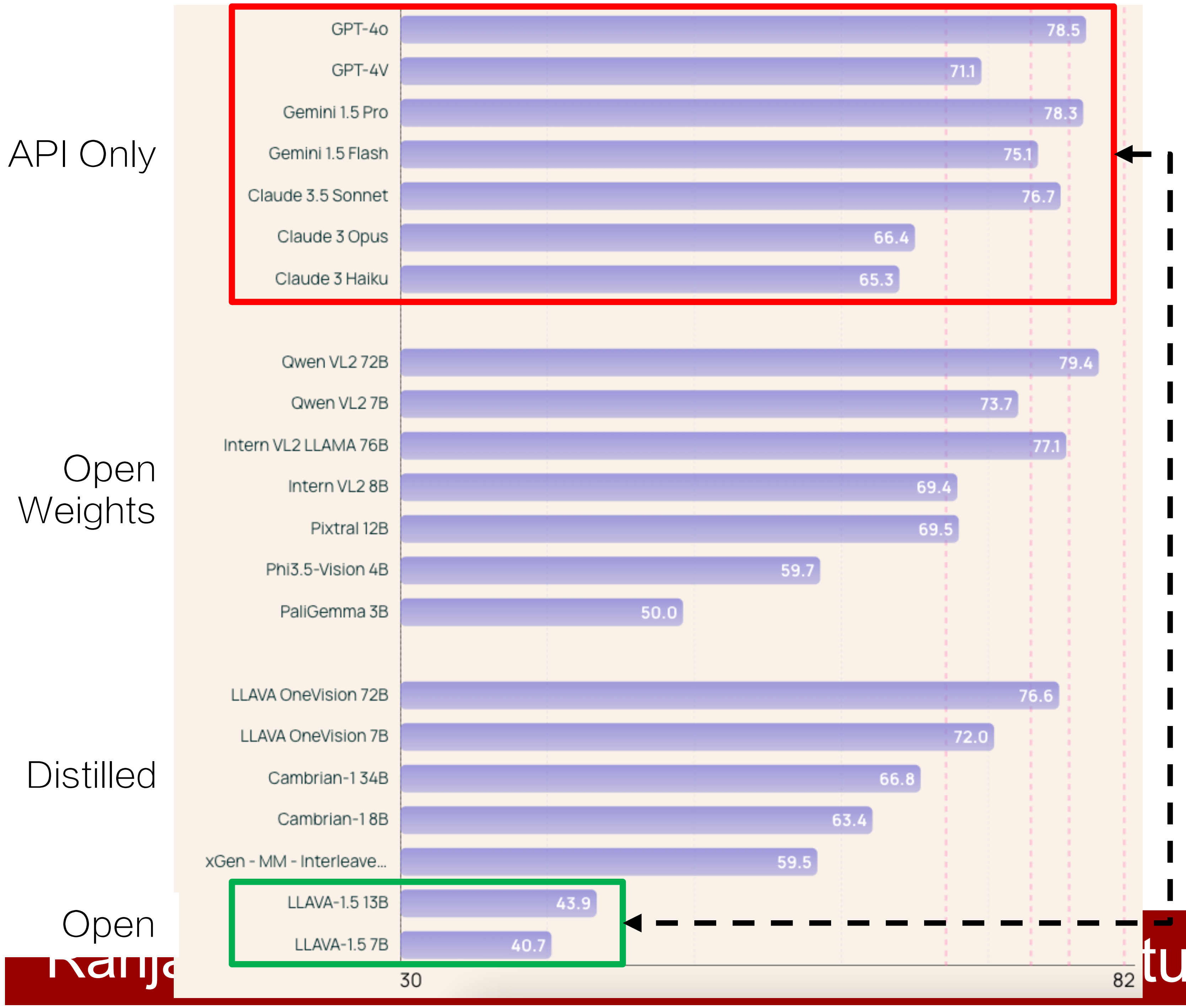
Method	FT	Shot	OKVQA	VQAv2	COCO	MSVDQA	VATEX	VizWiz	Flick30K	MSRVTTQA	iVQA	YouCook2	STAR	VisDial	TextVQA	NextQA	HatefulMemes	RareAct
Zero/Few shot SOTA	✗	(X)	[39] 43.3 (16)	[124] 38.2 (4)	[134] 32.2 (0)	[64] 35.2 (0)	-	-	-	[64] 19.2 (0)	[145] 12.2 (0)	-	[153] 39.4 (0)	[87] 11.6 (0)	-	-	[94] 66.1 (0)	[94] 40.7 (0)
Flamingo-3B	✗	0	41.2	49.2	73.0	27.5	40.1	28.9	60.6	11.0	32.7	55.8	39.6	46.1	30.1	21.3	53.7	58.4
	✗	4	43.3	53.2	85.0	33.0	50.0	34.0	72.0	14.9	35.7	64.6	41.3	47.3	32.7	22.4	53.6	-
	✗	8	44.6	55.4	90.6	37.0	54.5	38.4	71.7	19.6	36.8	68.0	40.6	47.6	32.4	23.9	54.7	-
	✗	16	45.6	56.7	95.4	40.2	57.1	43.3	73.4	23.4	37.4	73.2	40.1	47.5	31.8	25.2	55.3	-
	✗	32	45.9	57.1	99.0	42.6	59.2	45.5	71.2	25.6	37.7	76.7	41.6	OOC	30.6	26.1	56.3	-
Flamingo-9B	✗	0	44.7	51.8	79.4	30.2	39.5	28.8	61.5	13.7	35.2	55.0	41.8	48.0	31.8	23.0	57.0	57.9
	✗	4	49.3	56.3	93.1	36.2	51.7	34.9	72.6	18.2	37.7	70.8	42.8	50.4	33.6	24.7	62.7	-
	✗	8	50.0	58.0	99.0	40.8	55.2	39.4	73.4	23.9	40.0	75.0	**43.4**	51.2	33.6	25.8	63.9	-
	✗	16	50.8	59.4	102.2	44.5	58.5	43.0	72.7	27.6	41.5	77.2	42.4	51.3	33.5	27.6	64.5	-
	✗	32	51.0	60.4	106.3	47.2	57.4	44.0	72.8	29.4	40.7	77.3	41.2	OOC	32.6	28.4	63.5	-
Flamingo	✗	0	50.6	56.3	84.3	35.6	46.7	31.6	67.2	17.4	40.7	60.1	39.7	52.0	35.0	26.7	46.4	**60.8**
	✗	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	55.6	36.5	30.8	68.6	-
	✗	8	57.5	65.6	108.8	45.5	60.6	44.8	78.2	27.6	44.8	80.7	42.3	56.4	37.3	32.3	**70.0**	-
	✗	16	57.8	66.8	110.5	48.4	62.8	48.4	**78.9**	30.0	45.2	84.2	41.1	**56.8**	37.6	32.9	**70.0**	-
	✗	32	**57.8**	**67.6**	**113.8**	**52.3**	**65.1**	**49.8**	75.4	**31.0**	**45.3**	**86.8**	42.2	OOC	**37.9**	**33.5**	**70.0**	-
Pretrained FT SOTA	✔	(X)	54.4 [39] (10K)	80.2 [150] (444K)	143.3 [134] (500K)	47.9 [32] (27K)	76.3 [165] (500K)	57.2 [70] (20K)	67.4 [162] (30K)	46.8 [57] (130K)	35.4 [145] (6K)	138.7 [142] (10K)	36.7 [138] (46K)	75.2 [87] (123K)	54.7 [147] (20K)	25.2 [139] (38K)	75.4 [60] (9K)	-

Today, average performance across 11 visual understanding benchmarks



Rapid Decline in Brazilians' Assessment of Economy
Current economic situation in Brazil is ...
100%
Bad
Good
0
2010
2011
2012
2013
2014
2015
BILL WILLIAMS STUDIO, INC.

There are open-weight models
but they are all distilled from GPT



Rapid Decline in Brazilians' Assessment of Economy
Current economic situation in Brazil is ...
100%
Bad
Good
2010
2011
2012
2013
2014
2015
BILL WILLIAMS STUDIO, INC.

How do we close the gap without relying on proprietary models?

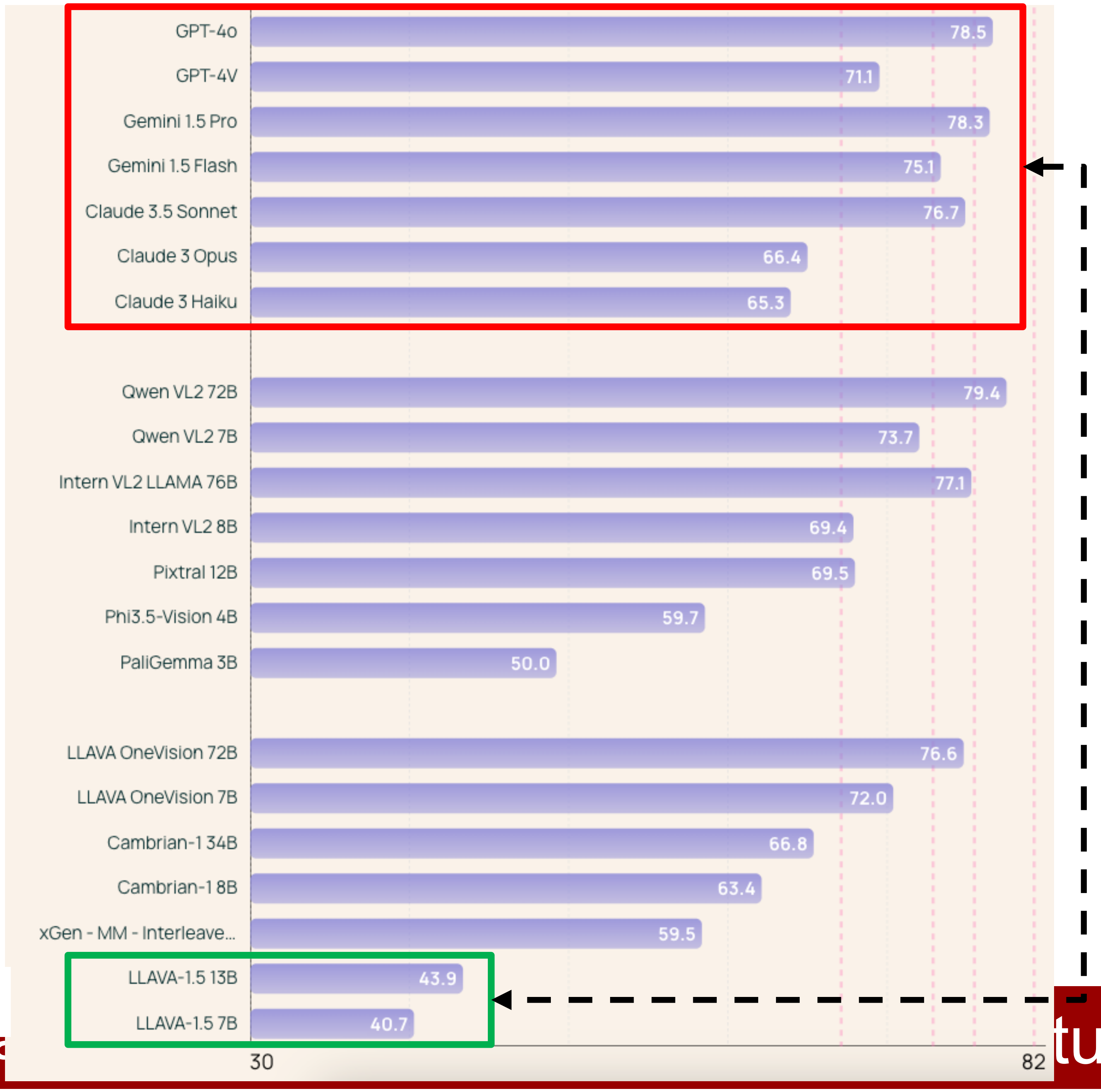
There are open-weight models
but they are all distilled from GPT

API Only

Open Weights

Distilled

Open



Rapid Decline in Brazilians' Assessment of Economy
Current economic situation in Brazil is ...
Bad
Good
2010
2011
2012
2013
2014
2015
BILL WILLIAMS STUDIO, INC.

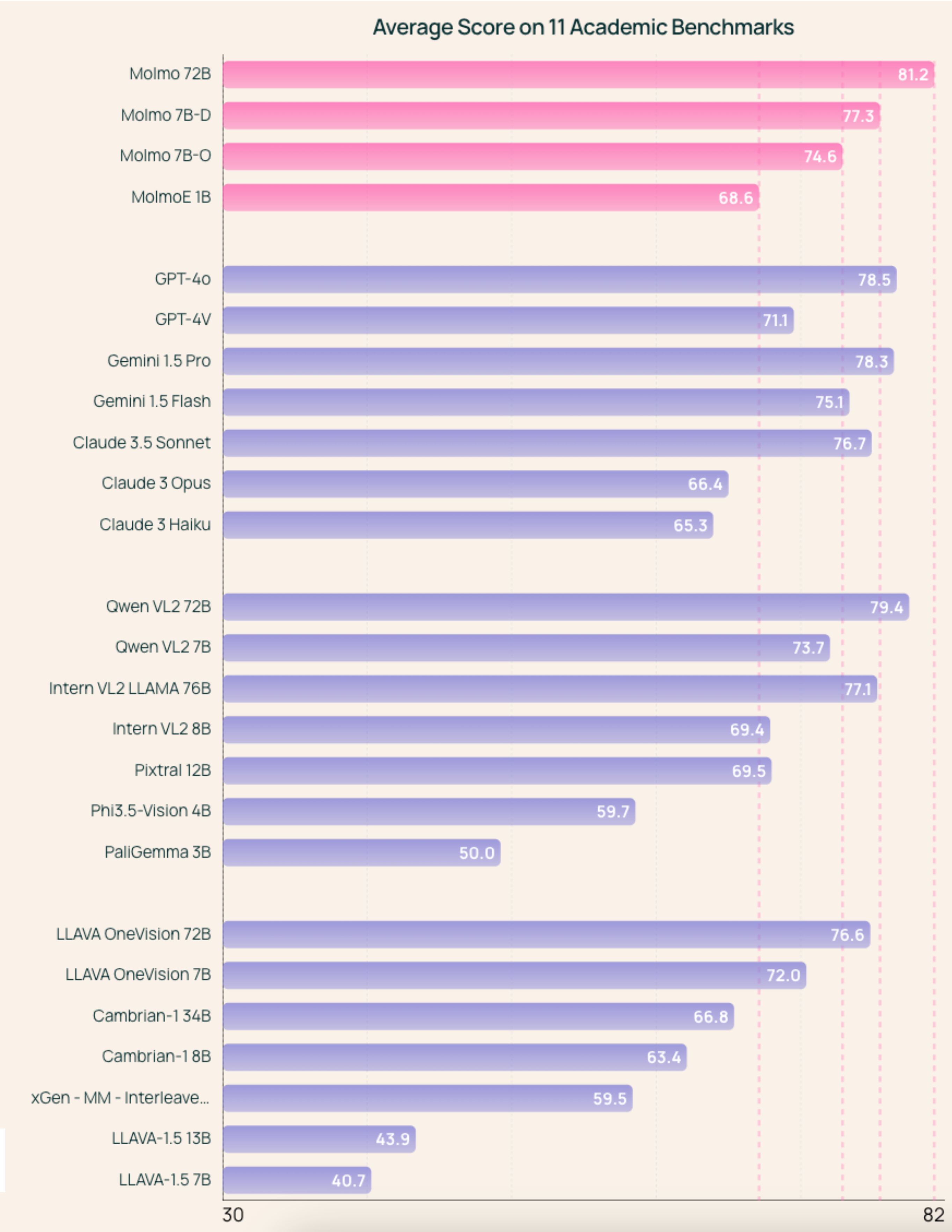
Open
Weights
Data
Code
Evals

API Only

Open
Weights

Distilled

Open



Completely
Open
Open Weights
Open Data
Open Code
Open Evals

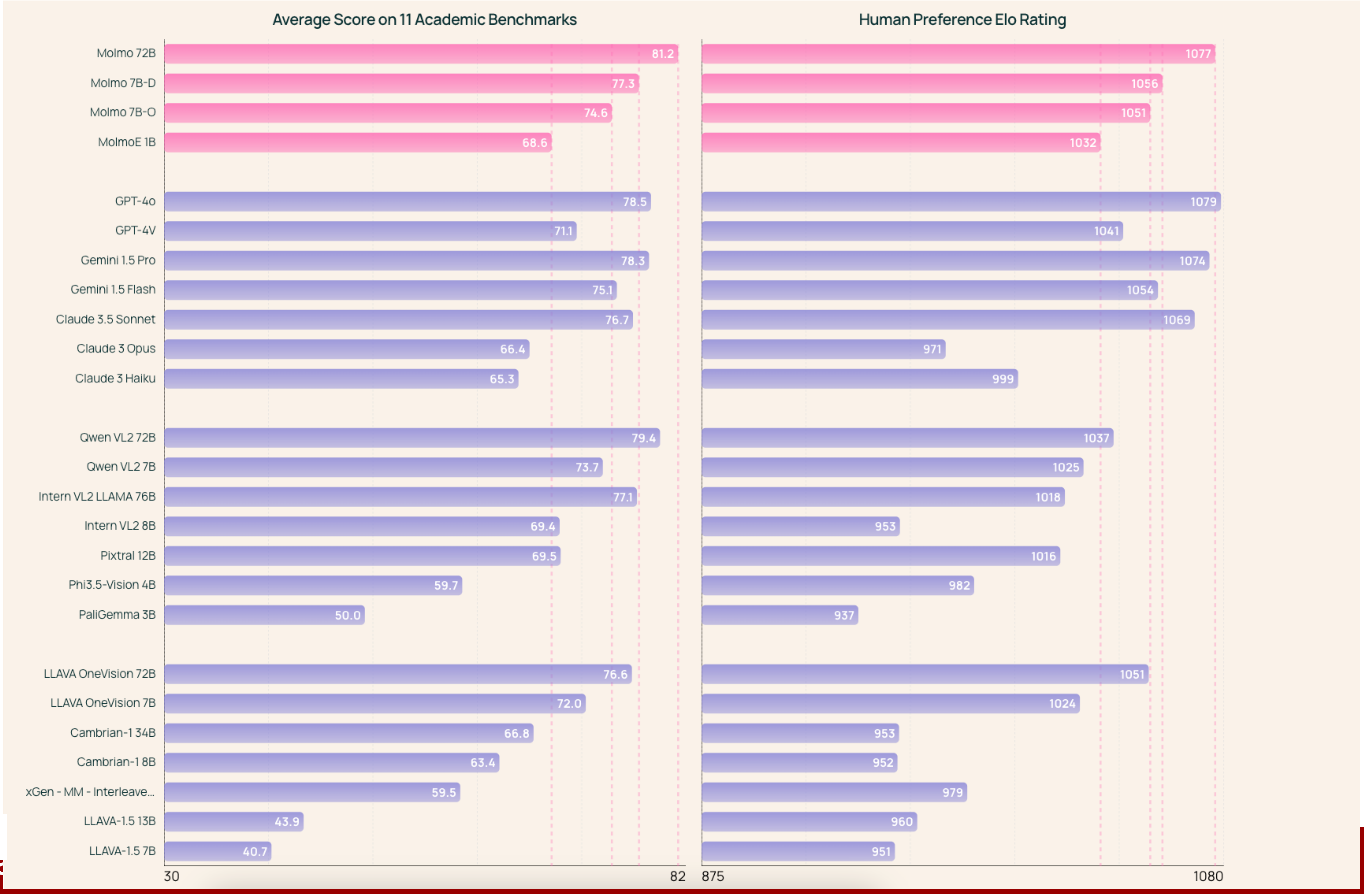
Open
Weights
Data
Code
Evals

API Only

Open
Weights

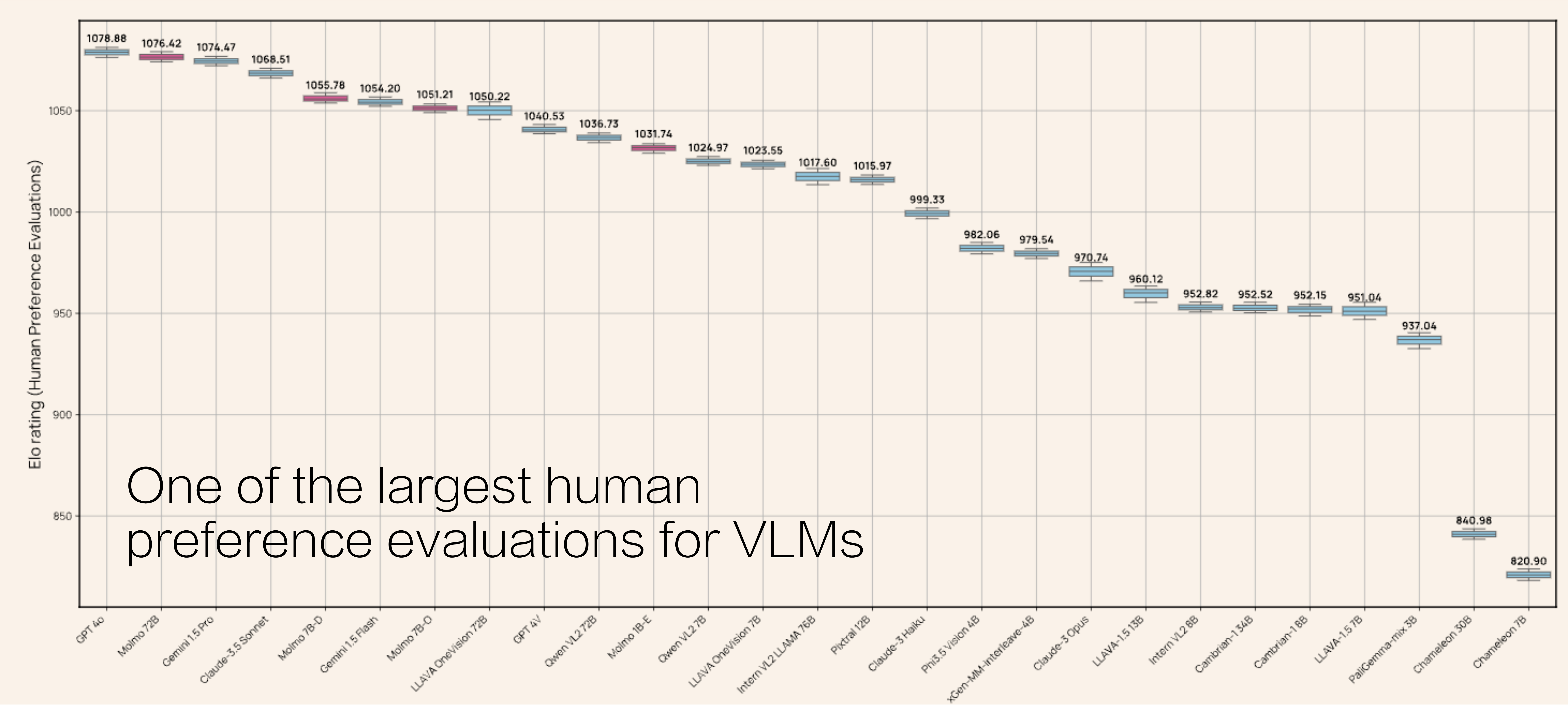
Distilled

Open



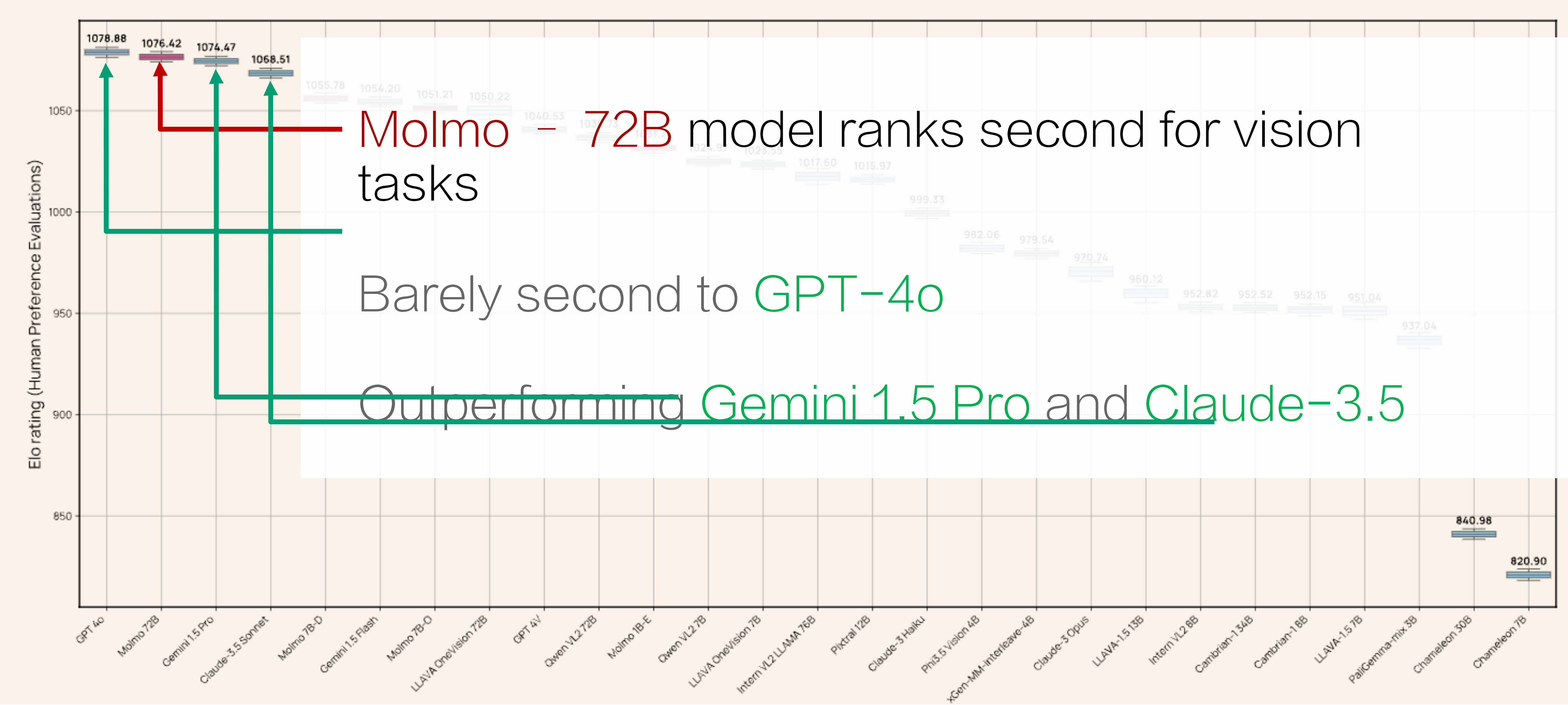
Ranja

2025



One of the largest human preference evaluations for VLMs

with 325k pairwise comparisons
and 870 human annotators

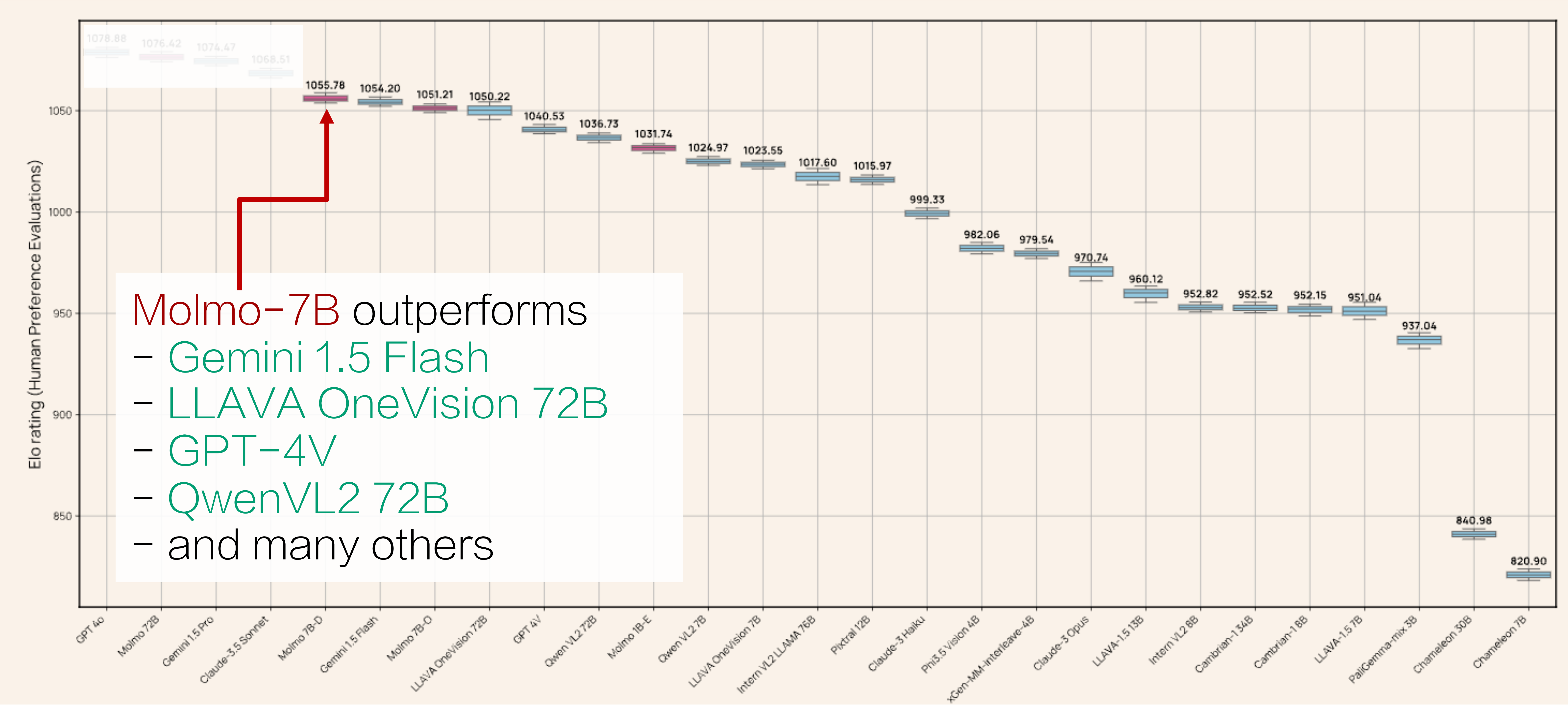


Molmo – 72B model ranks second for vision tasks

Barely second to GPT-4o

Outperforming Gemini 1.5 Pro and Claude-3.5

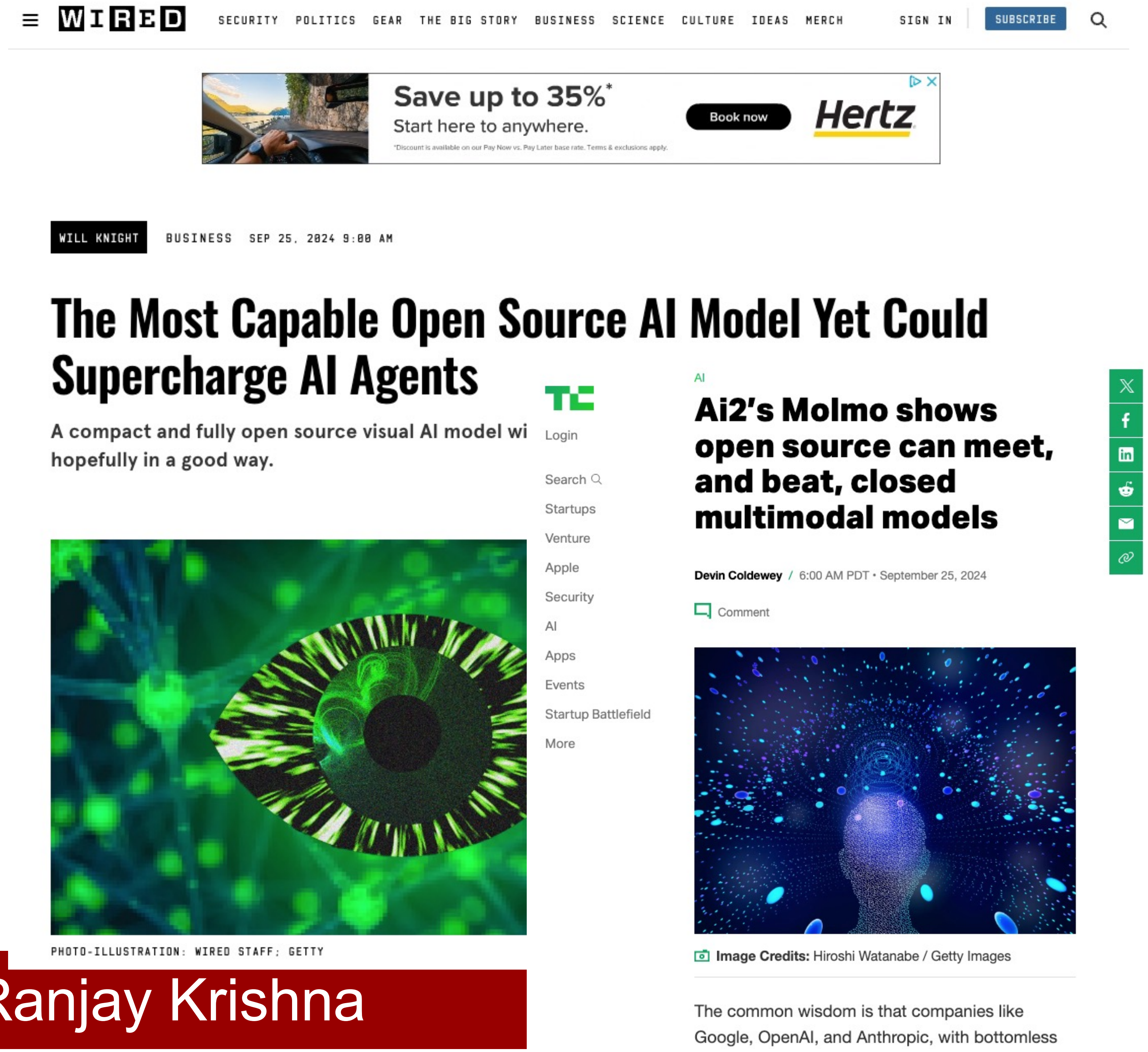
with 325k pairwise comparisons
and 870 human annotators



Molmo-7B outperforms
- Gemini 1.5 Flash
- LLAVA OneVision 72B
- GPT-4V
- QwenVL2 72B
- and many others

with 325k pairwise comparisons
and 870 human annotators

Reaction online – released Sep 25, 2024



Save up to 35%*
Start here to anywhere.
*Discount is available on our Pay Now vs. Pay Later base rate. Terms & exclusions apply.
Book now
Hertz

WILL KNIGHT BUSINESS SEP 25, 2024 9:00 AM

The Most Capable Open Source AI Model Yet Could Supercharge AI Agents

A compact and fully open source visual AI model wi
hopefully in a good way.

PHOTO-ILLUSTRATION: WIRED STAFF; GETTY

AI

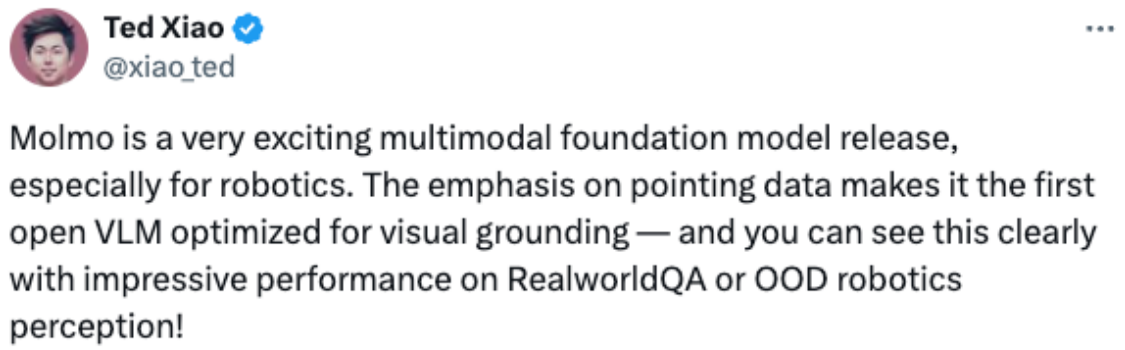
Ai2's Molmo shows open source can meet, and beat, closed multimodal models

Devin Coldewey / 6:00 AM PDT • September 25, 2024

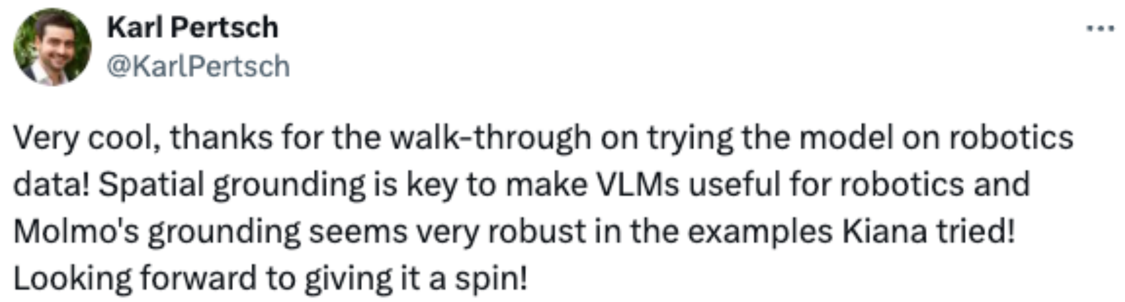
Comment

Image Credits: Hiroshi Watanabe / Getty Images

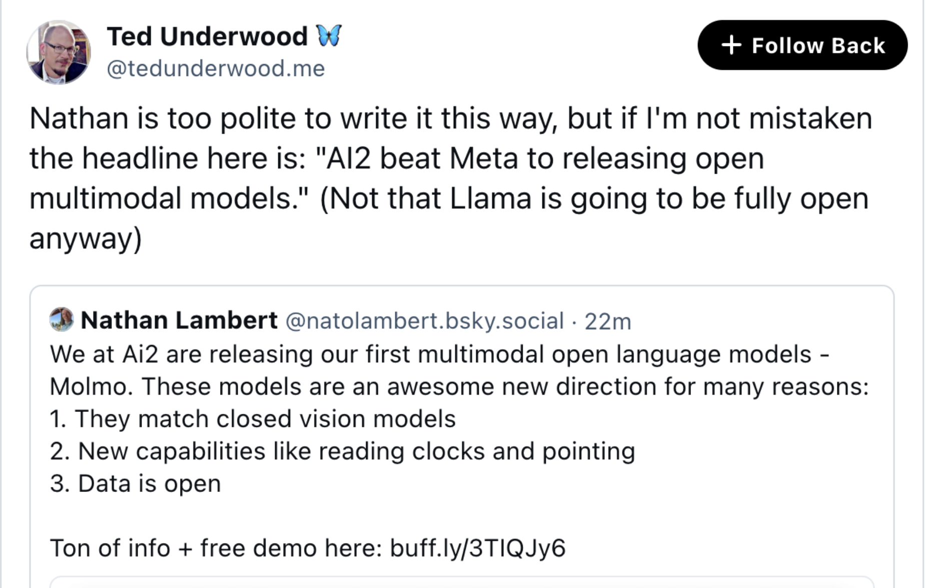
The common wisdom is that companies like Google, OpenAI, and Anthropic, with bottomless


Ted Xiao @xiao_ted

Molmo is a very exciting multimodal foundation model release, especially for robotics. The emphasis on pointing data makes it the first open VLM optimized for visual grounding — and you can see this clearly with impressive performance on RealworldQA or OOD robotics perception!


Karl Pertsch @KarlPertsch

Very cool, thanks for the walk-through on trying the model on robotics data! Spatial grounding is key to make VLMs useful for robotics and Molmo's grounding seems very robust in the examples Kiana tried! Looking forward to giving it a spin!

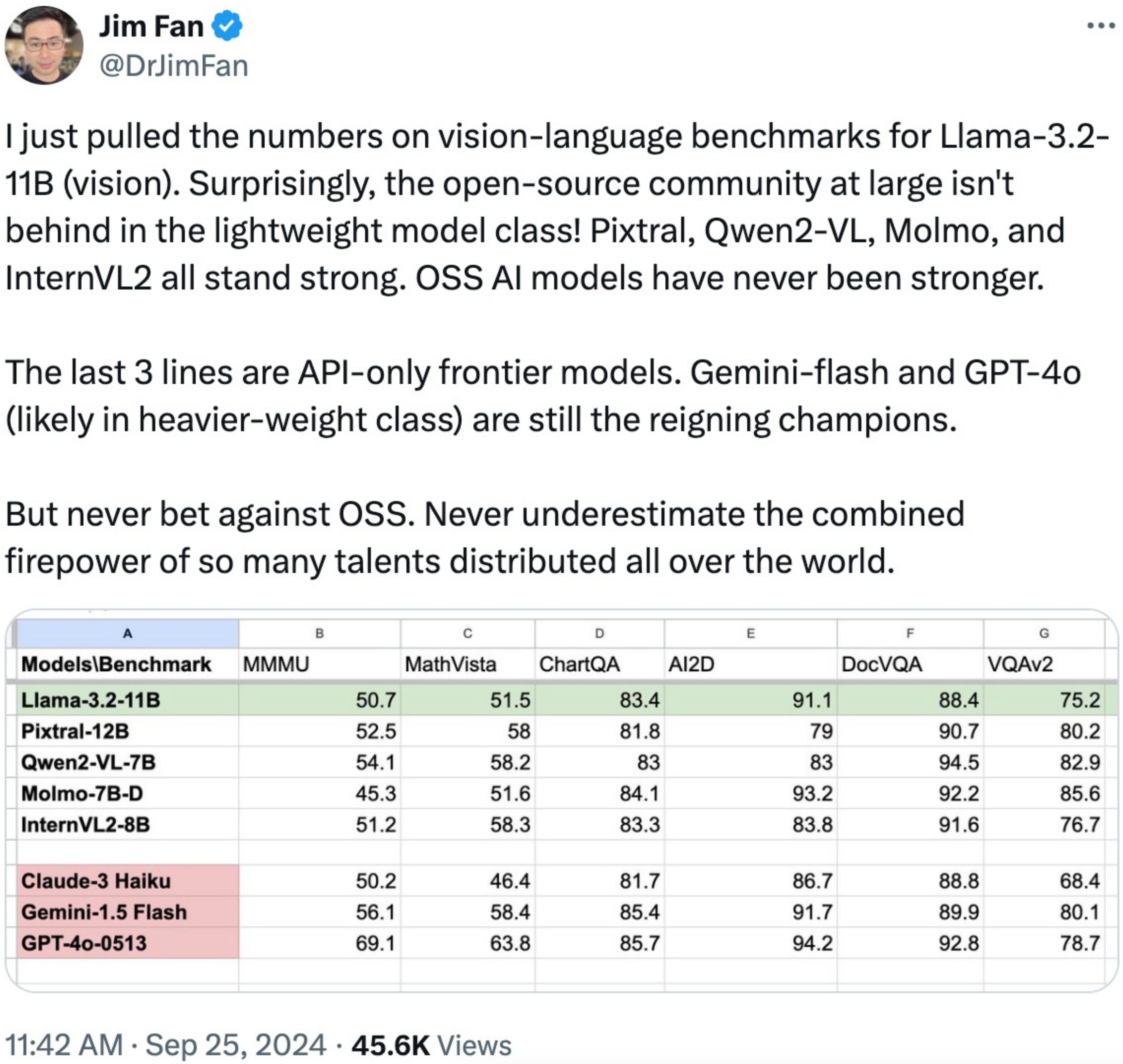

Ted Underwood @tedunderwood.me

Nathan is too polite to write it this way, but if I'm not mistaken the headline here is: "AI2 beat Meta to releasing open multimodal models." (Not that Llama is going to be fully open anyway)

Nathan Lambert @natolambert.bsky.social · 22m
We at Ai2 are releasing our first multimodal open language models - Molmo. These models are an awesome new direction for many reasons:
1. They match closed vision models
2. New capabilities like reading clocks and pointing
3. Data is open

Ton of info + free demo here: buff.ly/3TIQJy6

Never bet against open-source software!



Jim Fan
@DrJimFan

I just pulled the numbers on vision-language benchmarks for Llama-3.2-11B (vision). Surprisingly, the open-source community at large isn't behind in the lightweight model class! Pixtral, Qwen2-VL, Molmo, and InternVL2 all stand strong. OSS AI models have never been stronger.

The last 3 lines are API-only frontier models. Gemini-flash and GPT-4o (likely in heavier-weight class) are still the reigning champions.

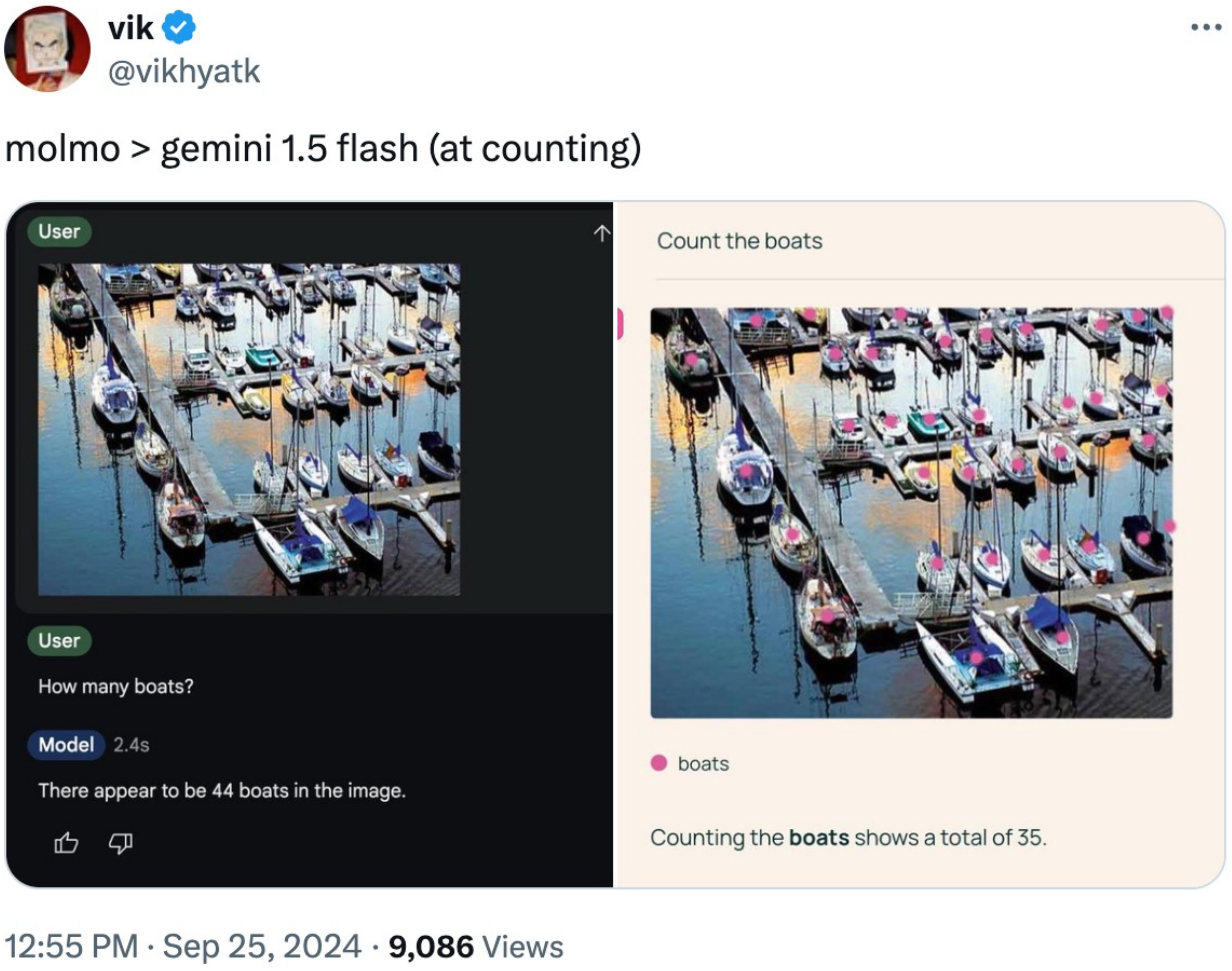
But never bet against OSS. Never underestimate the combined firepower of so many talents distributed all over the world.

Models\Benchmark	MMMU	MathVista	ChartQA	AI2D	DocVQA	VQAv2
Llama-3.2-11B	50.7	51.5	83.4	91.1	88.4	75.2
Pixtral-12B	52.5	58	81.8	79	90.7	80.2
Qwen2-VL-7B	54.1	58.2	83	83	94.5	82.9
Molmo-7B-D	45.3	51.6	84.1	93.2	92.2	85.6
InternVL2-8B	51.2	58.3	83.3	83.8	91.6	76.7
Claude-3 Haiku	50.2	46.4	81.7	86.7	88.8	68.4
Gemini-1.5 Flash	56.1	58.4	85.4	91.7	89.9	80.1
GPT-4o-0513	69.1	63.8	85.7	94.2	92.8	78.7

11:42 AM · Sep 25, 2024 · **45.6K** Views

Molmo grounds reasoning directly in the pixels

Example, it points when it counts



vik @vikhyatk

molmo > gemini 1.5 flash (at counting)

User

User

How many boats?

Model 2.4s

There appear to be 44 boats in the image.

Count the boats

● boats

Counting the **boats** shows a total of 35.

12:55 PM · Sep 25, 2024 · **9,086** Views

Data matters! Quality over quantity even for pretraining

LLAMA 3.1V

6 Billion Image-Text pairs

Molmo is trained with

PixMo

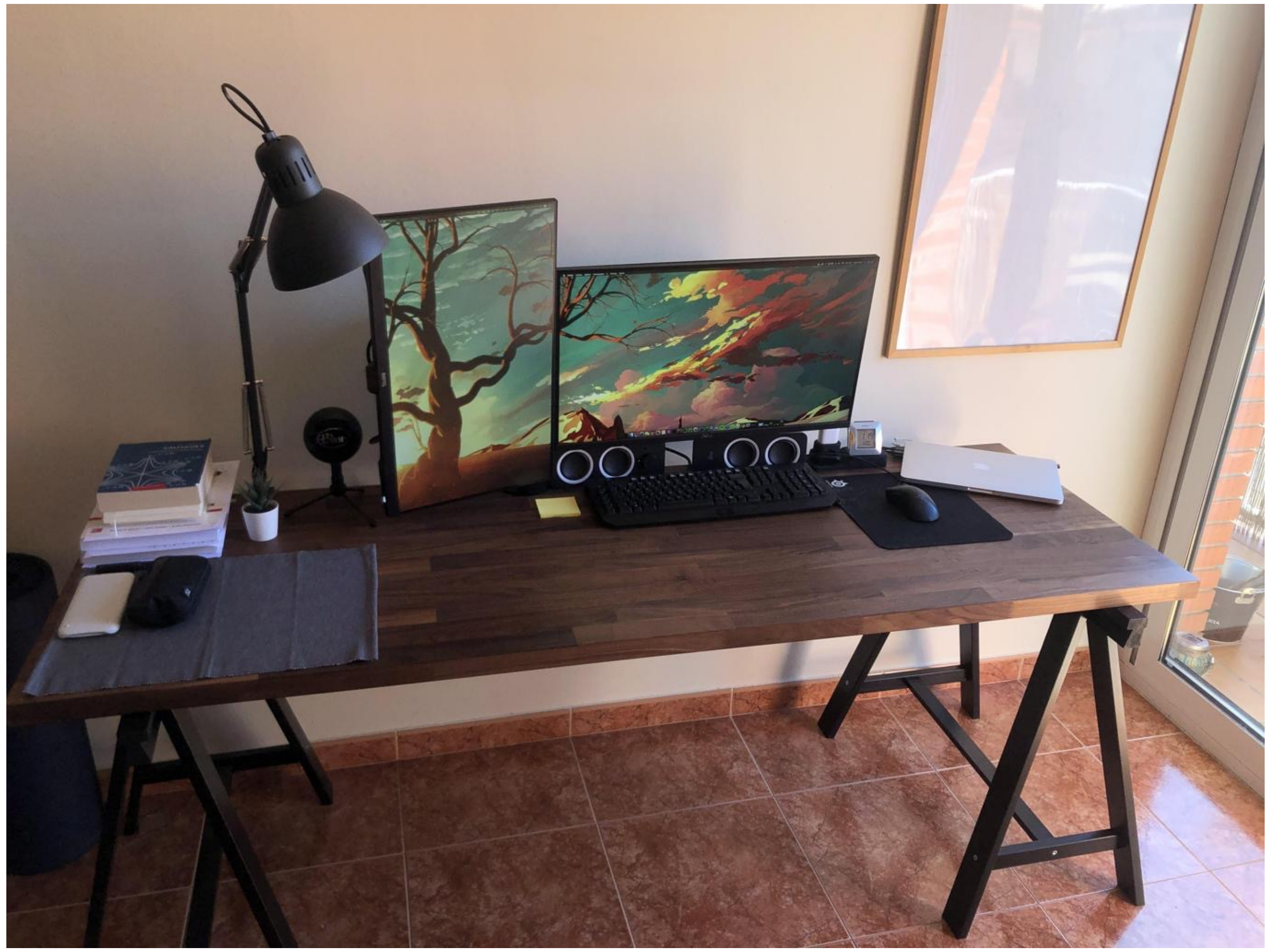
700k Image−Text pairs

Internet data is incidental
Human annotated data is intentional

pink, japan,
aesthetic image

love this winter picture by
person

PixMo data is intentional:

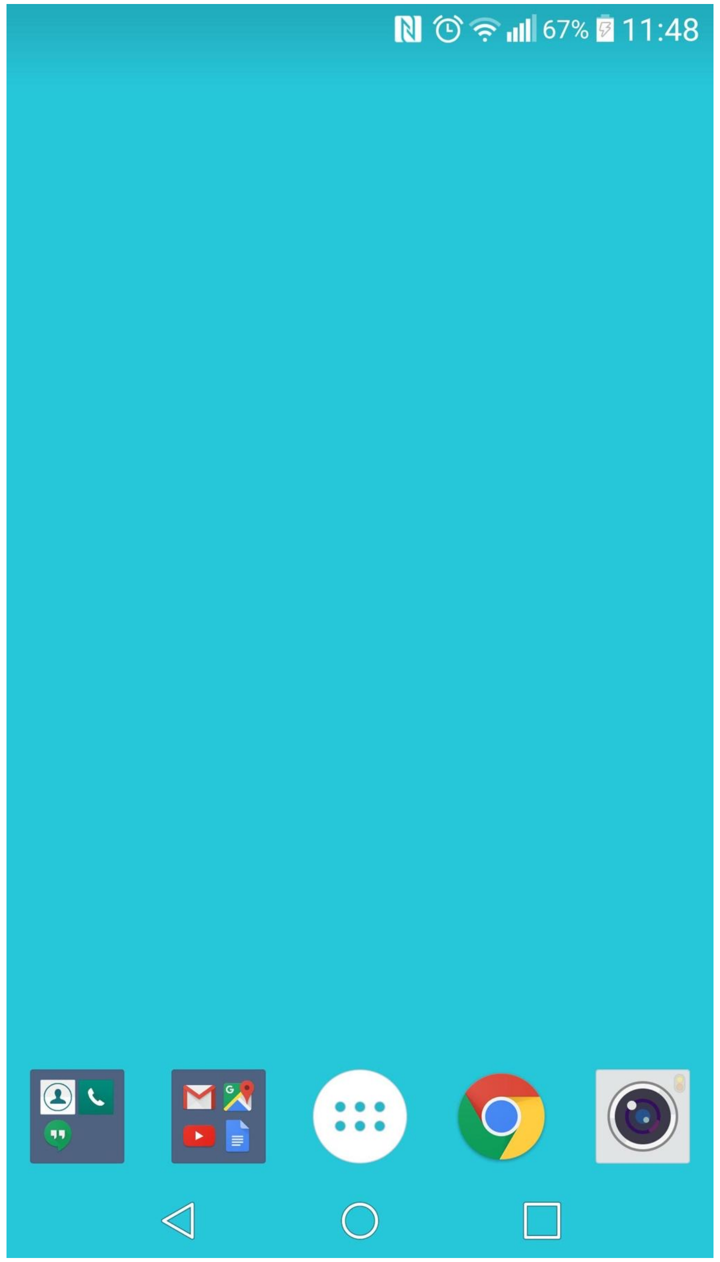


This photograph captures a well-organized work desk set prominently in the middle of the frame. The desk is large and rectangular, made from a polished, rich wood that spans horizontally across the image. Its structure is supported by four distinctive A-shaped legs, adding an elegant touch. On the desk, a striking dual-monitor setup is noticeable: a tall, vertical screen placed behind and to the right of a wider, horizontal computer monitor.

To the right of these monitors, a black mouse rests on a mouse pad. Scattered around the mouse pad, some white papers or letters are strewn across the far right side. On the left side of the desk, a black desk lamp with an extended arm hangs down, illuminating the workspace. Nearby, a stack of books is neatly placed in the upper left corner of the table.

The background wall is painted a subtle beige-white, complementing the refined ambiance of the space. The floor below the desk features elegant pinkish marble tiles, enhancing the room’s sophisticated look. To the far right of the image, a large window or patio door allows natural light to pour in, with clear glass that offers a glimpse into the outside area. This exterior view includes part of a rustic brick wall and a metal pail, hinting at an adjacent patio.

Collecting dense captions is hard!!!



This image features a screenshot taken from a tablet device. At the top-right corner, the time is displayed as 11:48, alongside a battery icon indicating 67% power remaining. The device also shows a telephony signal strength of four out of five bars and roughly three-quarters Wi-Fi connectivity. Additionally, a clock alarm icon is present, as well as an icon resembling a white rectangle with a blue letter "N" in it, whose specific function is unclear.

The main portion of the screen, approximately 80% of it, is a solid medium blue color devoid of any content. At the bottom of the screen, there are several folders and icons representing various apps and functionalities:

1. The first folder contains three icons:
- An icon likely for contacts.
- An icon probably for telephone.
- An icon that seems to represent a text messaging app.

2. The second folder houses four apps:
- A Gmail app icon indicated by a red "M" on a white background.
- A YouTube app icon characterized by a red play button.
- A map app icon depicting a map.
- An unidentified app icon represented by a blue folder with its top corner bent down on the right.

3. The third icon is a white oval with six dots, arranged in two rows of three, likely representing an app drawer or menu.

4. The fourth icon resembles a red, yellow, and green shutter with a blue dot in the middle, suggesting it might be for a camera or photo viewing app.

5. Next to it, there is a silver colored camera icon with a black lens and a blue spot in the center, hinting at a camera application.

At the very bottom of the screen, there are three navigational icons:
- A left arrow triangle in white at the bottom-left corner.
- A white oval in the center, indicative of a home button.
- A white rectangle on the bottom-right corner, likely for accessing recent apps or multitasking.

Overall, this image captures the home screen of a tablet, providing an overview of available functionalities and connectivity status.

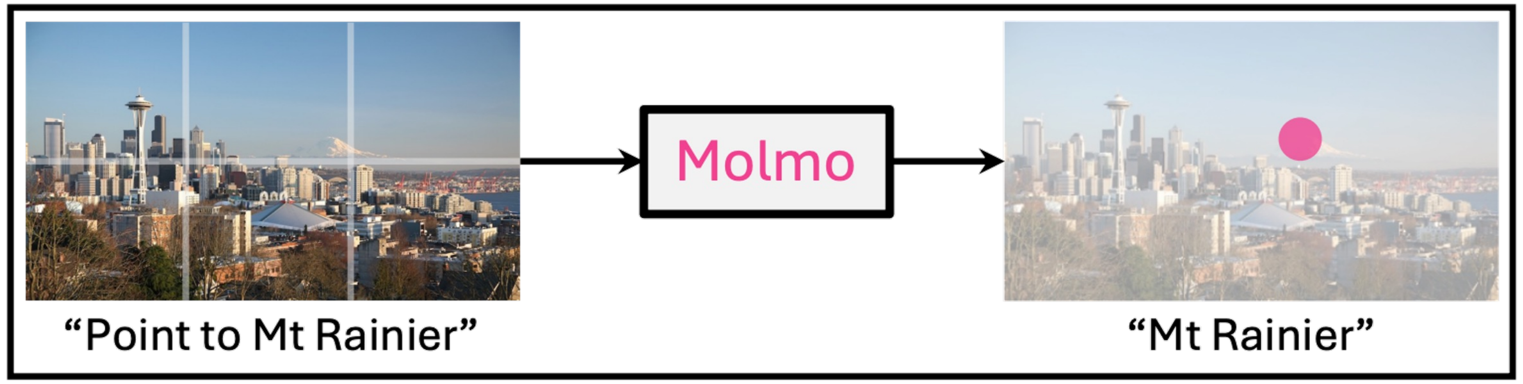
Questions designed to extract meaningful visual information from annotators

- What is the image at first glance?
- What are the objects and their counts?
- What does the text say?
- What are the positions of the objects?
- What subtle details are noticeable?
- What is in the background?
- What is the style and color?

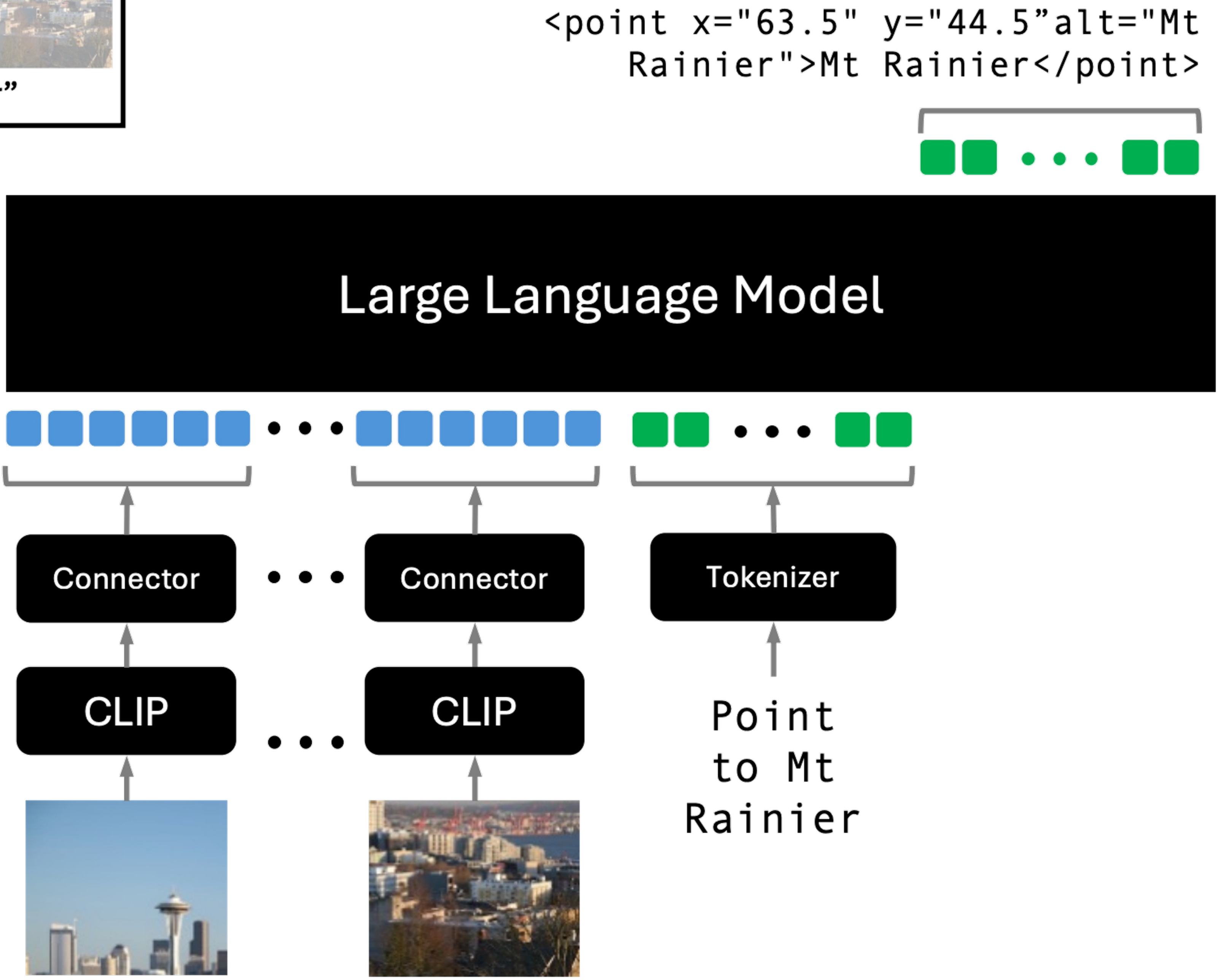
People don’t like to type
… but they love to talk

We ask annotators to speak for 60 to 90 seconds about an image

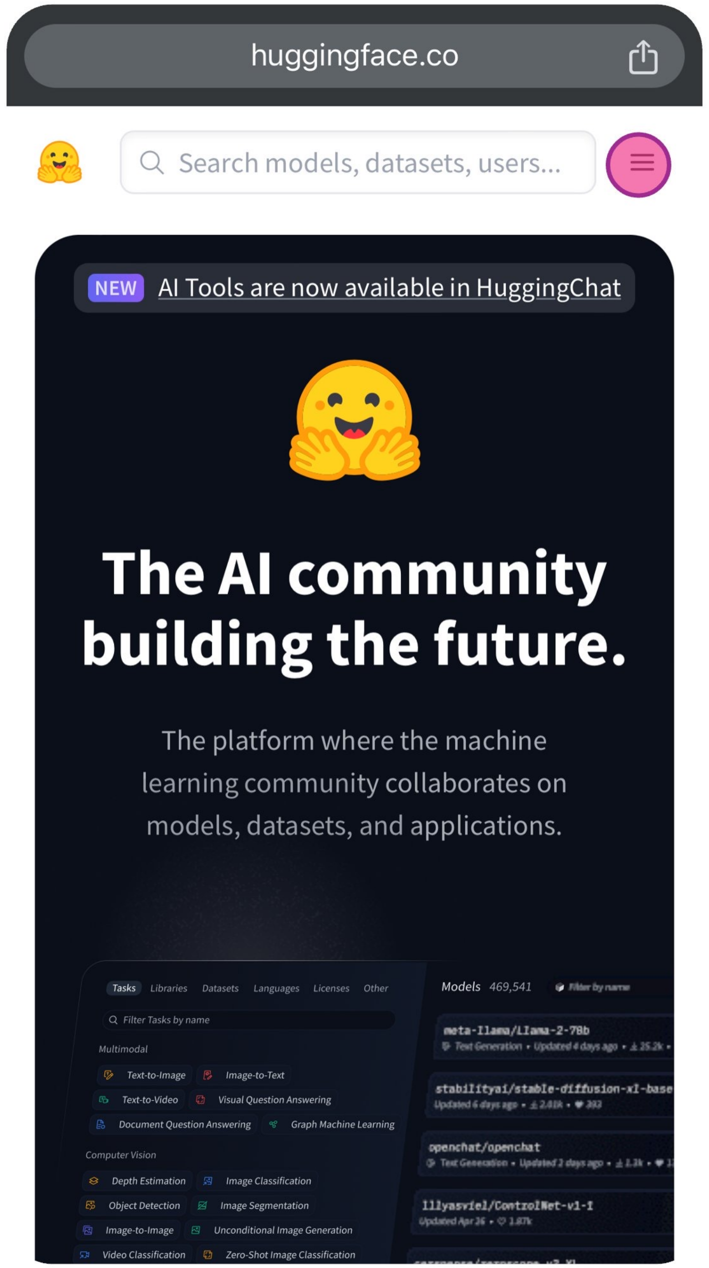
We automatically convert speech into text for pretraining



Start with off-the-self
Large Language Model
& Visual Encoder


```
<point x="63.5" y="44.5”alt="Mt
Rainier">Mt Rainier</point>
```

From perception To action

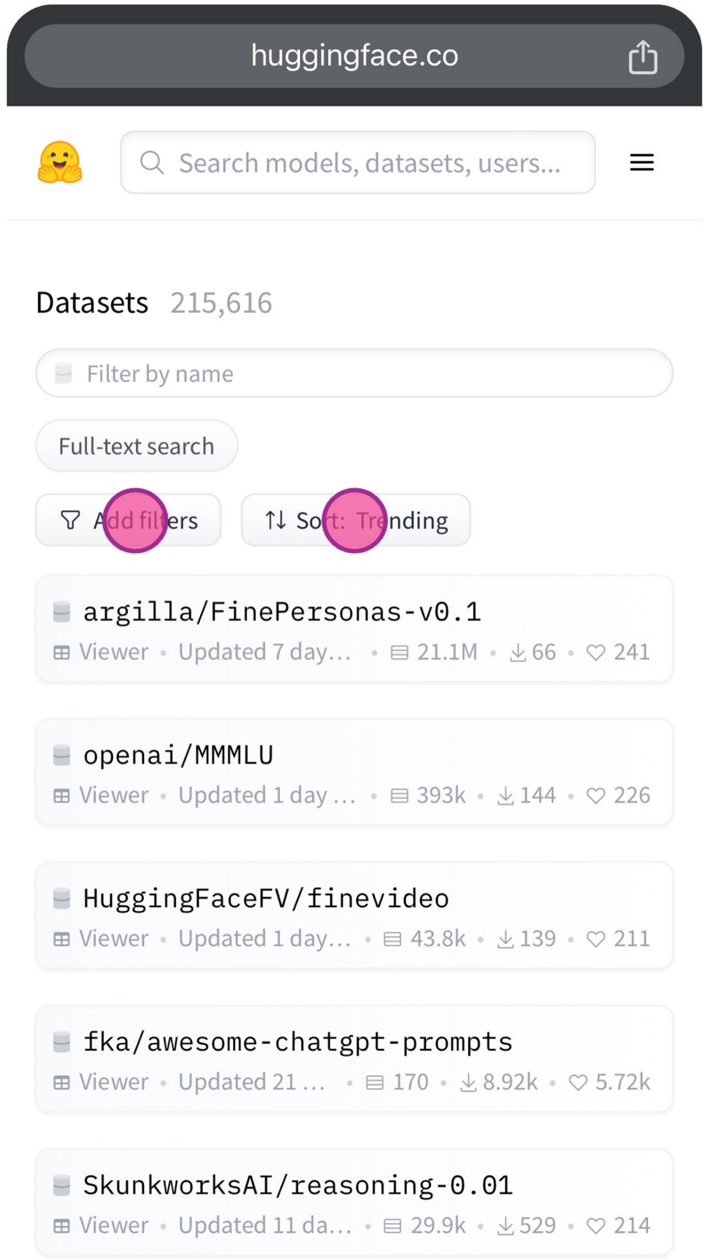


NEW AI Tools are now available in HuggingChat

The AI community building the future.

The platform where the machine learning community collaborates on models, datasets, and applications.

“Point to the menu”



Datasets 215,616

argilla/FinePersonas-v0.1
Viewer • Updated 7 day… • 21.1M • 66 • 241

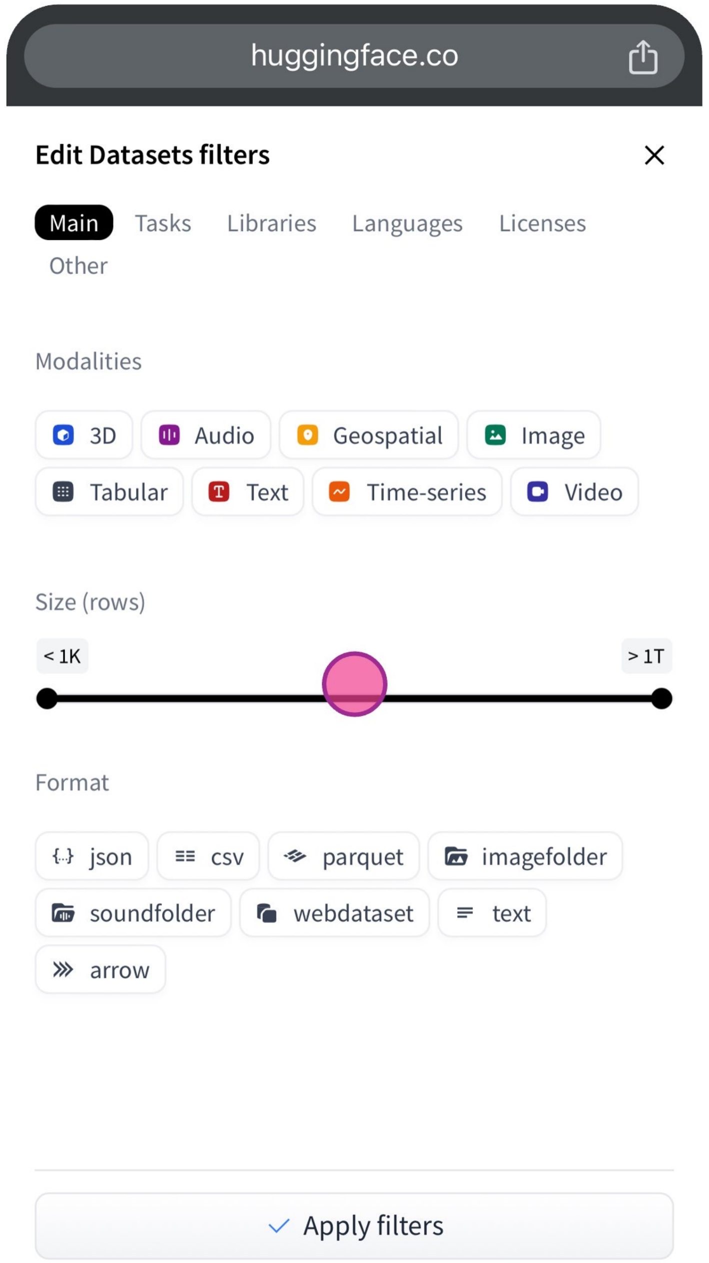
openai/MMMLU
Viewer • Updated 1 day … • 393k • 144 • 226

HuggingFaceFV/finevideo
Viewer • Updated 1 day… • 43.8k • 139 • 211

fka/awesome-chatgpt-prompts
Viewer • Updated 21 … • 170 • 8.92k • 5.72k

SkunkworksAI/reasoning-0.01
Viewer • Updated 11 da… • 29.9k • 529 • 214

“Point to where I can set search options”



Edit Datasets filters

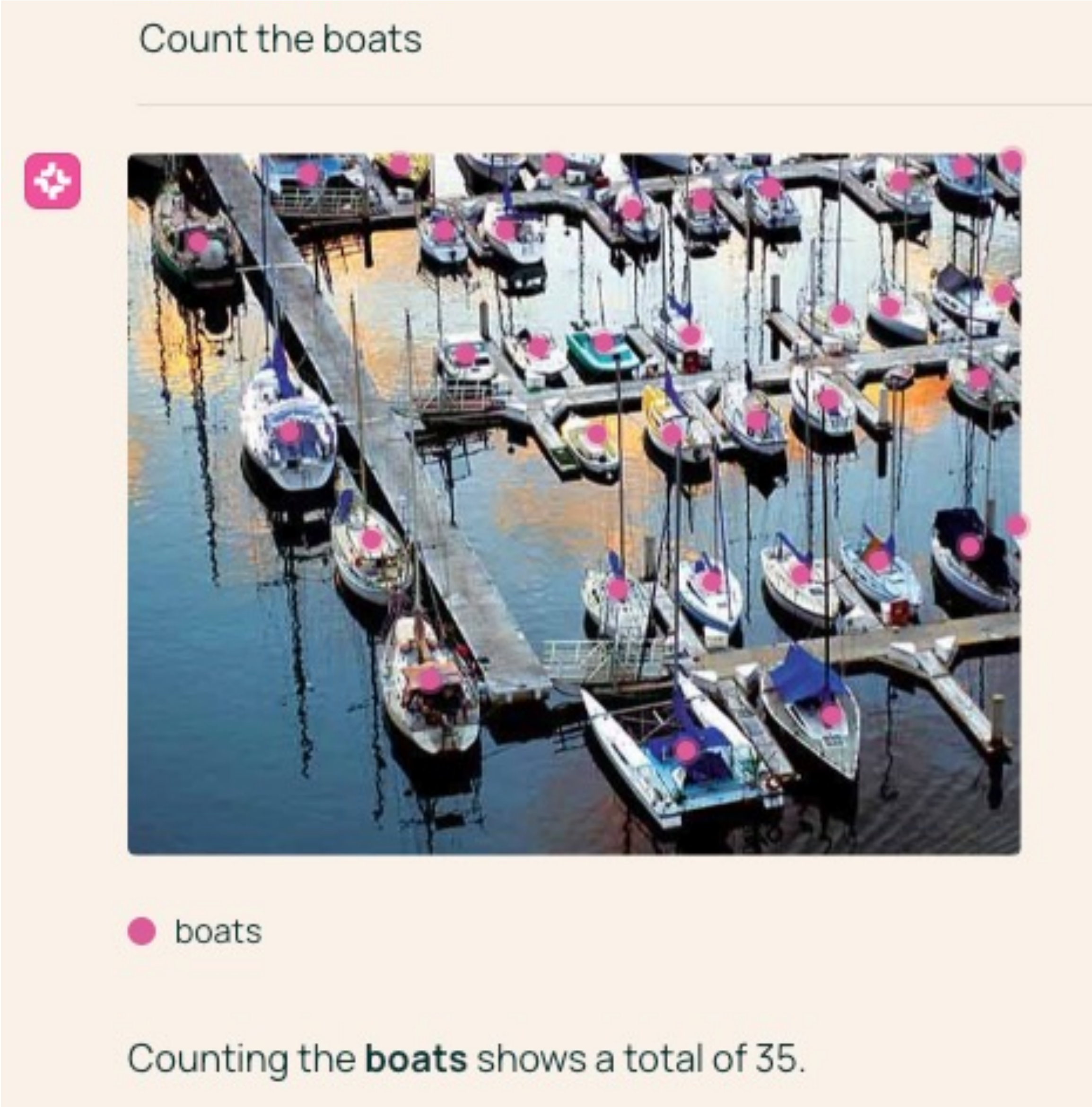
Modalities

Size (rows)

Format

“Point to where I can find mid size datasets”

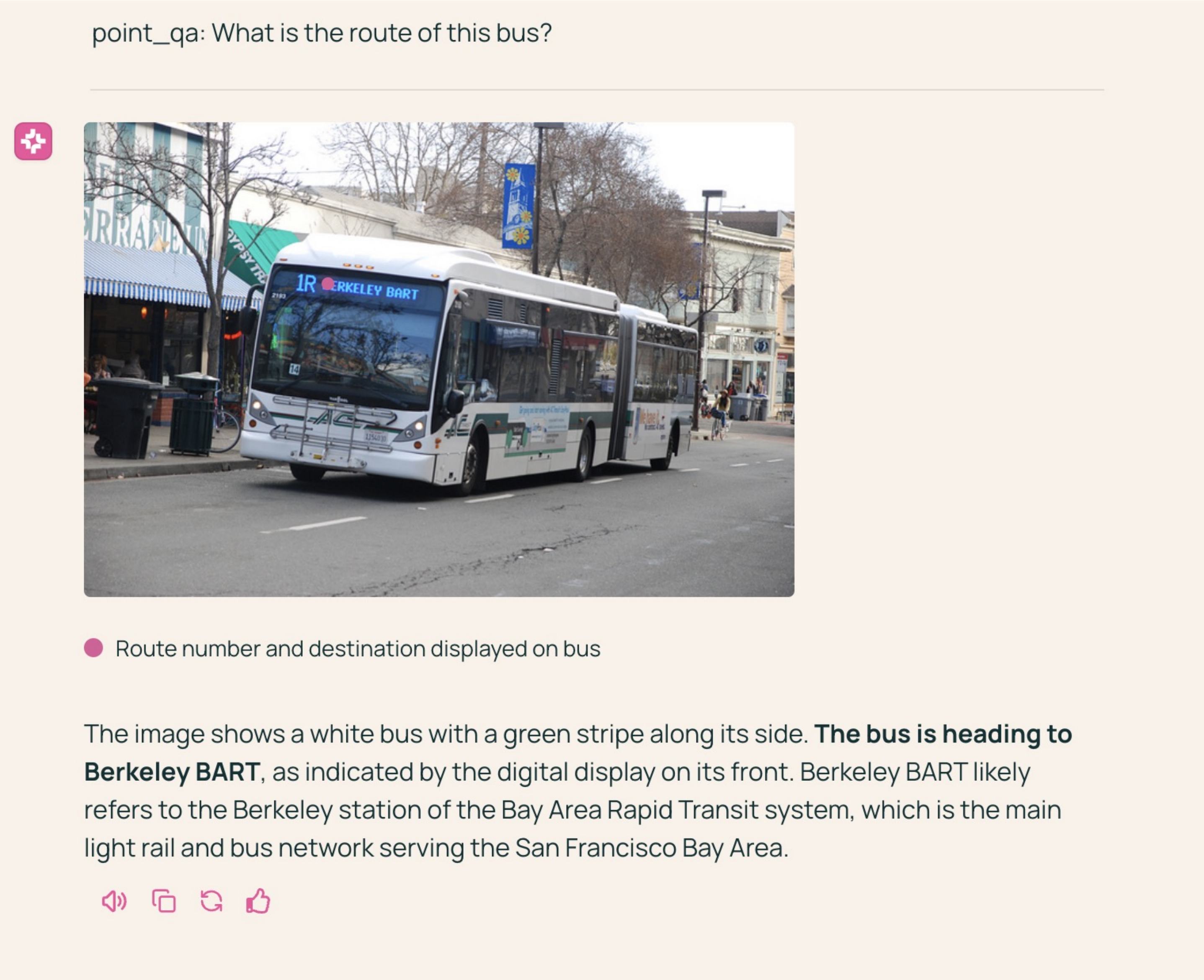
Pointing to count, pointing to ground



Count the boats

● boats

Counting the **boats** shows a total of 35.

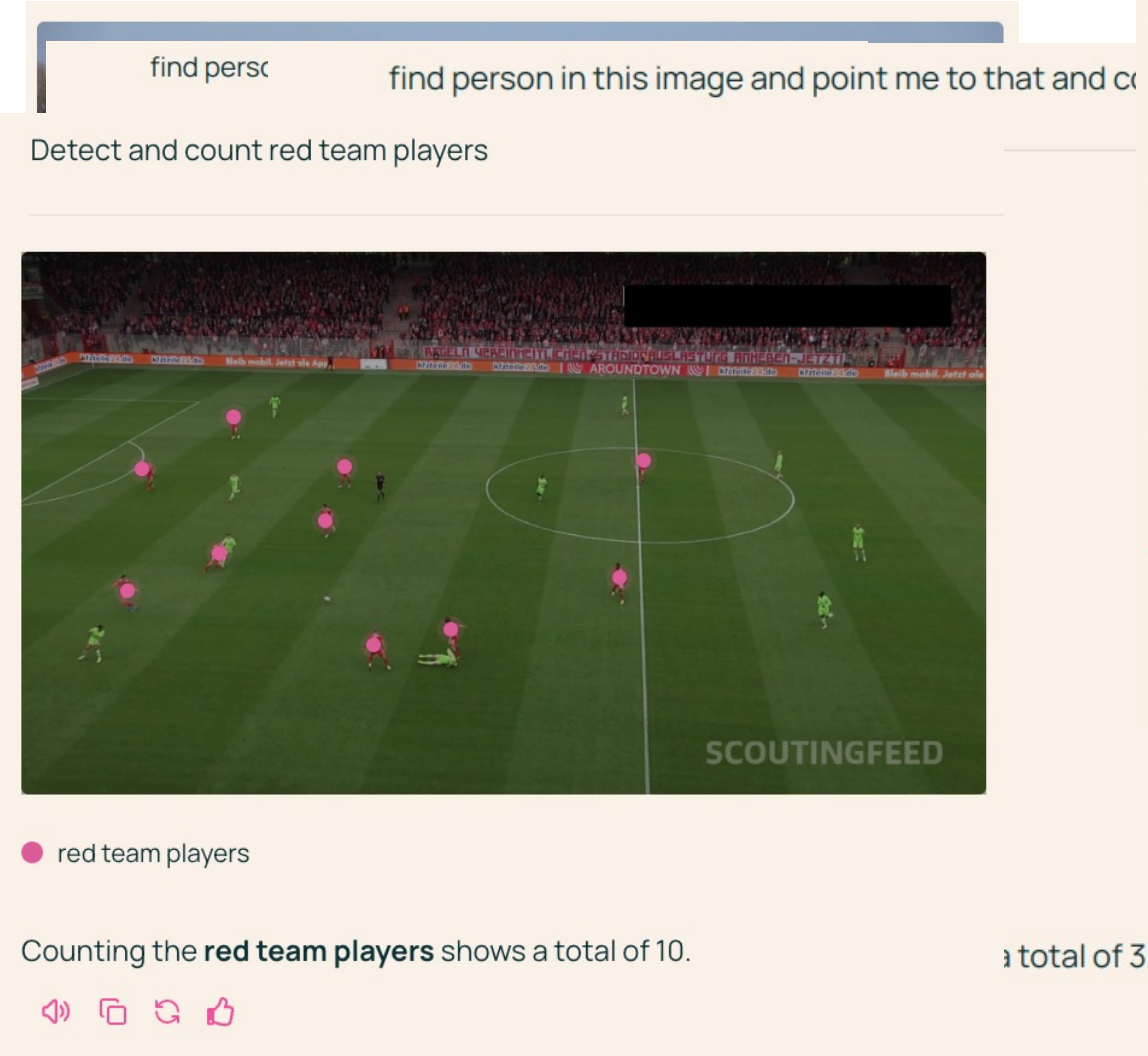


point_qa: What is the route of this bus?

● Route number and destination displayed on bus

The image shows a white bus with a green stripe along its side. **The bus is heading to Berkeley BART**, as indicated by the digital display on its front. Berkeley BART likely refers to the Berkeley station of the Bay Area Rapid Transit system, which is the main light rail and bus network serving the San Francisco Bay Area.

Pointing examples



find persc

find person in this image and point me to that and c

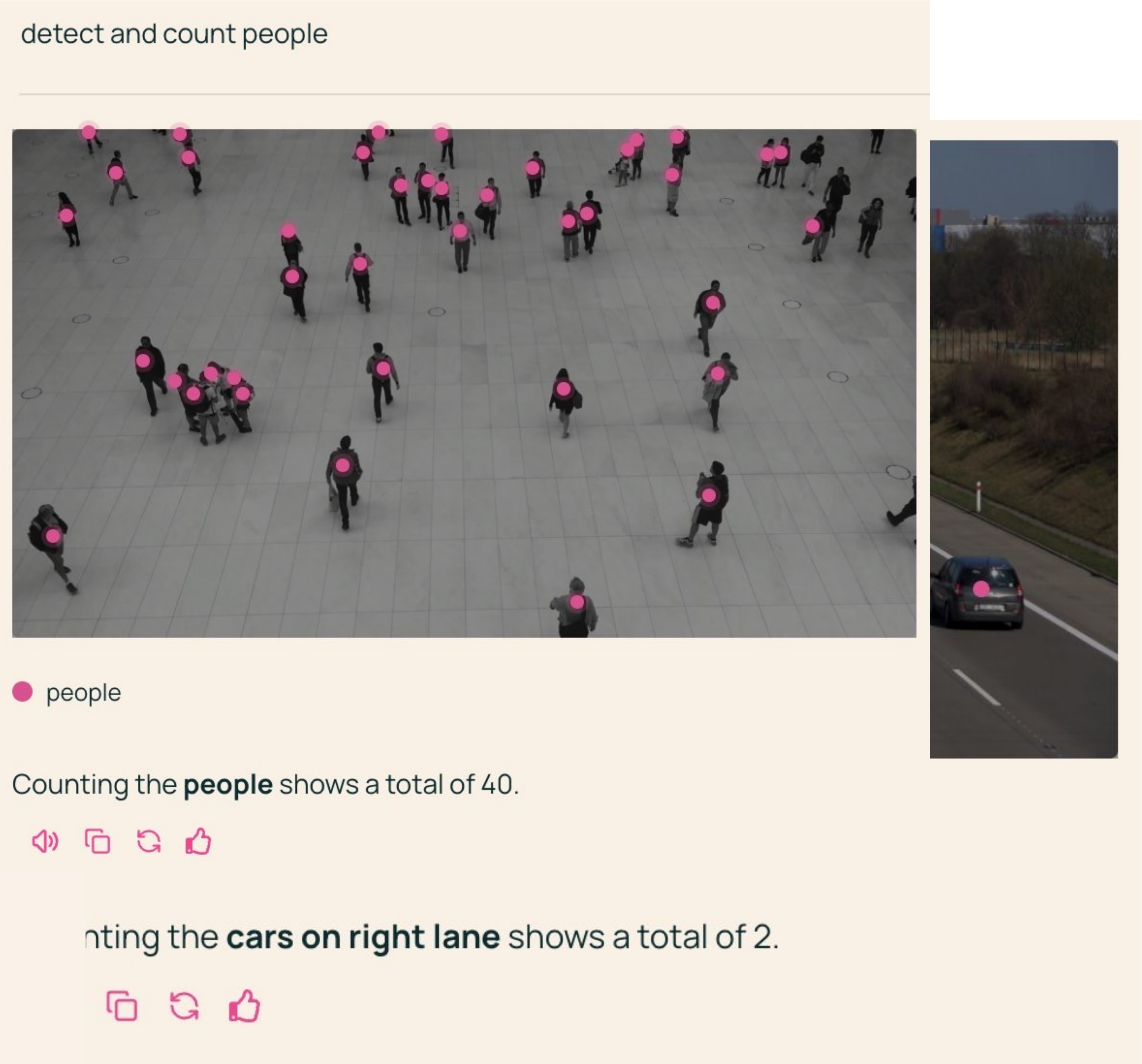
Detect and count red team players

[redacted]

● red team players

Counting the **red team players** shows a total of 10.

total of 3.



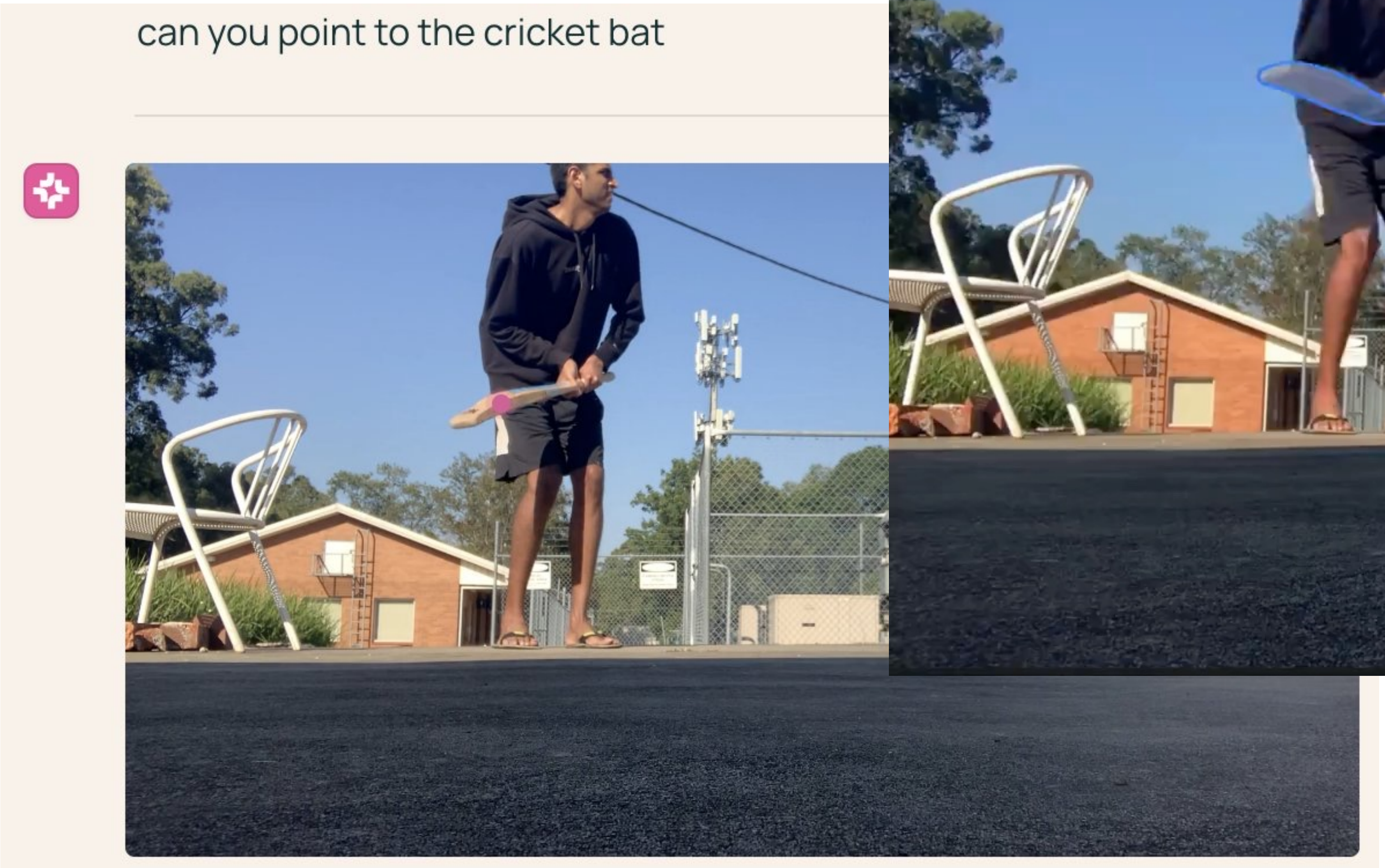
detect and count people

● people

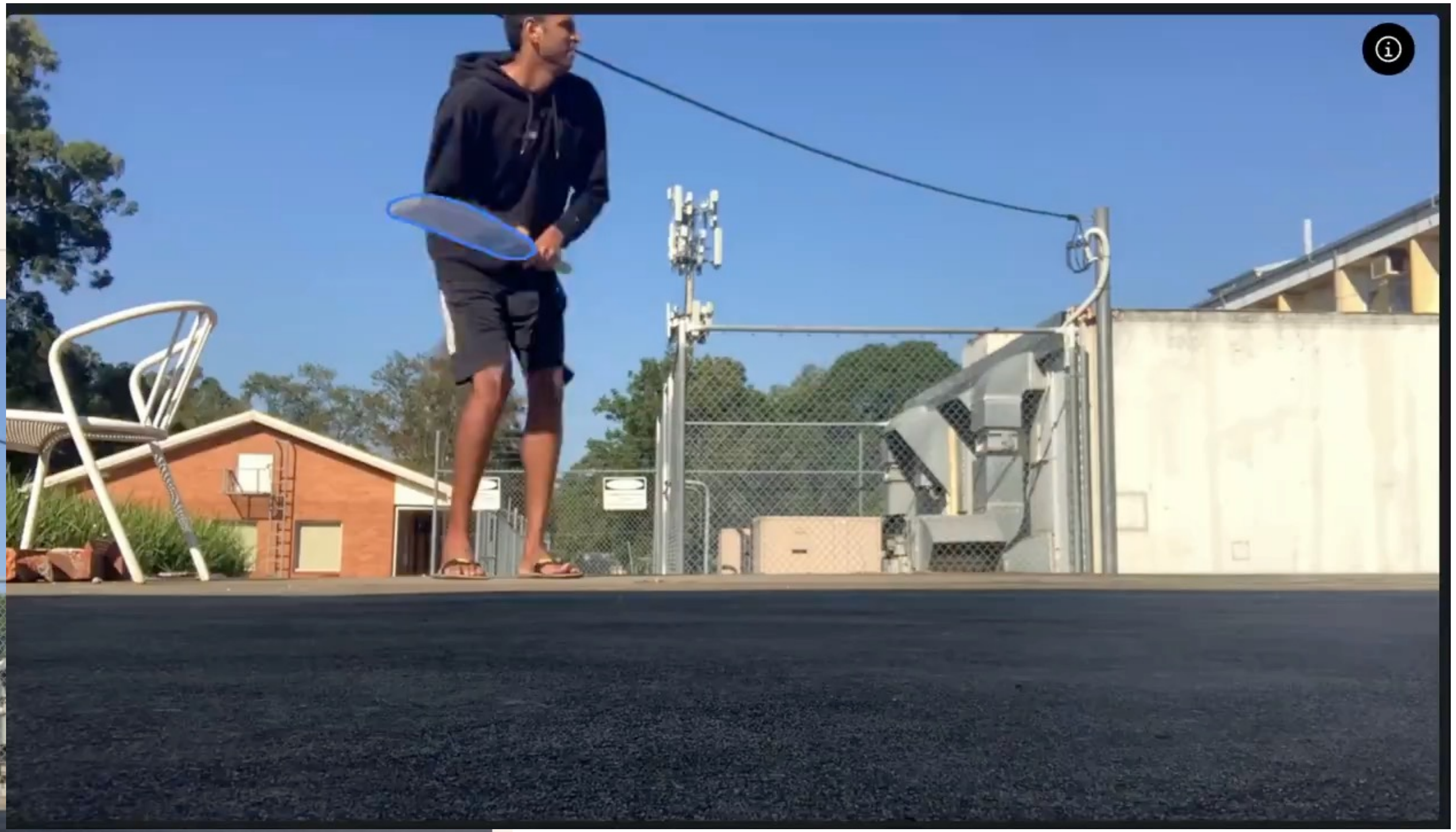
Counting the **people** shows a total of 40.

nting the **cars on right lane** shows a total of 2.

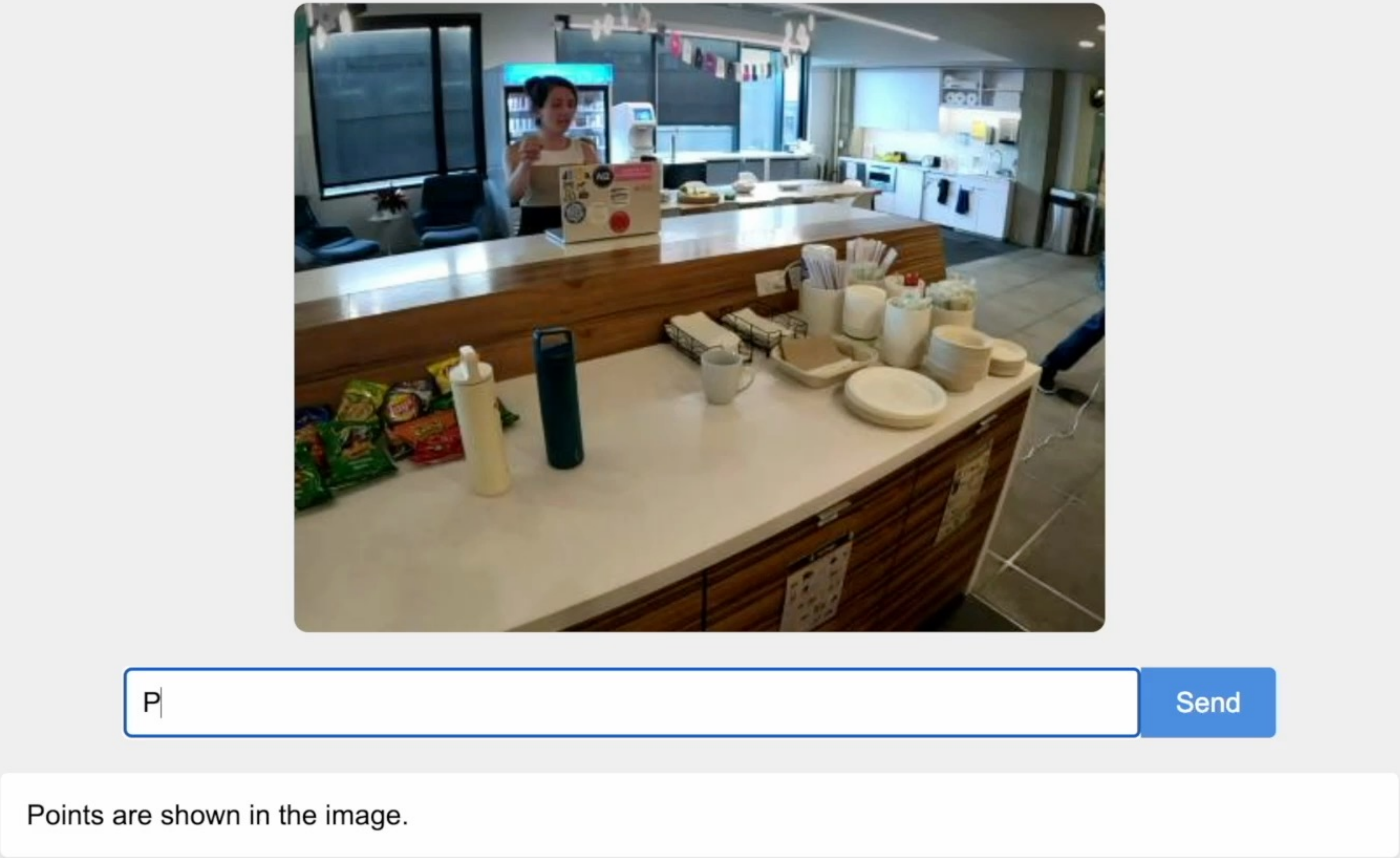
Chaining Molmo + SAM 2



can you point to the cricket bat



Future: Embodied AI for Navigation & Manipulation



P

Points are shown in the image.

Demo
molmo.allenai.org

Foundation Models

Language

- **ELMo**
- **BERT**
- **GPT**
- **T5**

Classification

- **CLIP**
- **CoCa**

LM + Vision

- **LLaVA**
- **Flamingo**
- GPT-4V
- Gemini
- **Molmo**

And More!

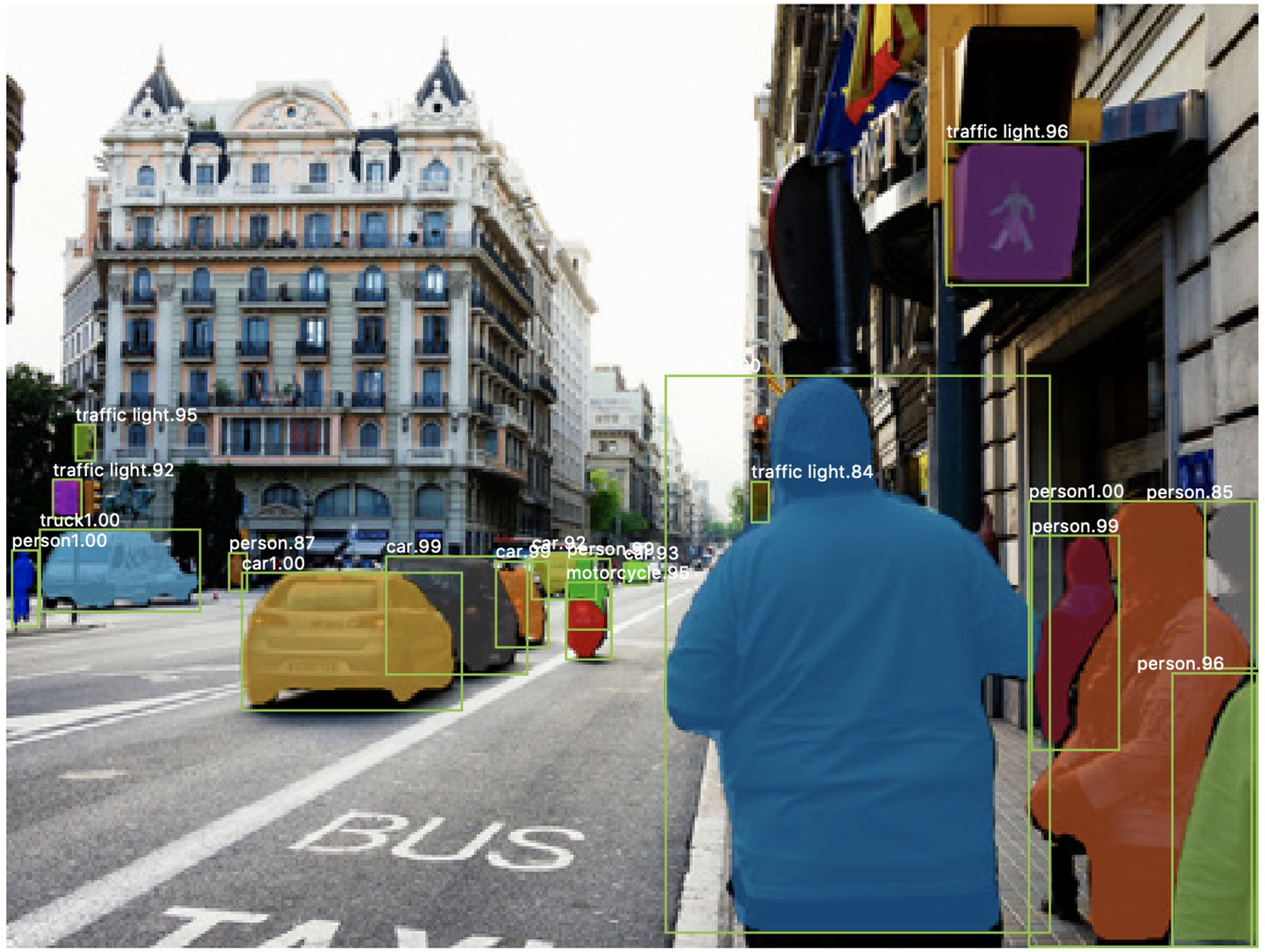
- **Segment Anything**
- Whisper
- Dalle
- Stable Diffusion
- Imagen

Chaining

- **LMs + CLIP**
- **Visual Programming**

Segment Anything Model (SAM)

What does it mean to have a segmentation foundation model?



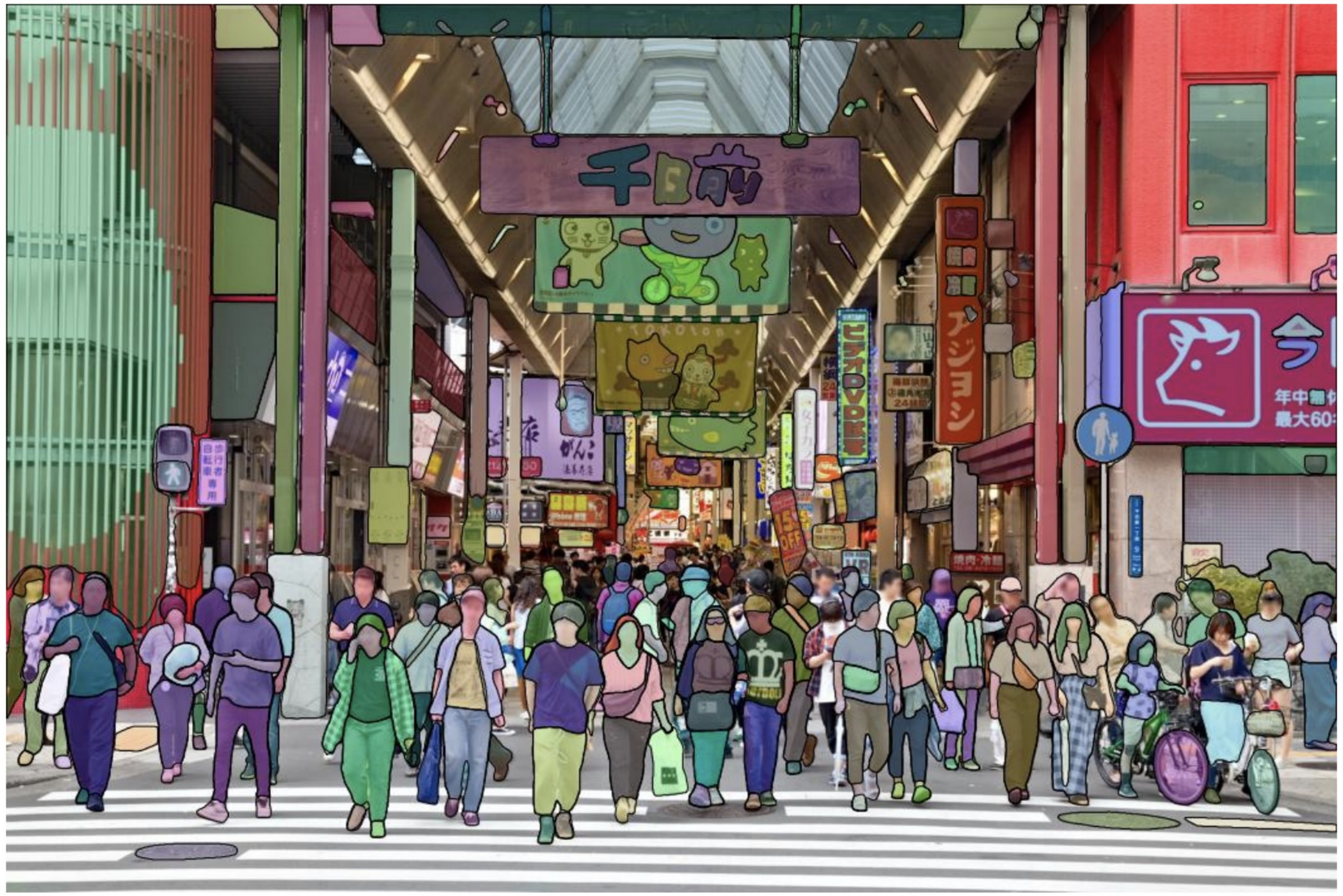
Masking model trained on dataset of specific number of objects (80 in COCO)

Model outputs masks of all objects in that image that is one of the categories of interest

Images: He et al. Mask R-CNN. 2017

Segment Anything Model (SAM)

What does it mean to have a segmentation foundation model?



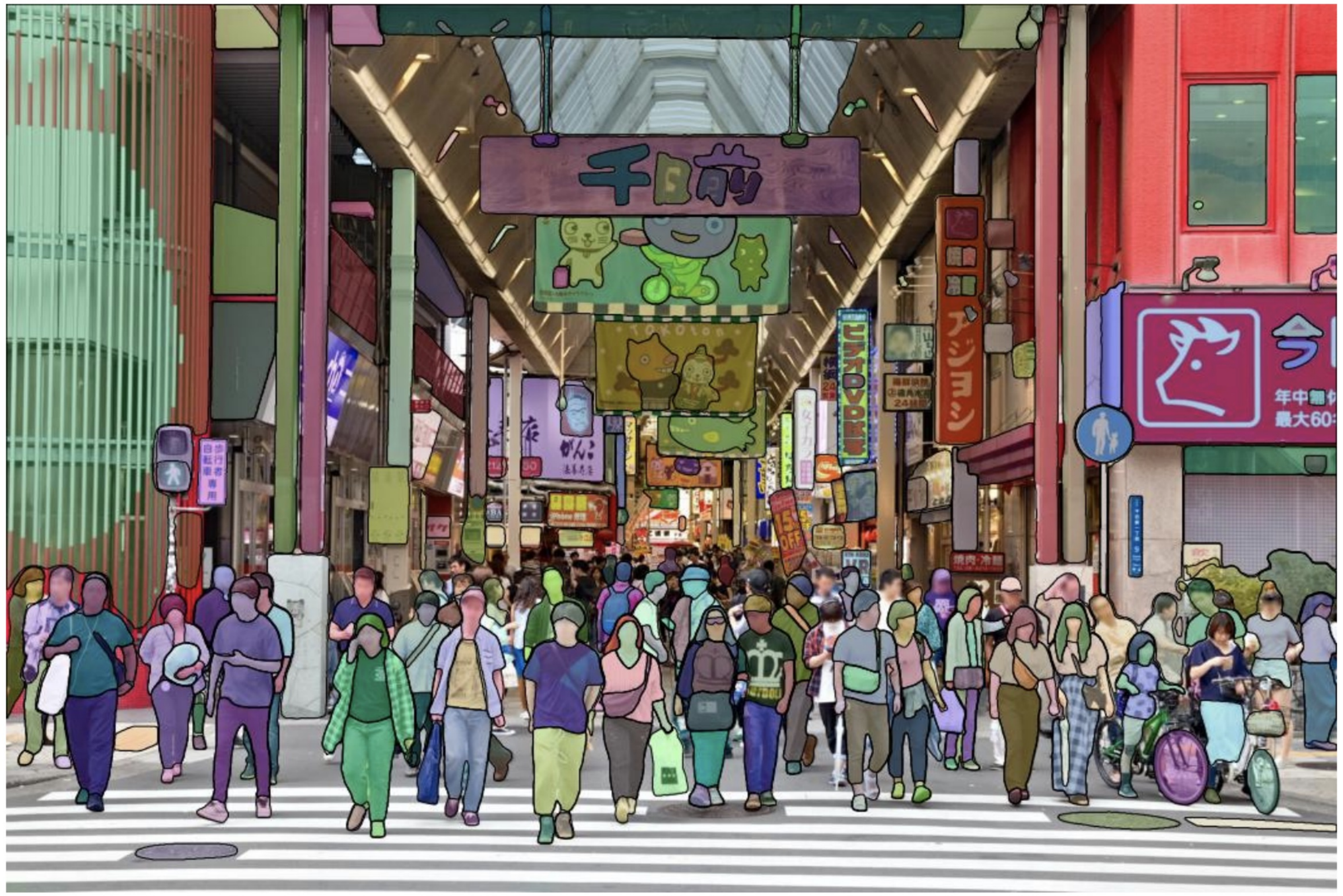
Masking model trained on a dataset of a huge number of categories

Model outputs mask of any objects that the user cares about

Images: Kirillov et al. Segment Anything. 2023.

Segment Anything Model (SAM)

What does it mean to have a segmentation foundation model?



Masking model trained on a dataset of a huge number of categories

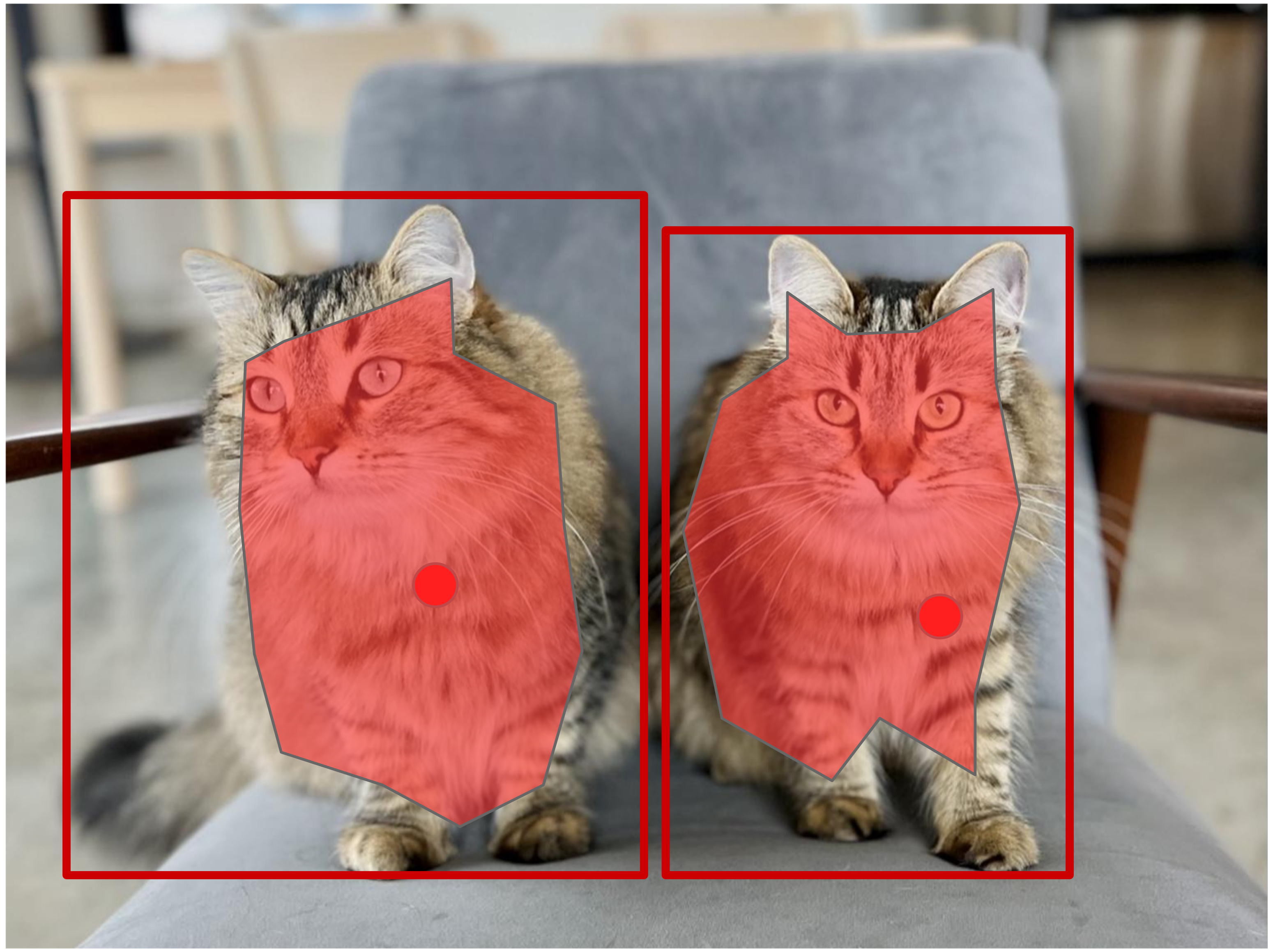
How to get this?

Model outputs mask of any objects that the user cares about

How to know this?

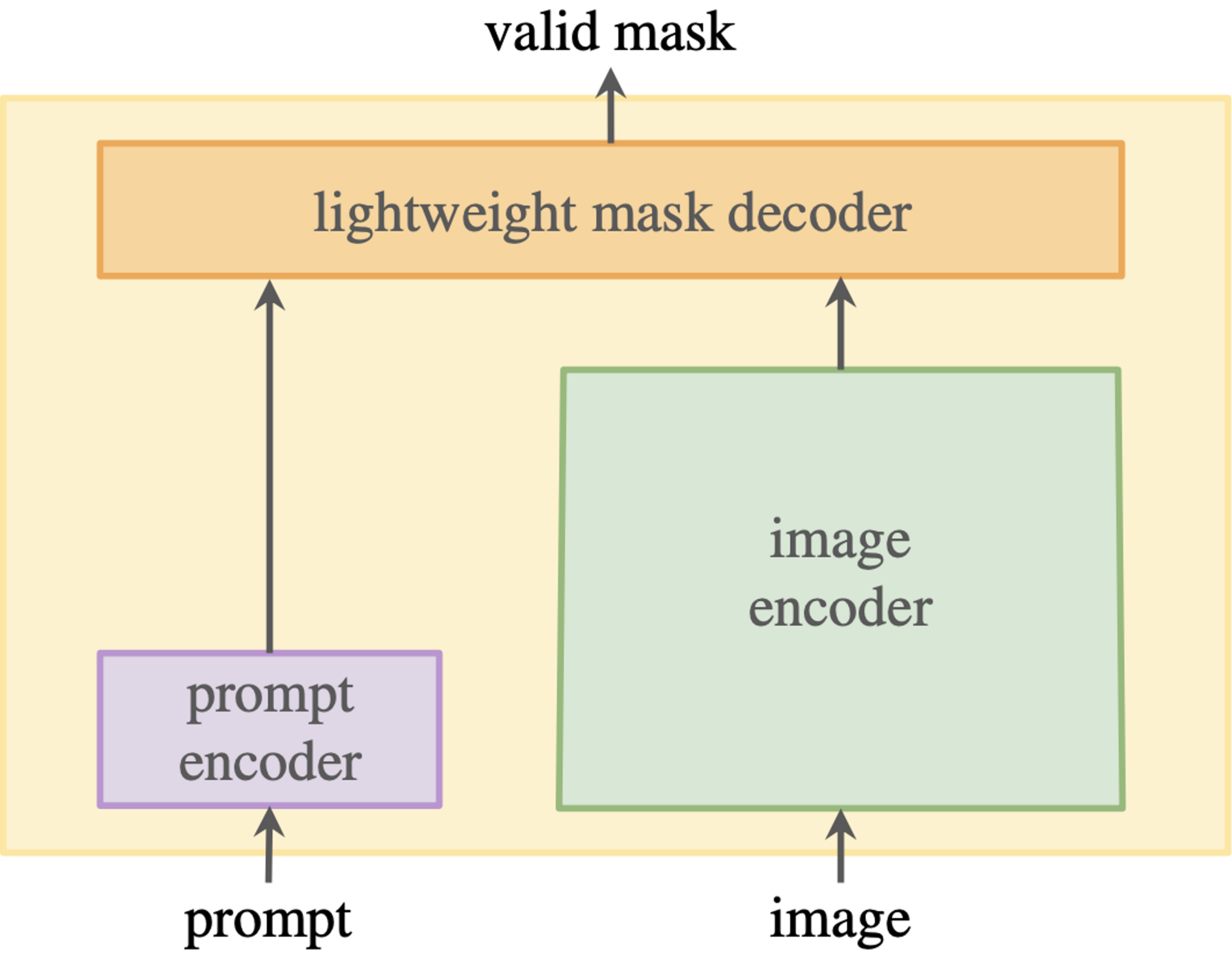
Images: Kirillov et al. Segment Anything. 2023.

How to know what to mask?



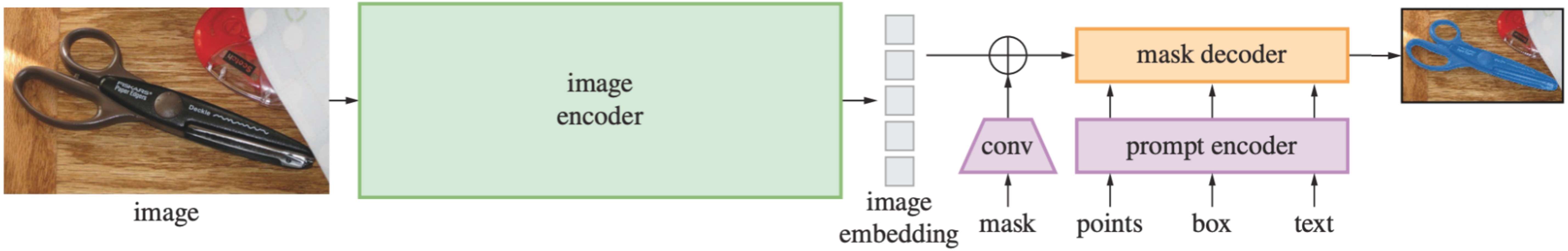
“Cats”

Basic SAM Architecture



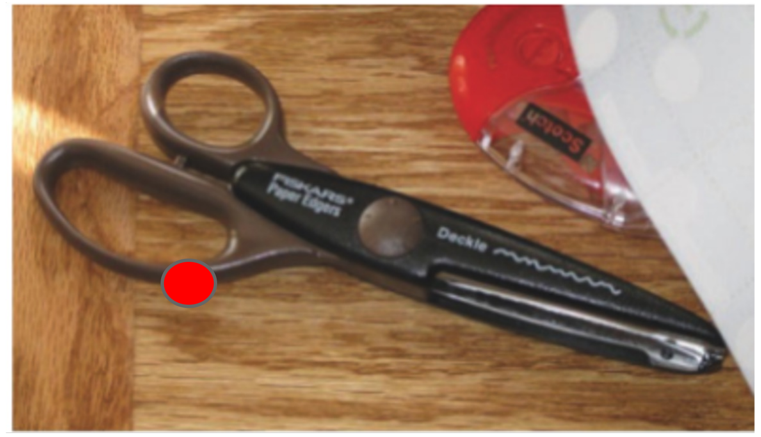
Images: Kirillov et al. Segment Anything. 2023.

SAM Architecture



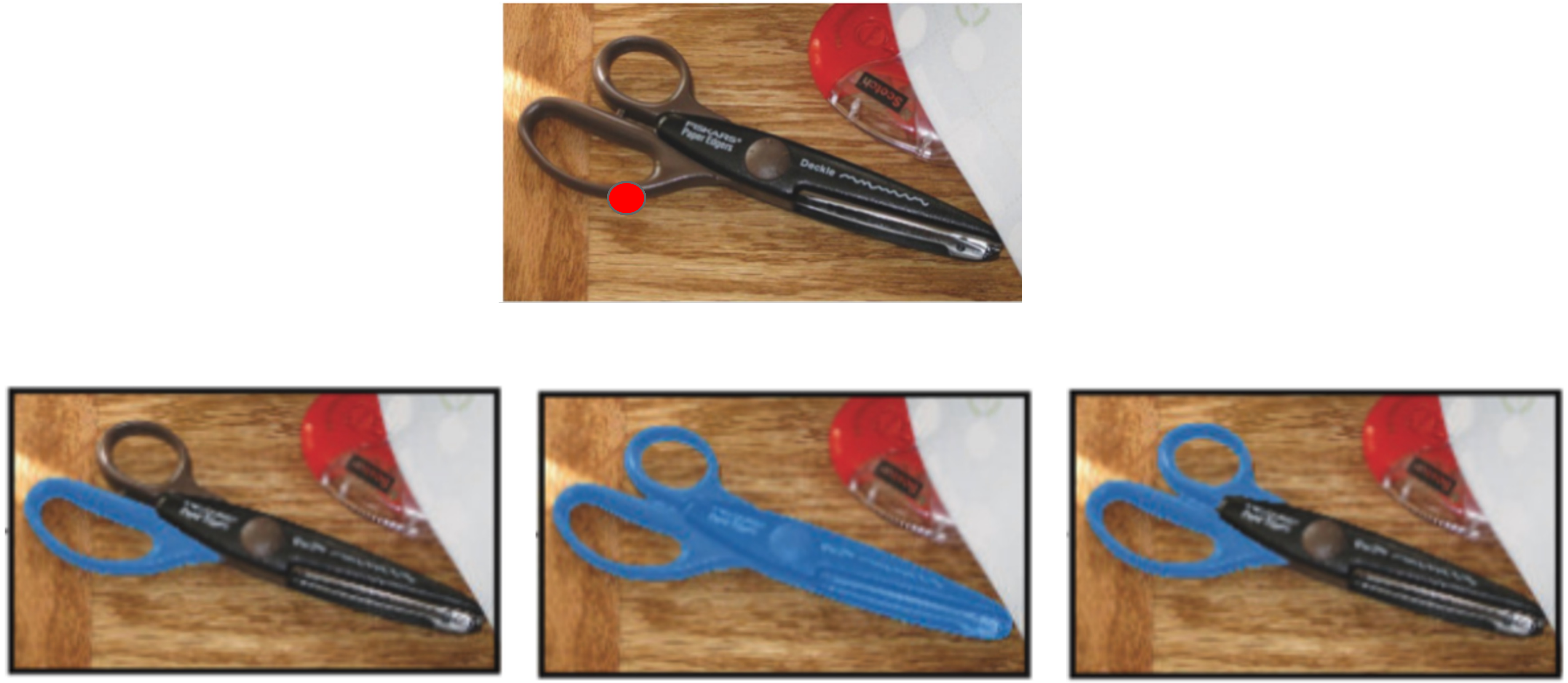
Images: Kirillov et al. Segment Anything. 2023.

Ambiguity in correct prompt



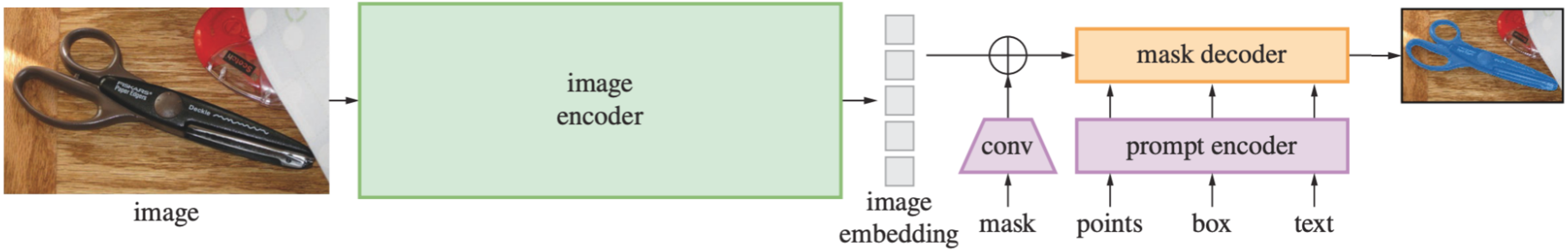
Images: Kirillov et al. Segment Anything. 2023.

Ambiguity in correct prompt



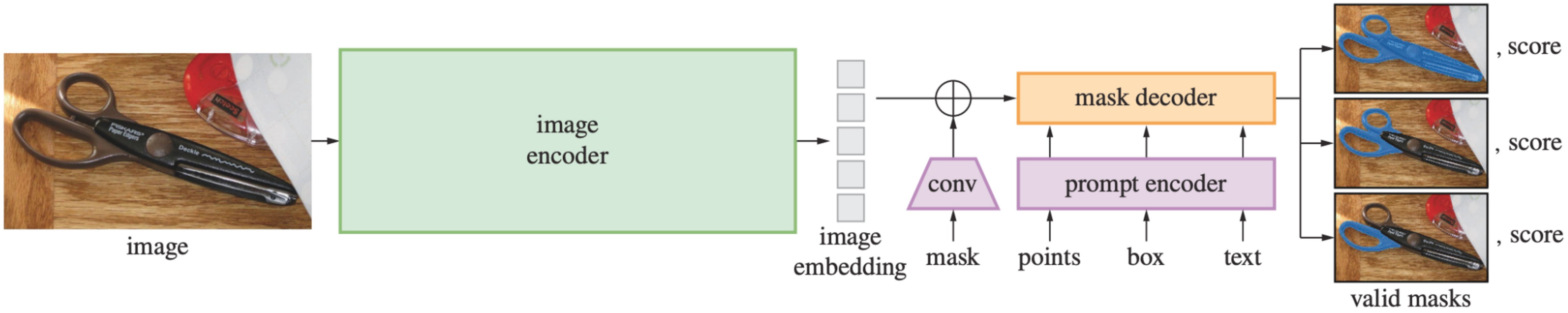
Images: Kirillov et al. Segment Anything. 2023.

SAM Architecture



Images: Kirillov et al. Segment Anything. 2023.

Basic SAM Architecture

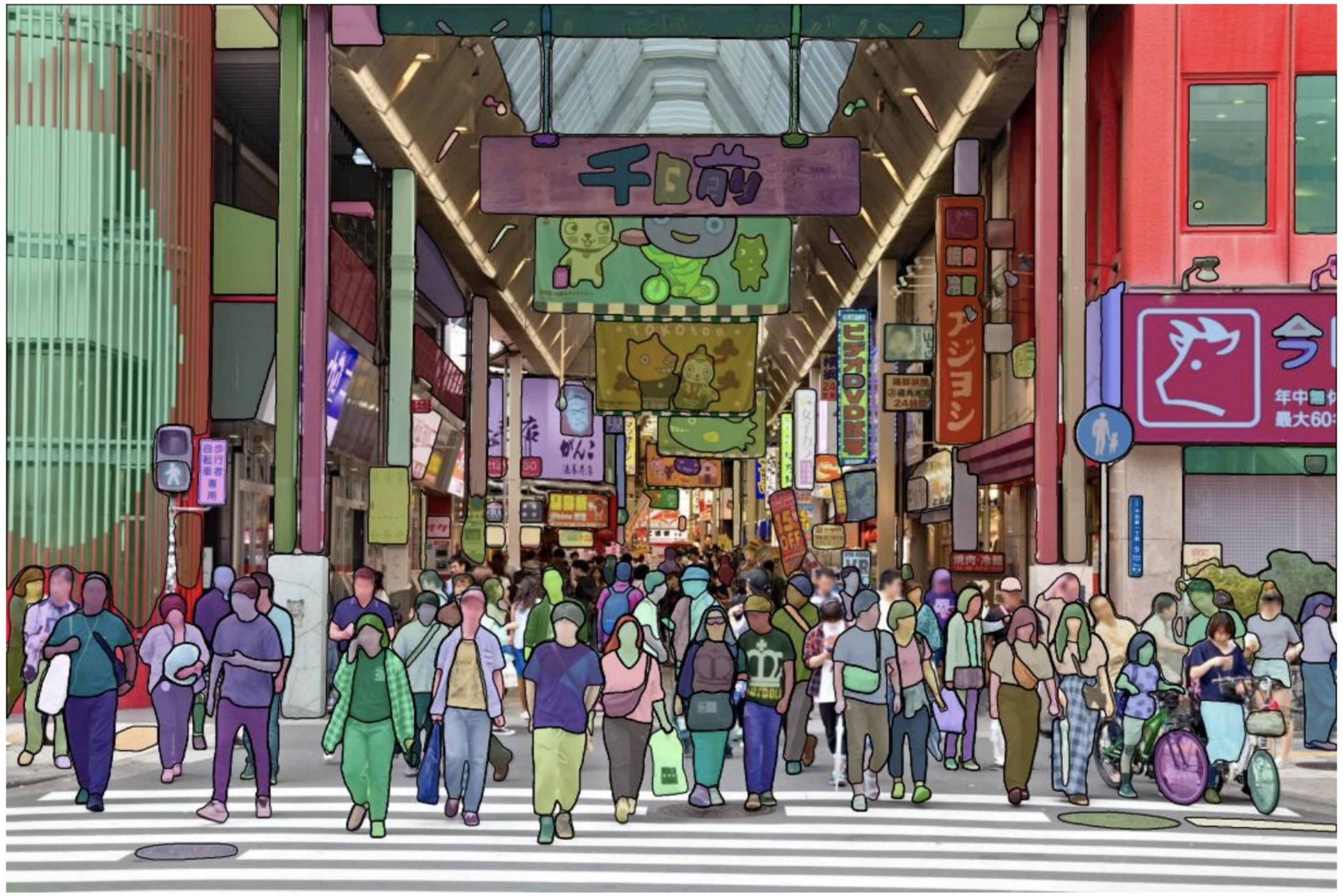


1. Loss only calculated with respect to best mask
2. Model also trained to output confidence score for each mask

Images: Kirillov et al. Segment Anything. 2023.

Segment Anything Model (SAM)

What does it mean to have a segmentation foundation model?



Masking model trained on a dataset of a huge number of categories

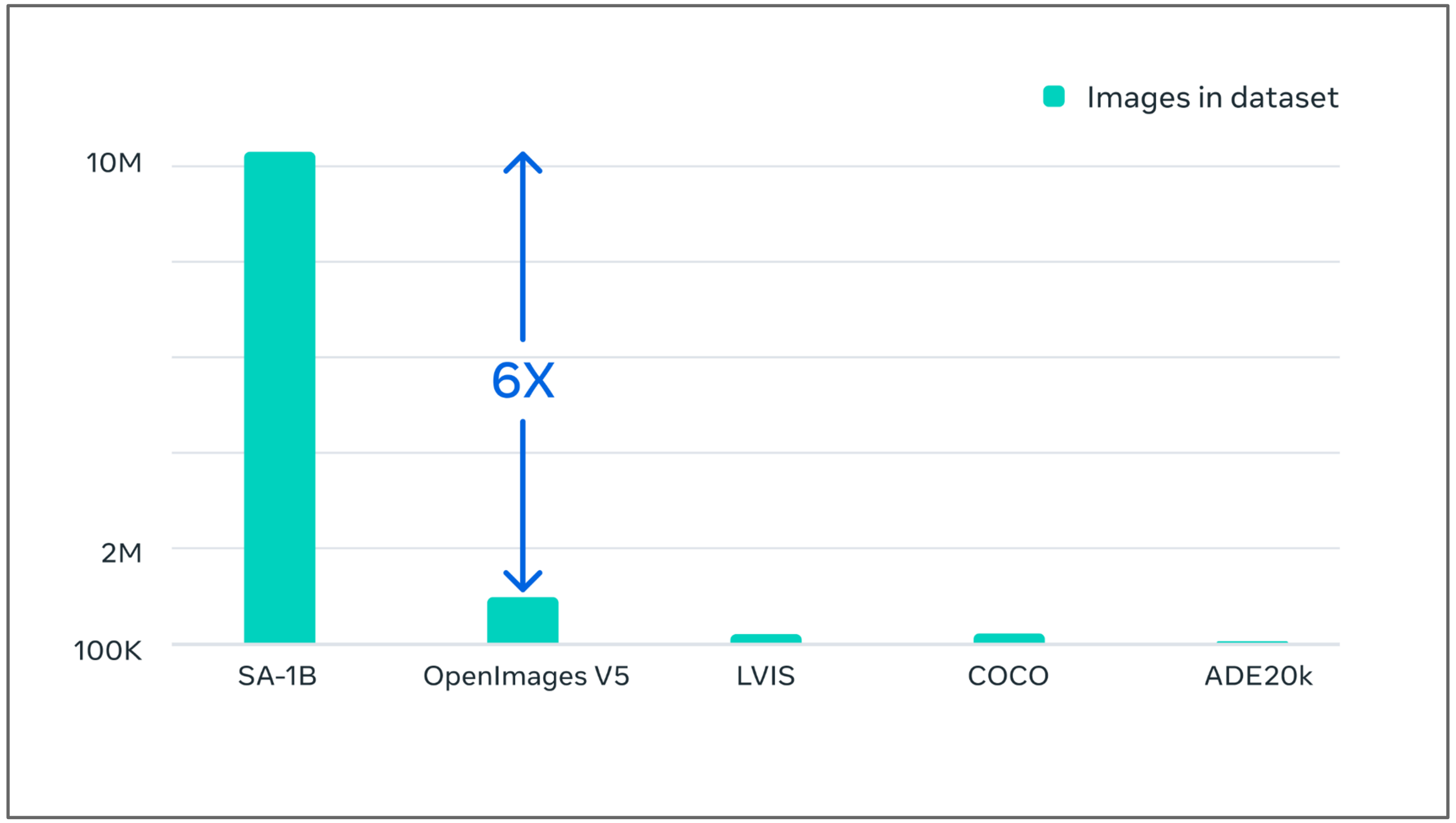
How to get this?

Model outputs mask of any objects that the user cares about

How to know this?

Images: Kirillov et al. Segment Anything. 2023.

Segment Anything Model (SAM)



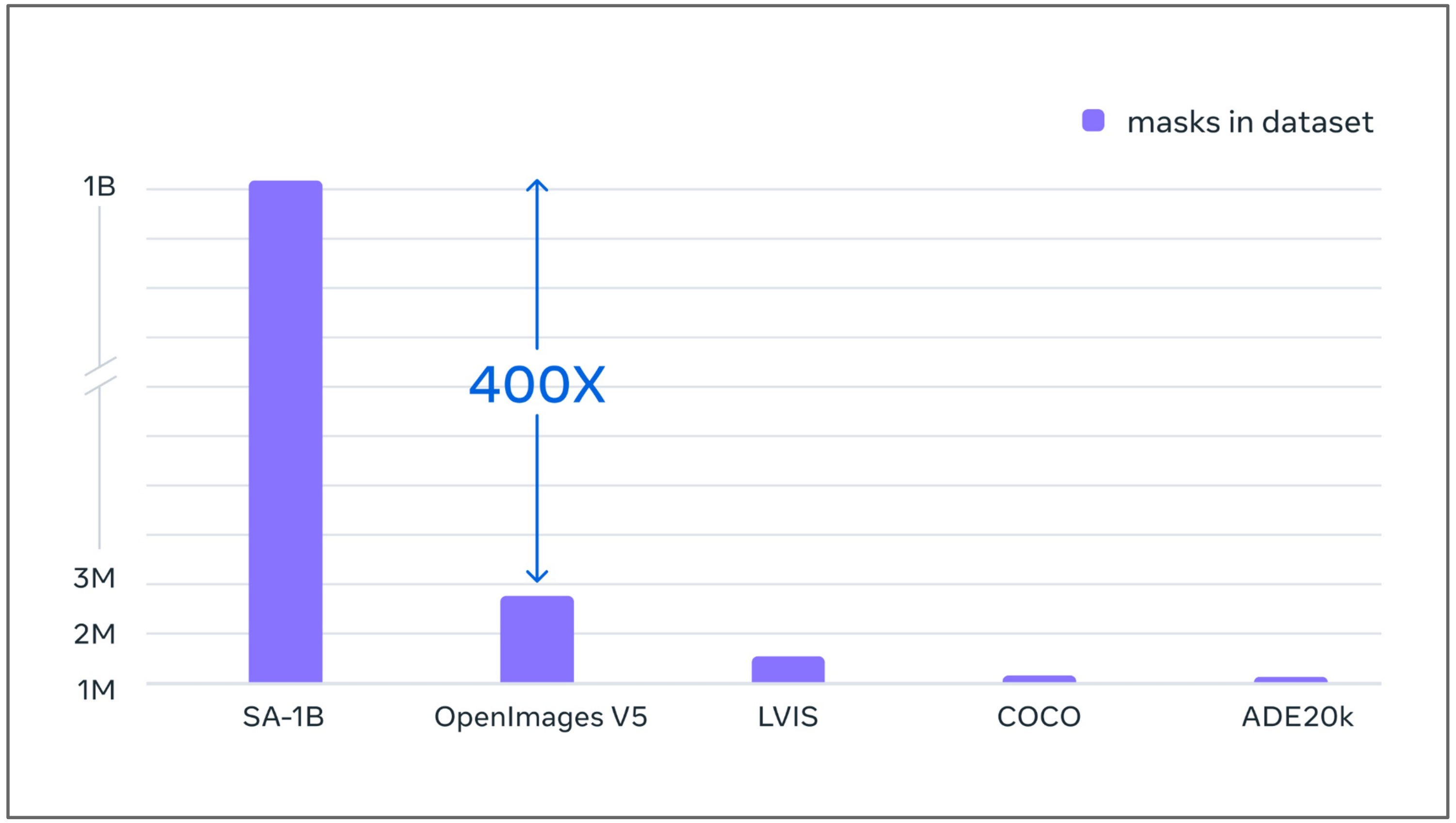
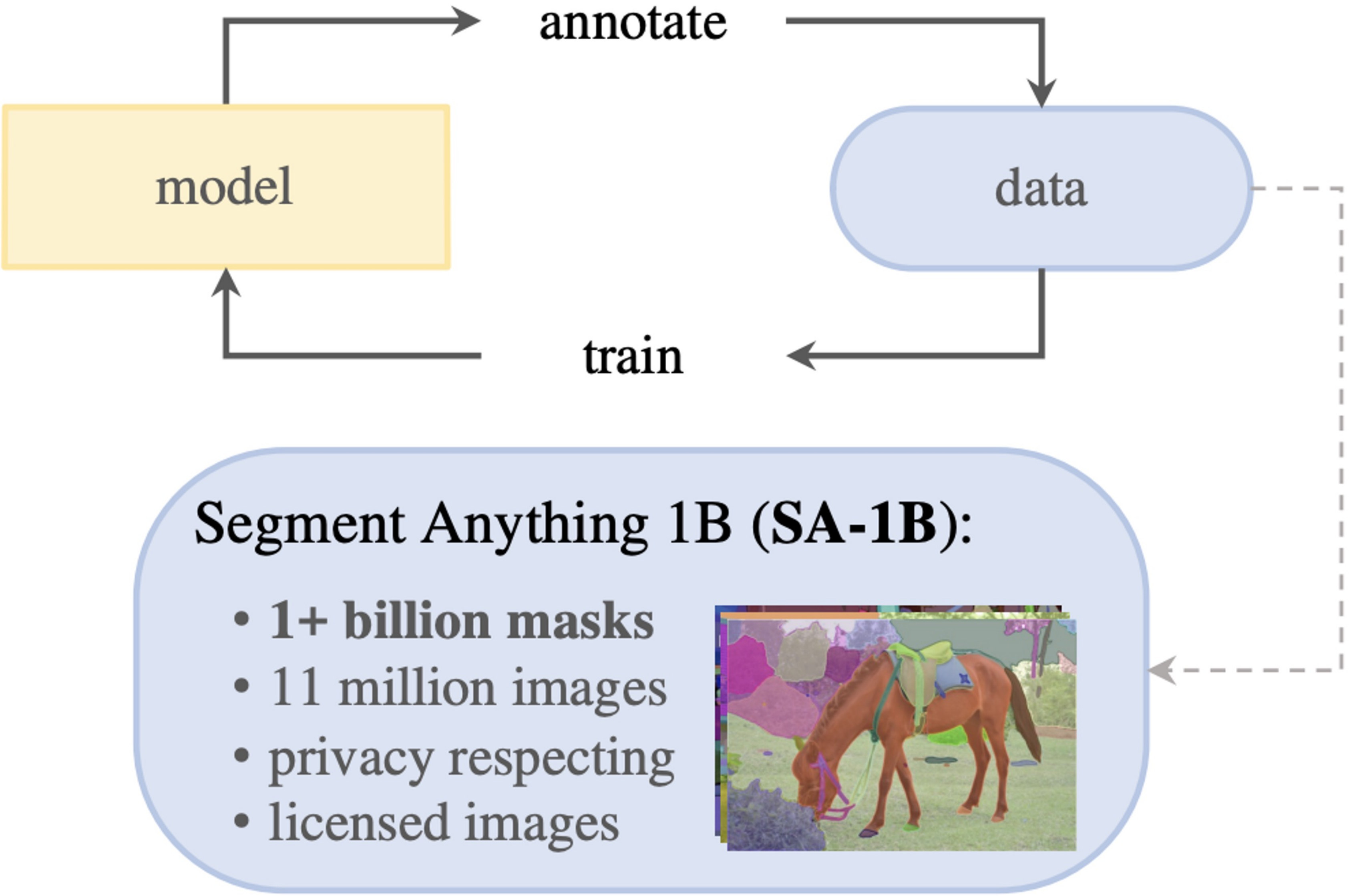


Image Source: https://segment-anything.com/

Segment Anything Model (SAM)



Segment Anything 1B (**SA-1B**):

- **1+ billion masks**
- 11 million images
- privacy respecting
- licensed images

Images: Kirillov et al. Segment Anything. 2023.

SAM Results

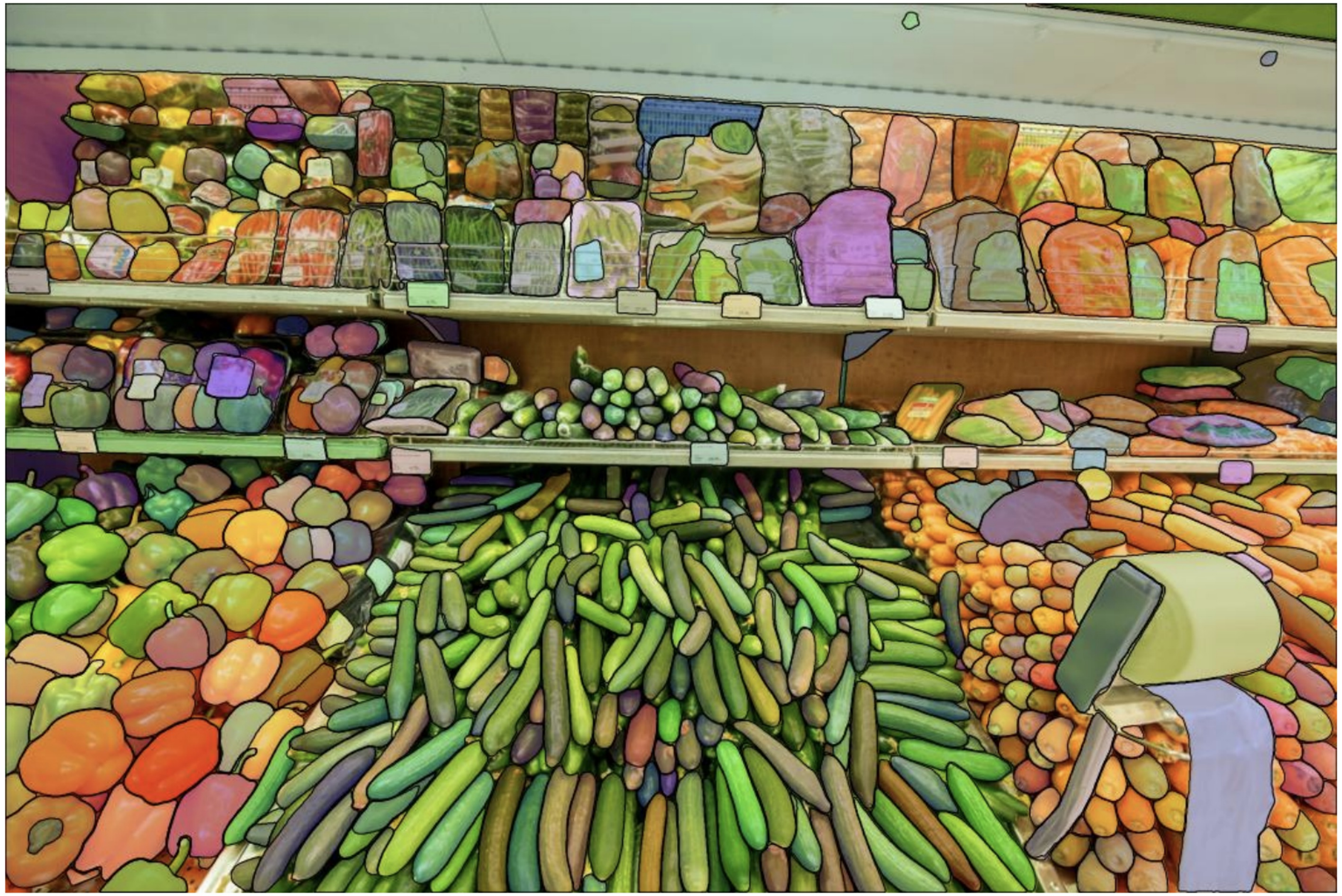


Image Source: Kirillov et al. Segment Anything. 2023

SAM Results

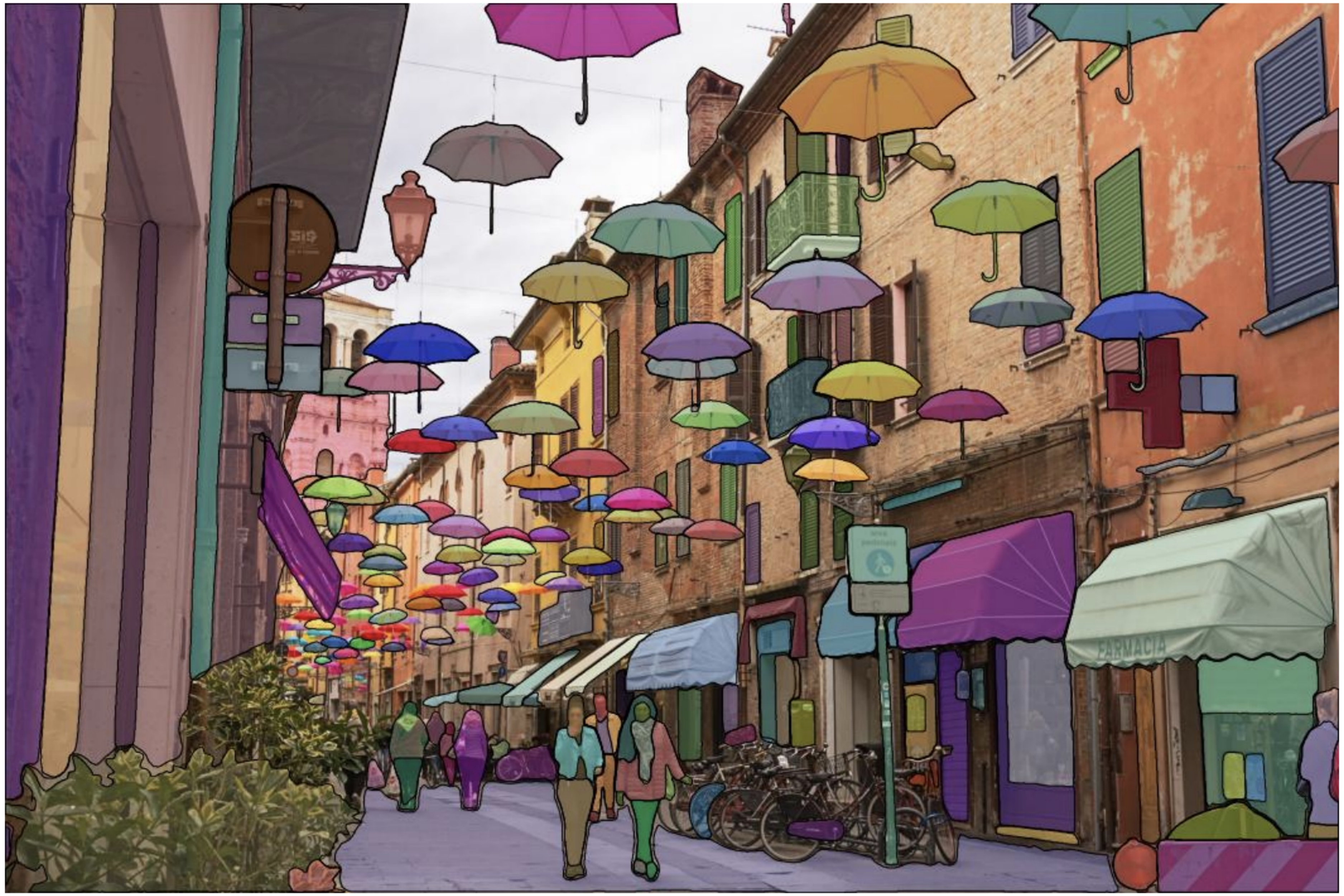


Image Source: Kirillov et al. Segment Anything. 2023

Zero-Shot with SAM

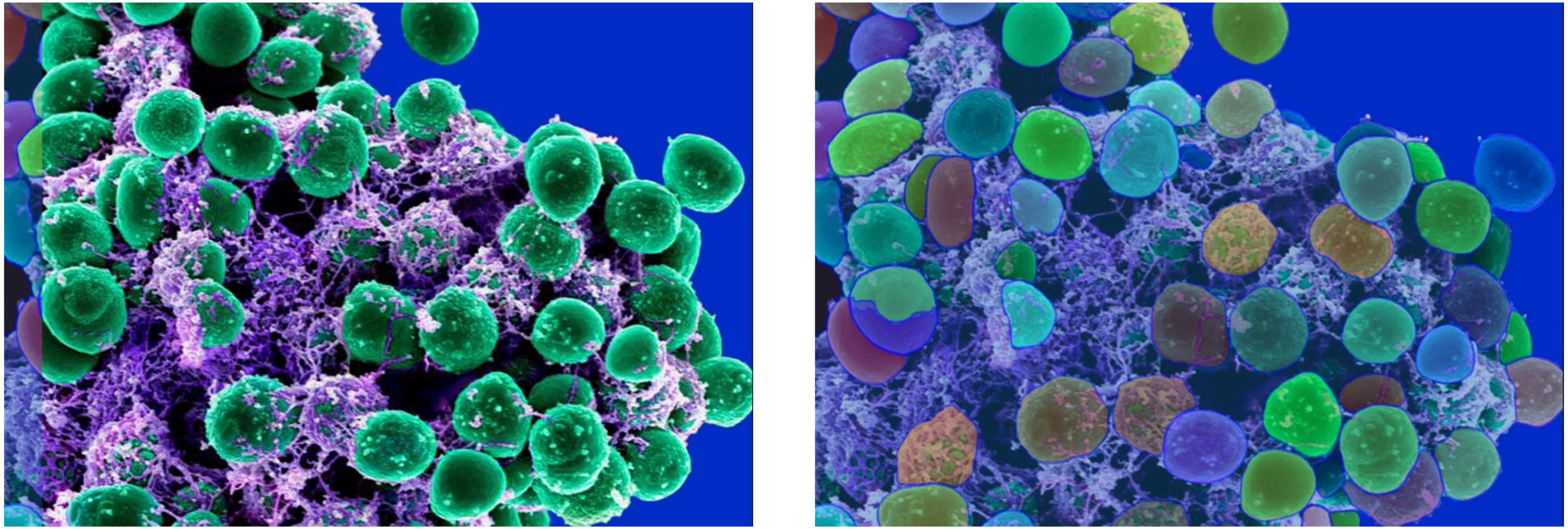


Image Source: https://segment-anything.com/

Zero-Shot with SAM

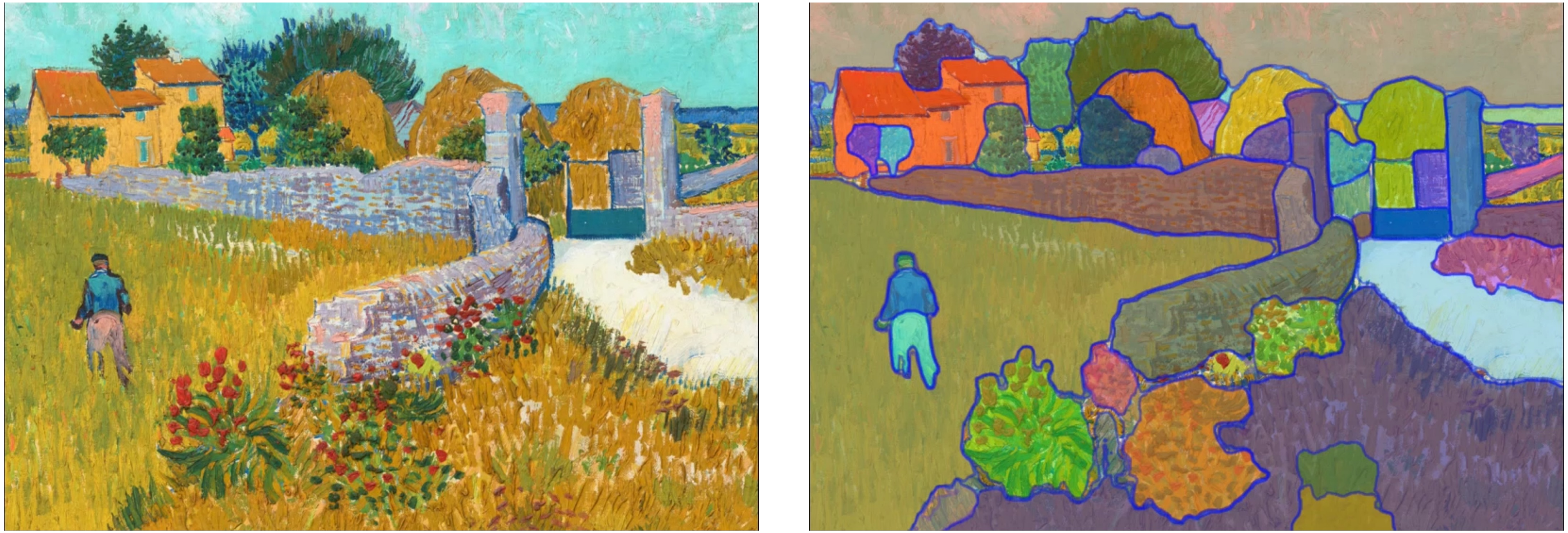


Image Source: https://segment-anything.com/

Foundation Models

Language

ELMo
BERT
GPT
T5

Classification

CLIP
CoCa

LM + Vision

LLaVA
Flamingo
GPT-4V
Gemini
Molmo

And More!

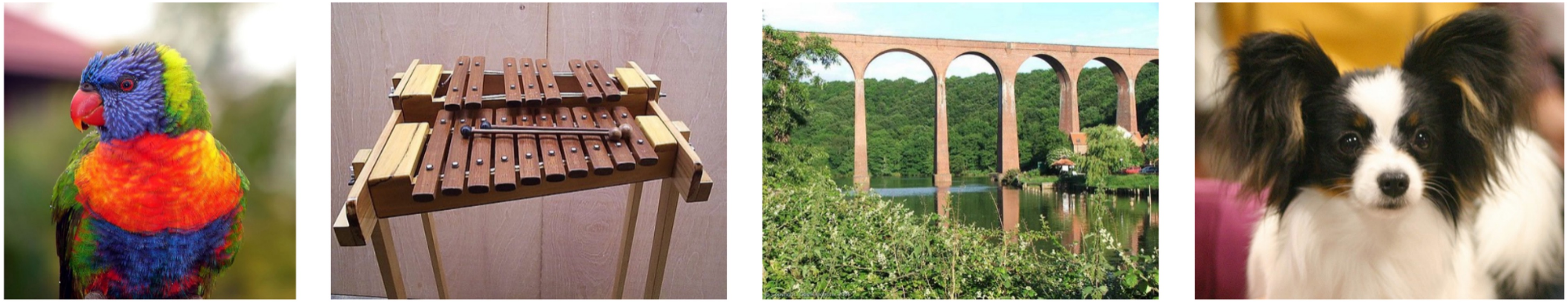
Segment Anything
Whisper
Dalle
Stable Diffusion
Imagen

Chaining

LMs + CLIP
Visual Programming

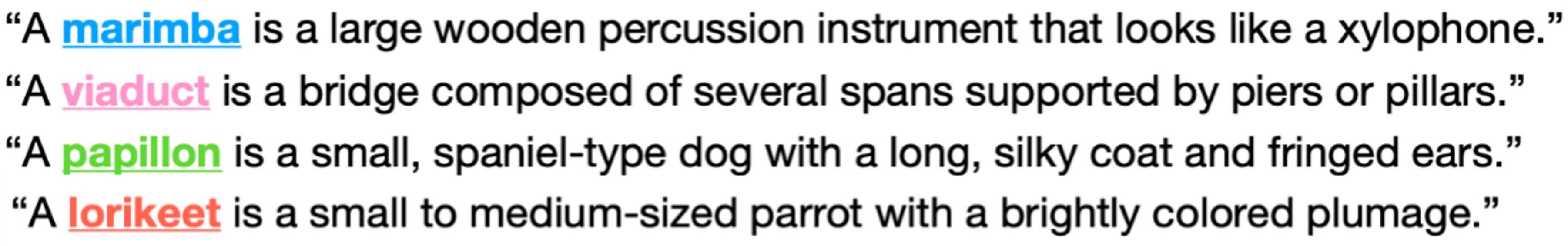
What happens when a model is asked to classify a concept it has never seen?

A photo of a marimba
A photo of a viaduct
A photo of a papillon
A photo of a lorikeet



Pratt et al “What does a platypus look like? Generating customized prompts for zero-shot image classification”. 2023.

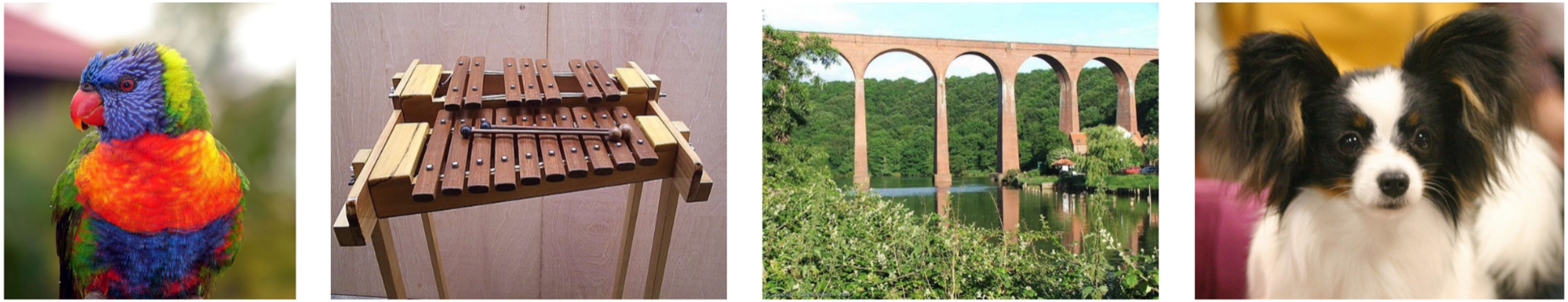
Solution: chaining
1. Get an LLM to generate a description.
2. Classify using the description


"A **marimba** is a large wooden percussion instrument that looks like a xylophone."

"A **viaduct** is a bridge composed of several spans supported by piers or pillars."

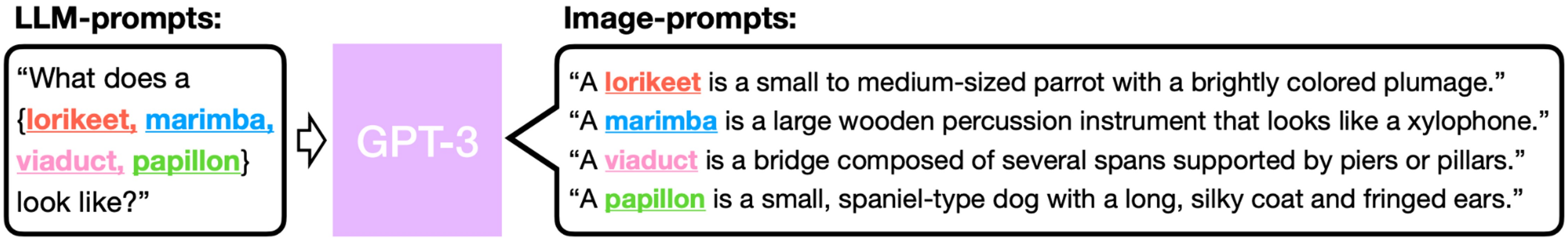
"A **papillon** is a small, spaniel-type dog with a long, silky coat and fringed ears."

"A **lorikeet** is a small to medium-sized parrot with a brightly colored plumage."



Pratt et al "What does a platypus look like? Generating customized prompts for zero-shot image classification". 2023.

CuPL (CUstomized Prompts via Language models)



LLM-prompts:

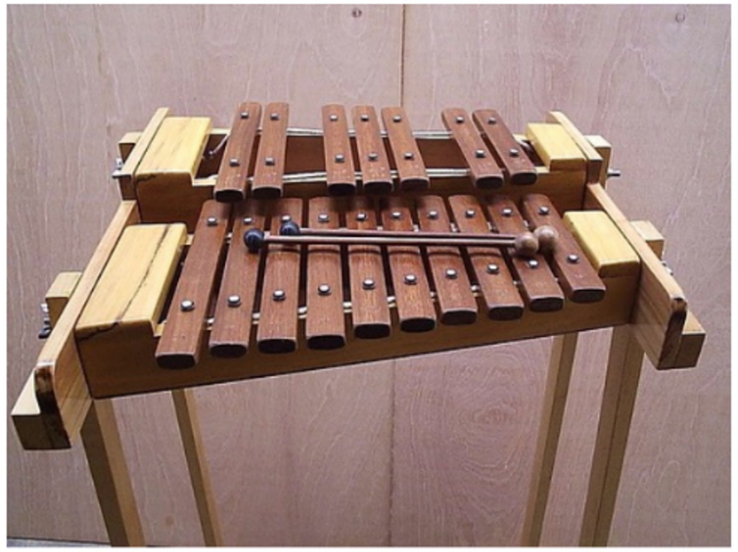
"What does a {lorikeet, marimba, viaduct, papillon} look like?"

GPT-3

Image-prompts:

"A lorikeet is a small to medium-sized parrot with a brightly colored plumage."
"A marimba is a large wooden percussion instrument that looks like a xylophone."
"A viaduct is a bridge composed of several spans supported by piers or pillars."
"A papillon is a small, spaniel-type dog with a long, silky coat and fringed ears."

Lorikeet



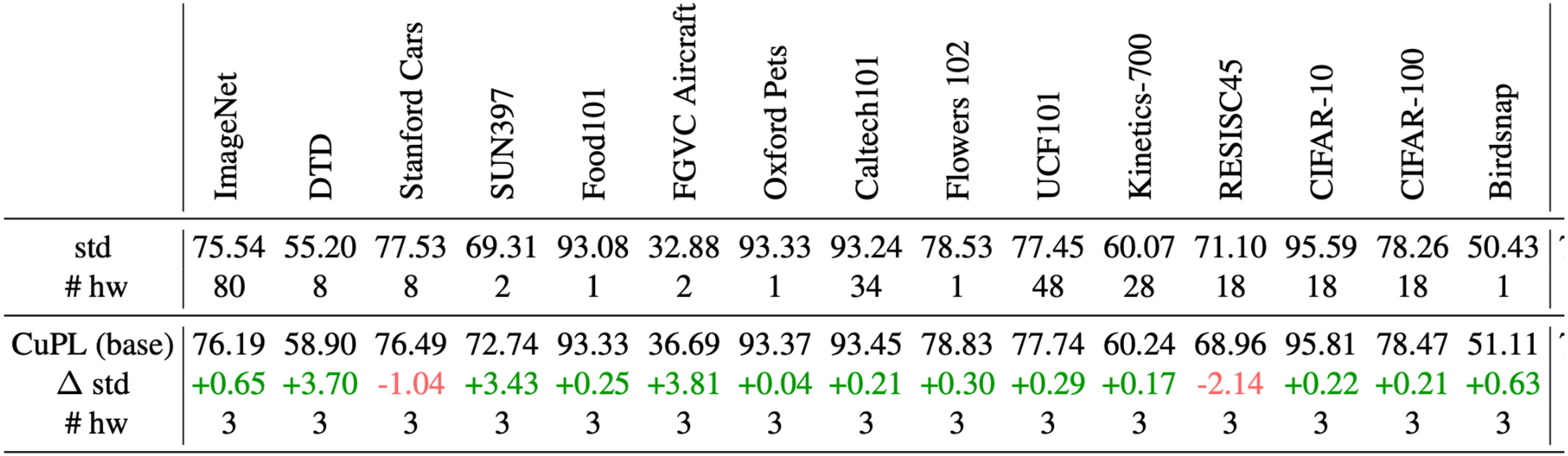
Marimba

Viaduct

Papillon

Pratt et al "What does a platypus look like? Generating customized prompts for zero-shot image classification". 2023.

CuPL (CUstomized Prompts via Language models)

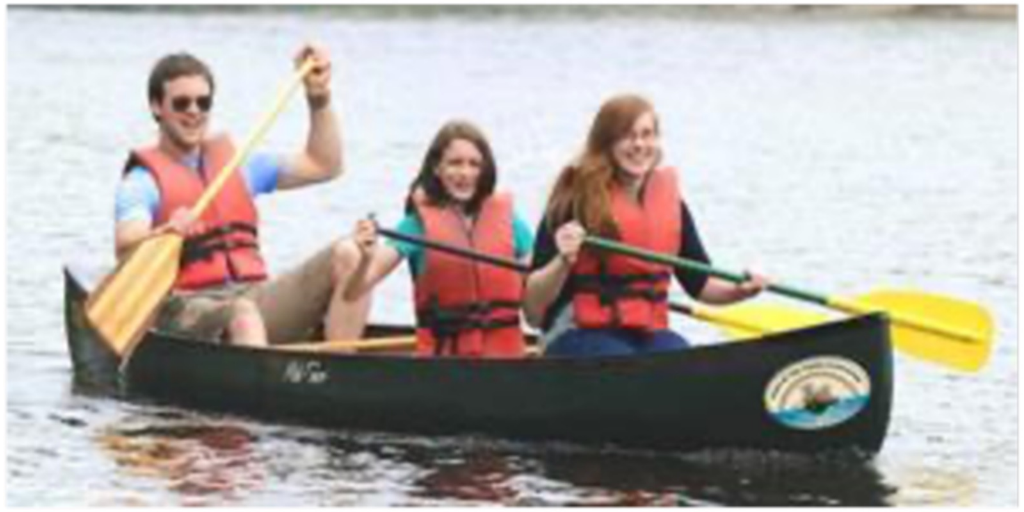


	ImageNet	DTD	Stanford Cars	SUN397	Food101	FGVC Aircraft	Oxford Pets	Caltech101	Flowers 102	UCF101	Kinetics-700	RESISC45	CIFAR-10	CIFAR-100	Birdsnap
std	75.54	55.20	77.53	69.31	93.08	32.88	93.33	93.24	78.53	77.45	60.07	71.10	95.59	78.26	50.43
# hw	80	8	8	2	1	2	1	34	1	48	28	18	18	18	1
CuPL (base)	76.19	58.90	76.49	72.74	93.33	36.69	93.37	93.45	78.83	77.74	60.24	68.96	95.81	78.47	51.11
Δ std	+0.65	+3.70	-1.04	+3.43	+0.25	+3.81	+0.04	+0.21	+0.30	+0.29	+0.17	-2.14	+0.22	+0.21	+0.63
# hw	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Pratt et al "What does a platypus look like? Generating customized prompts for zero-shot image classification". 2023.

Can we generalize the idea of chaining to all vision tasks?

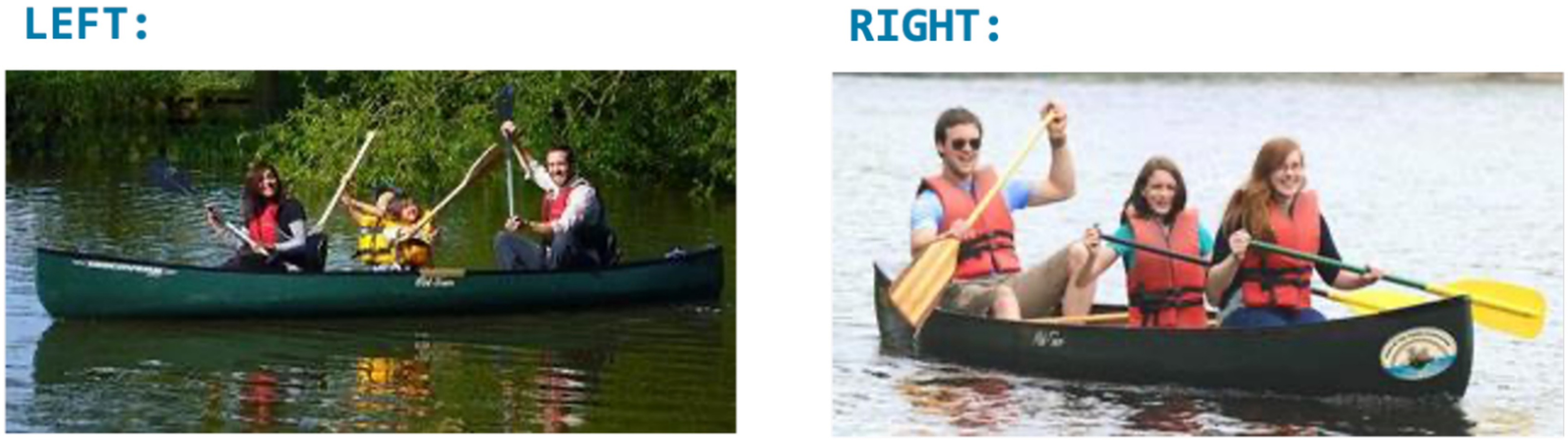
Many Visual Question Answering models which have been trained to do this type of task



Are there 3 people in the boat?

Gupta et al "Visual Programming: Compositional visual reasoning without training". 2023.

VisProg (visual programming)

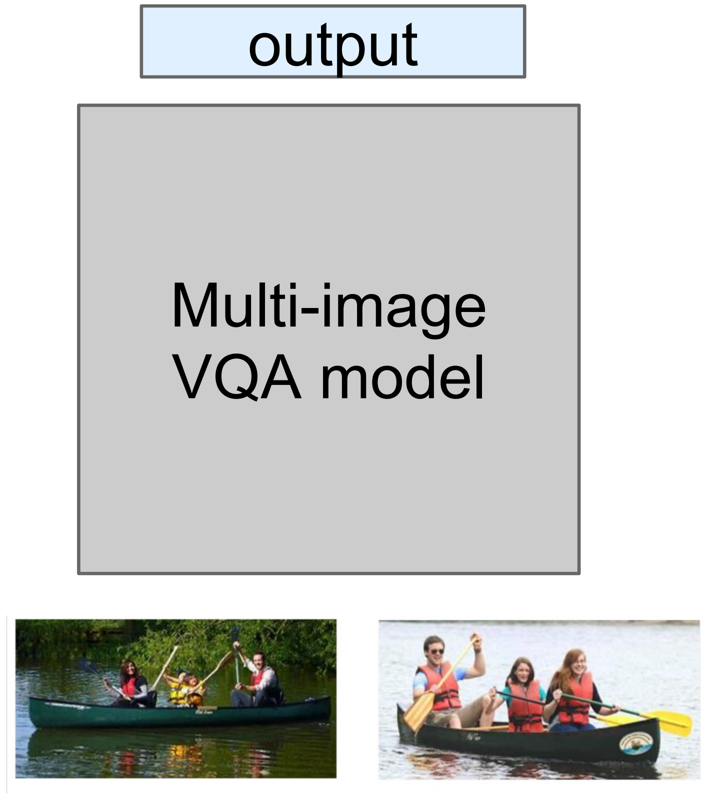


Statement: `The left and right image contains a total of six people and two boats.`

Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)

Train a new model for your task



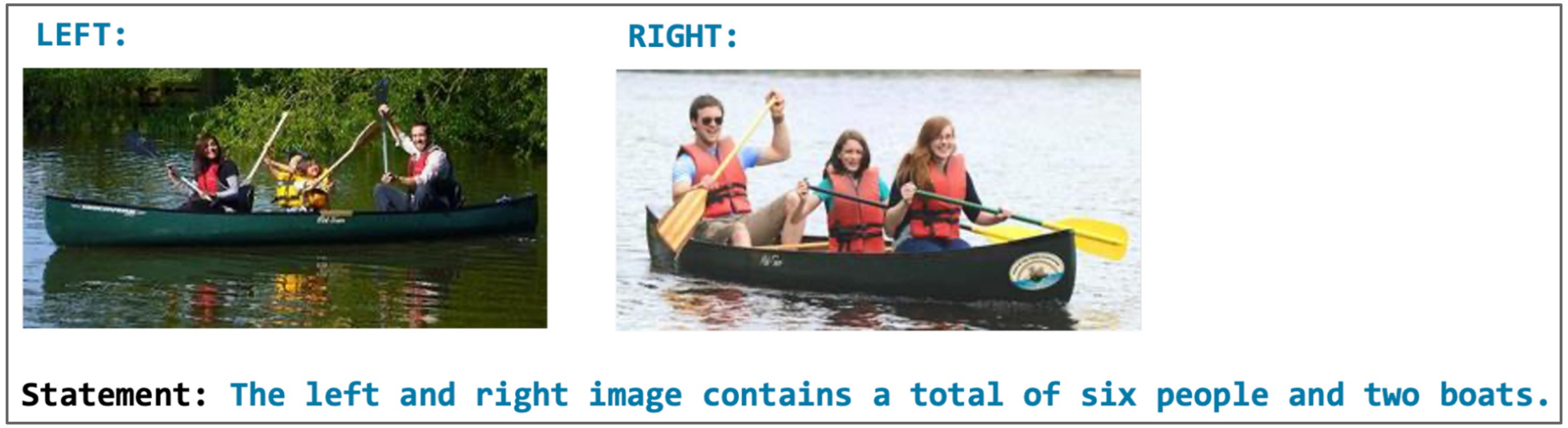
Write a python script with the models you have

```
Class MyMultiImageVQA():

  Def ProcessIms():
    Ans1 = VQA(Image1)
    Ans2 = VQA(Image2)
    Return Ans1 + Ans2
```

General to 2 images now, but not beyond that

Gupta et al "Visual Programming: Compositional visual reasoning without training". 2023.

VisProg (visual programming)



Statement: The left and right image contains a total of six people and two boats.

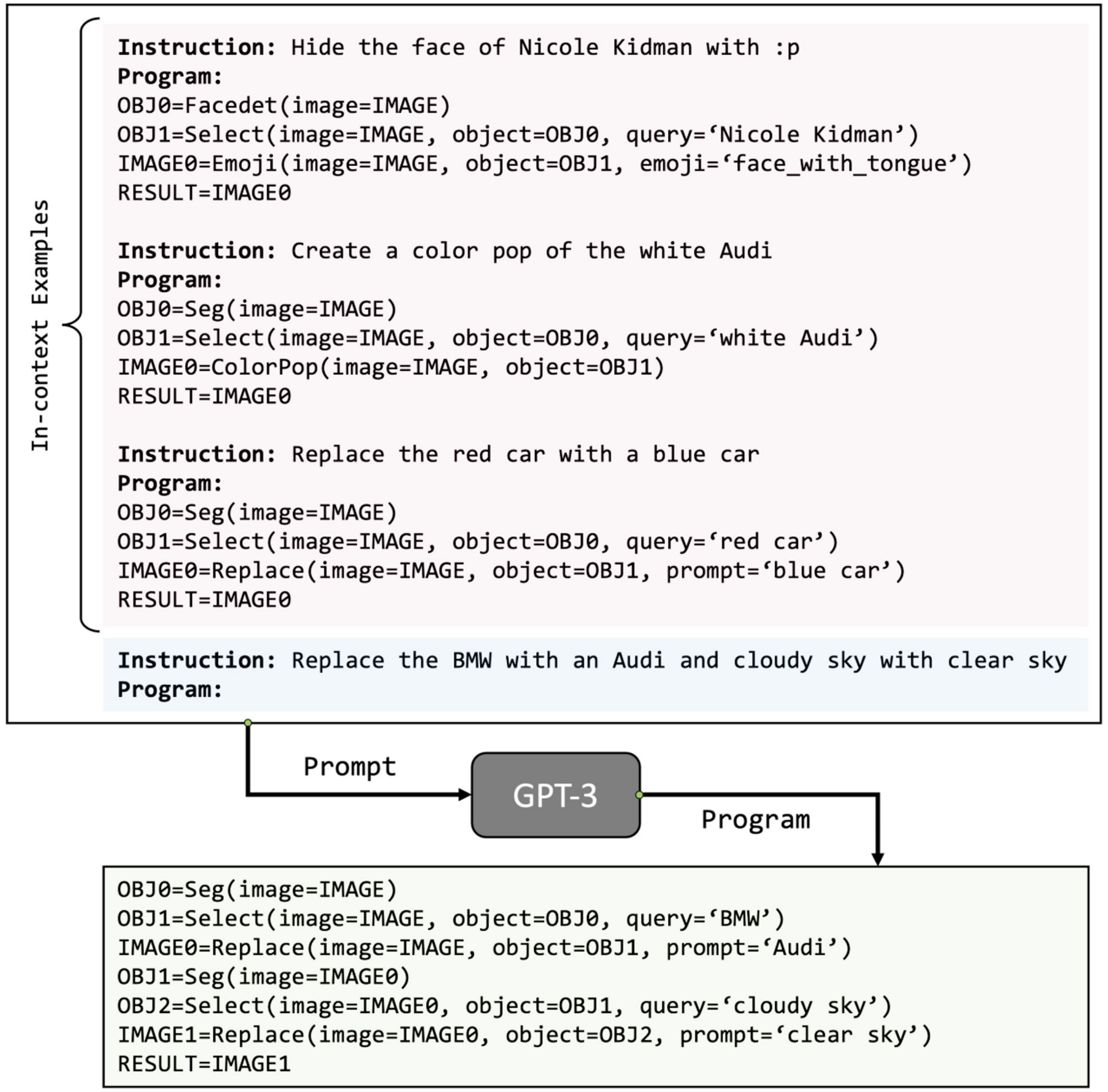
GPT

```
Class MyMultiImageVQA():

  Def ProcessIms():
    Ans1 = VQA(Image1)
    Ans2 = VQA(Image2)
    Return Ans1 + Ans2
```

False

Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)



```
Instruction: Hide the face of Nicole Kidman with :p
Program:
OBJ0=Facedet(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query=‘Nicole Kidman’)
IMAGE0=Emoji(image=IMAGE, object=OBJ1, emoji=‘face_with_tongue’)
RESULT=IMAGE0

Instruction: Create a color pop of the white Audi
Program:
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query=‘white Audi’)
IMAGE0=ColorPop(image=IMAGE, object=OBJ1)
RESULT=IMAGE0

Instruction: Replace the red car with a blue car
Program:
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query=‘red car’)
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt=‘blue car’)
RESULT=IMAGE0

Instruction: Replace the BMW with an Audi and cloudy sky with clear sky
Program:
```

```
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query=‘BMW’)
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt=‘Audi’)
OBJ1=Seg(image=IMAGE0)
OBJ2=Select(image=IMAGE0, object=OBJ1, query=‘cloudy sky’)
IMAGE1=Replace(image=IMAGE0, object=OBJ2, prompt=‘clear sky’)
RESULT=IMAGE1
```

Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)

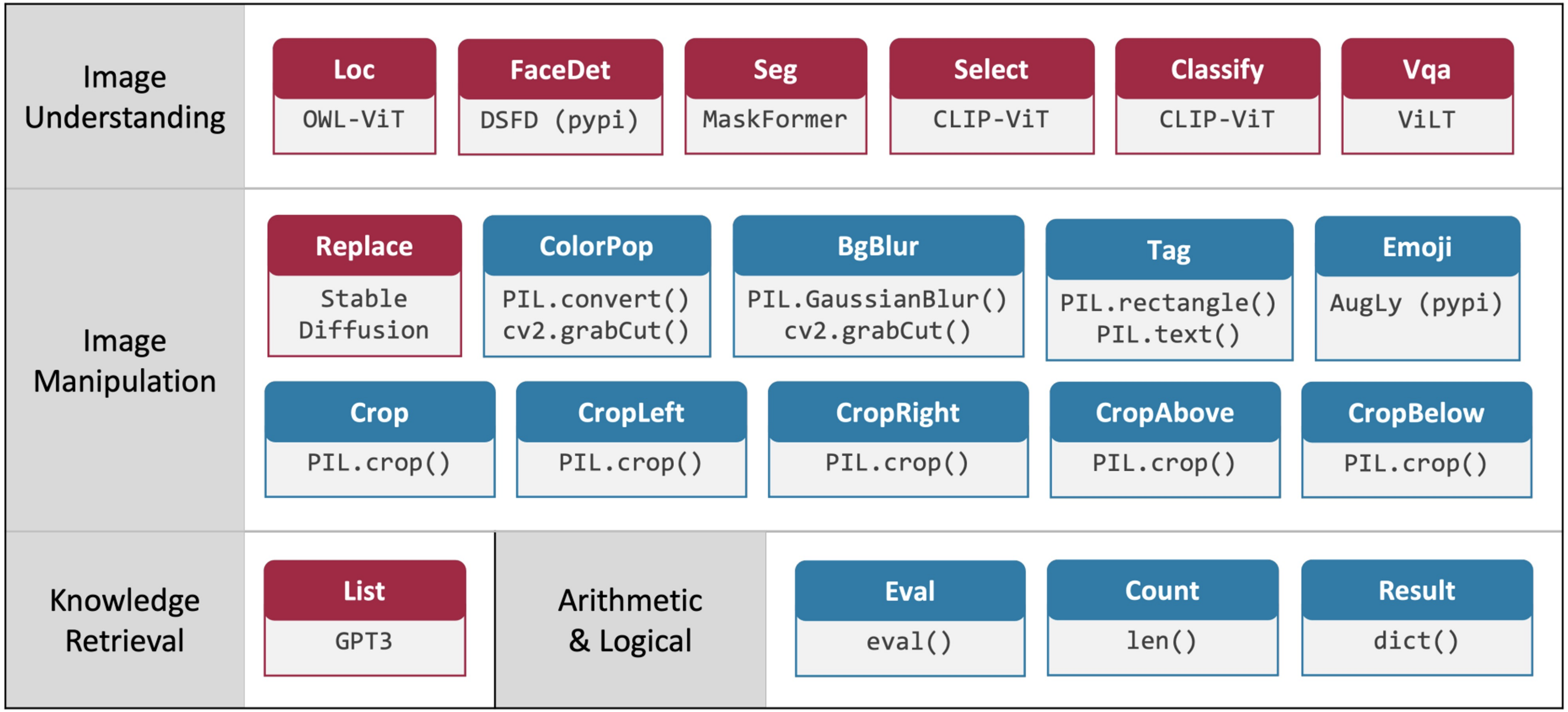


Image Understanding

Loc
OWL-ViT

FaceDet
DSFD (pypi)

Seg
MaskFormer

Select
CLIP-ViT

Classify
CLIP-ViT

Vqa
ViLT

Image Manipulation

Replace
Stable Diffusion

ColorPop
PIL.convert()
cv2.grabCut()

BgBlur
PIL.GaussianBlur()
cv2.grabCut()

Tag
PIL.rectangle()
PIL.text()

Emoji
AugLy (pypi)

Crop
PIL.crop()

CropLeft
PIL.crop()

CropRight
PIL.crop()

CropAbove
PIL.crop()

CropBelow
PIL.crop()

Knowledge Retrieval

List
GPT3

Arithmetic & Logical

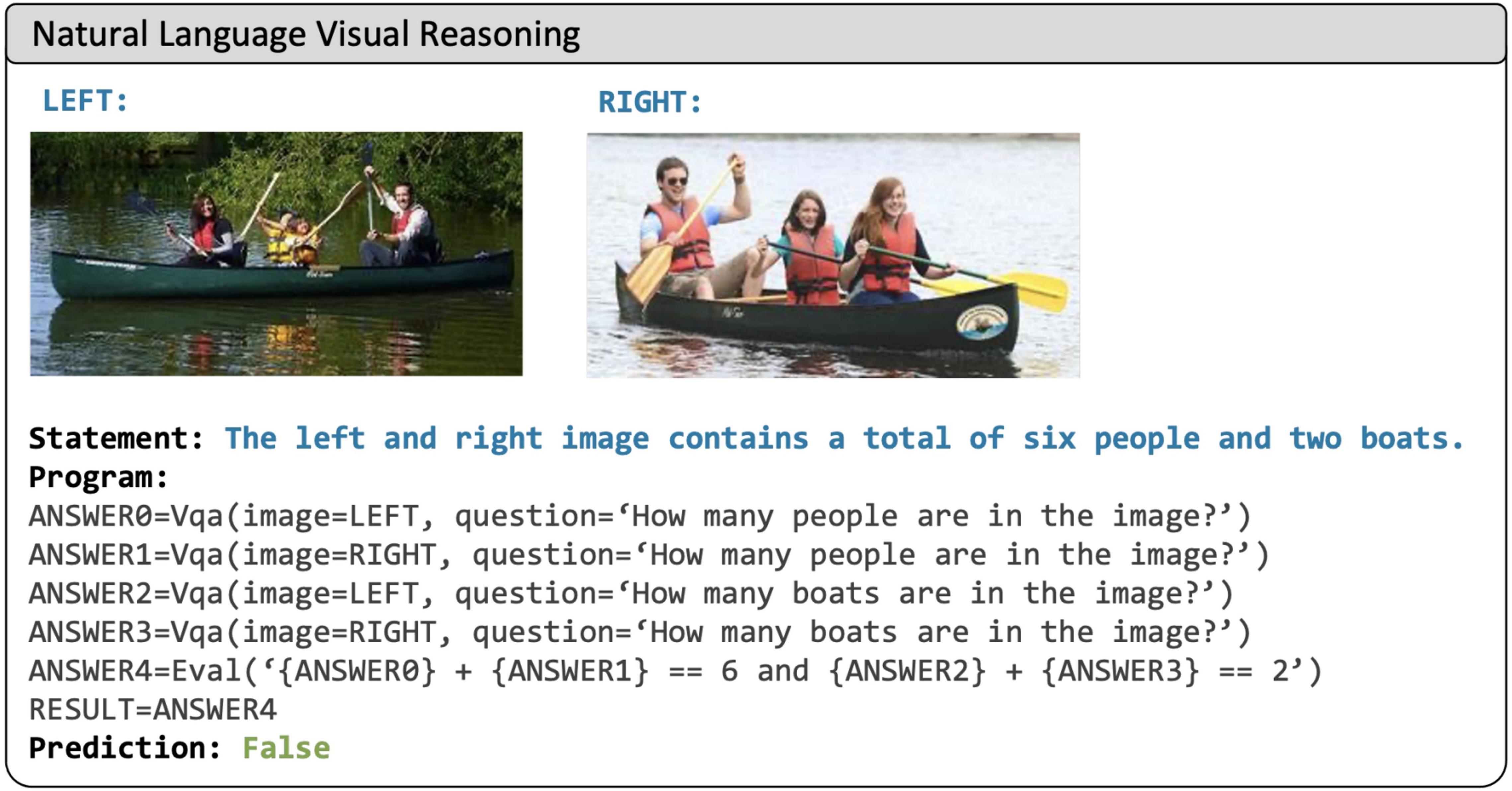
Eval
eval()

Count
len()

Result
dict()

Gupta et al "Visual Programming: Compositional visual reasoning without training". 2023.

VisProg (visual programming)



Natural Language Visual Reasoning

LEFT:

RIGHT:

Statement: The left and right image contains a total of six people and two boats.

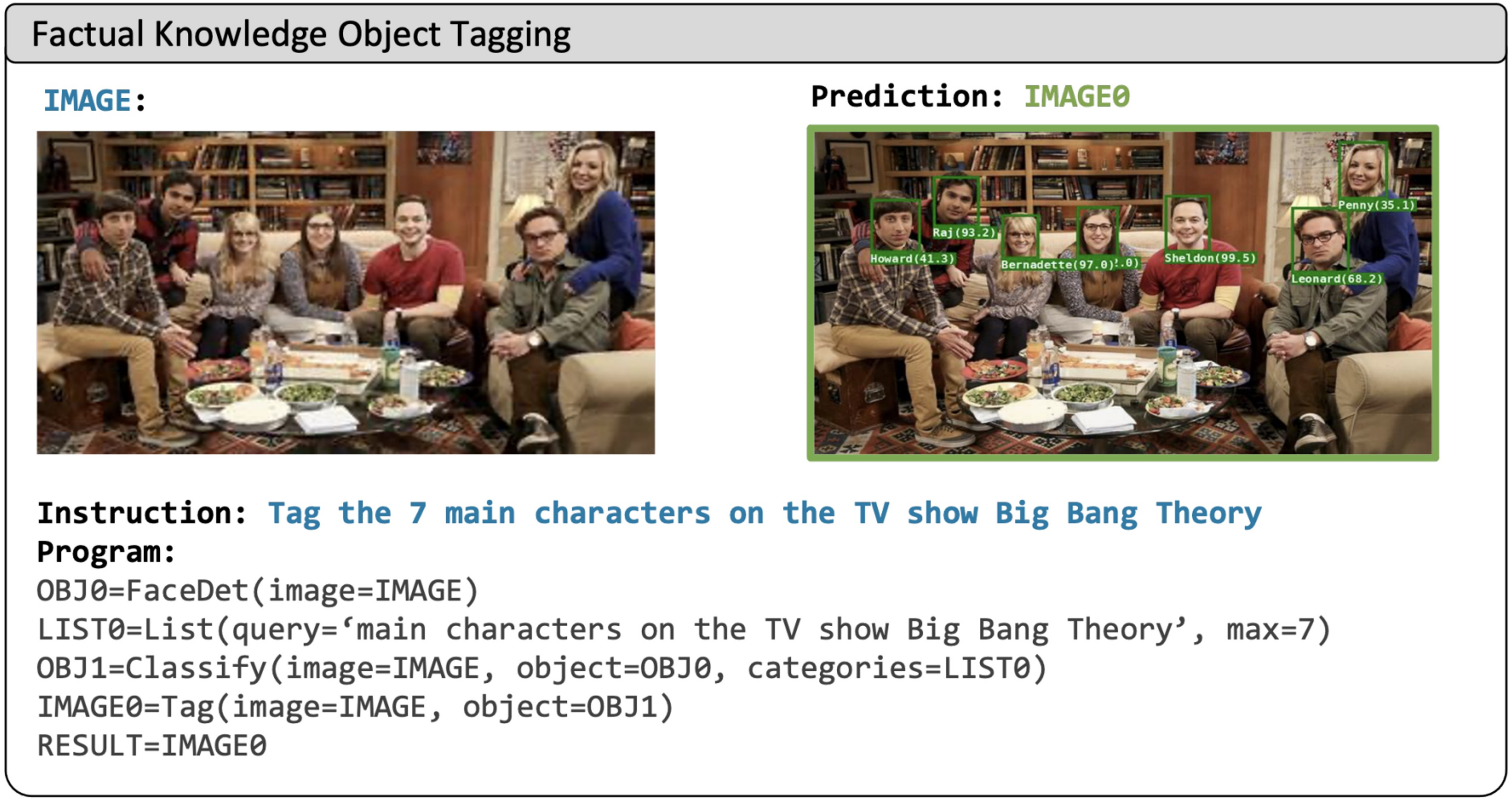
Program:

```
ANSWER0=Vqa(image=LEFT, question=‘How many people are in the image?’)
ANSWER1=Vqa(image=RIGHT, question=‘How many people are in the image?’)
ANSWER2=Vqa(image=LEFT, question=‘How many boats are in the image?’)
ANSWER3=Vqa(image=RIGHT, question=‘How many boats are in the image?’)
ANSWER4=Eval(‘{ANSWER0} + {ANSWER1} == 6 and {ANSWER2} + {ANSWER3} == 2’)
RESULT=ANSWER4
```

Prediction: False

Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)



Factual Knowledge Object Tagging

IMAGE:

Prediction: IMAGE0

Instruction: Tag the 7 main characters on the TV show Big Bang Theory

Program:

```
OBJ0=FaceDet(image=IMAGE)
LIST0=List(query='main characters on the TV show Big Bang Theory', max=7)
OBJ1=Classify(image=IMAGE, object=OBJ0, categories=LIST0)
IMAGE0=Tag(image=IMAGE, object=OBJ1)
RESULT=IMAGE0
```

Gupta et al "Visual Programming: Compositional visual reasoning without training". 2023.

VisProg (visual programming)

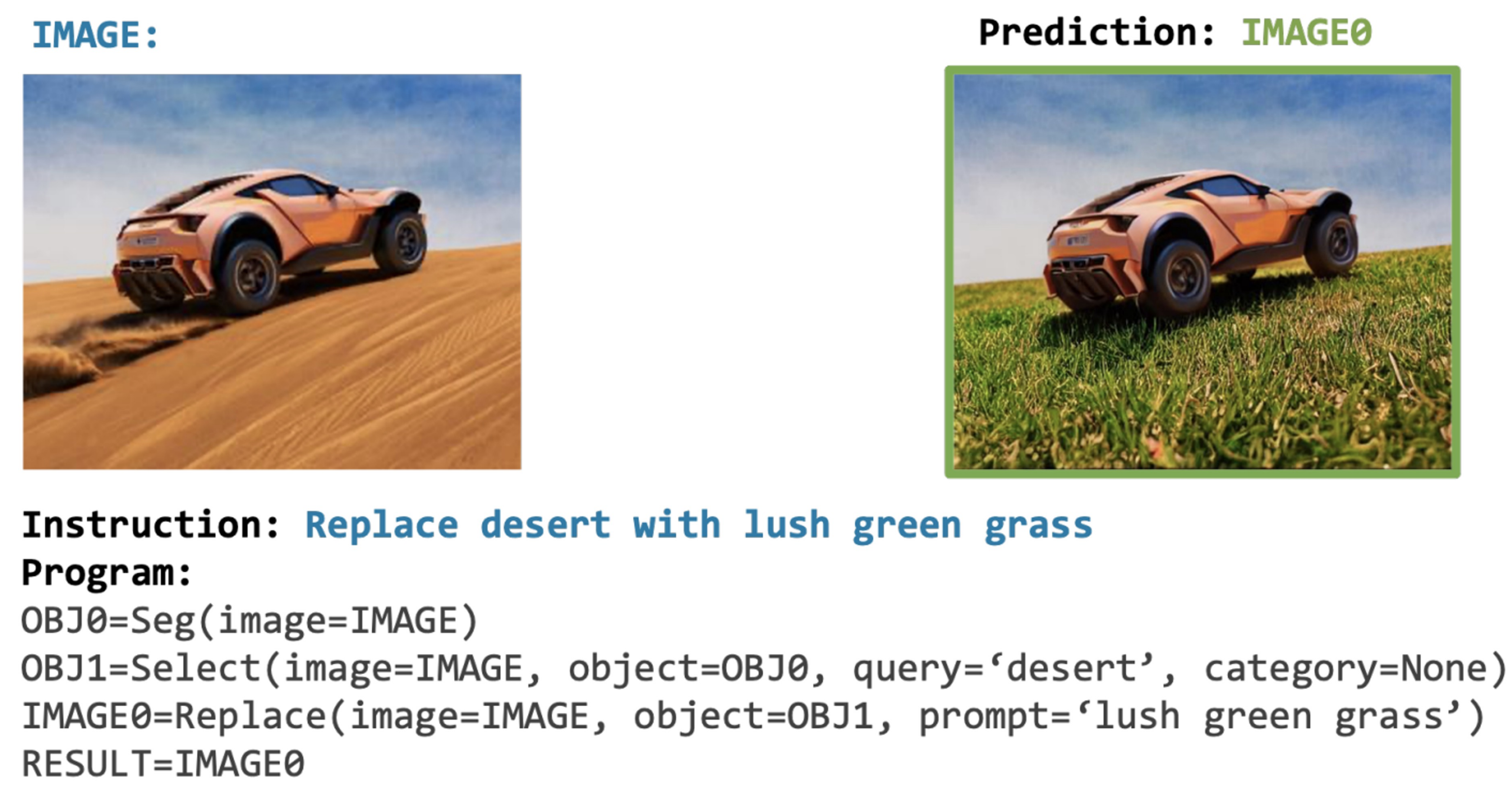


IMAGE:

Prediction: IMAGE0

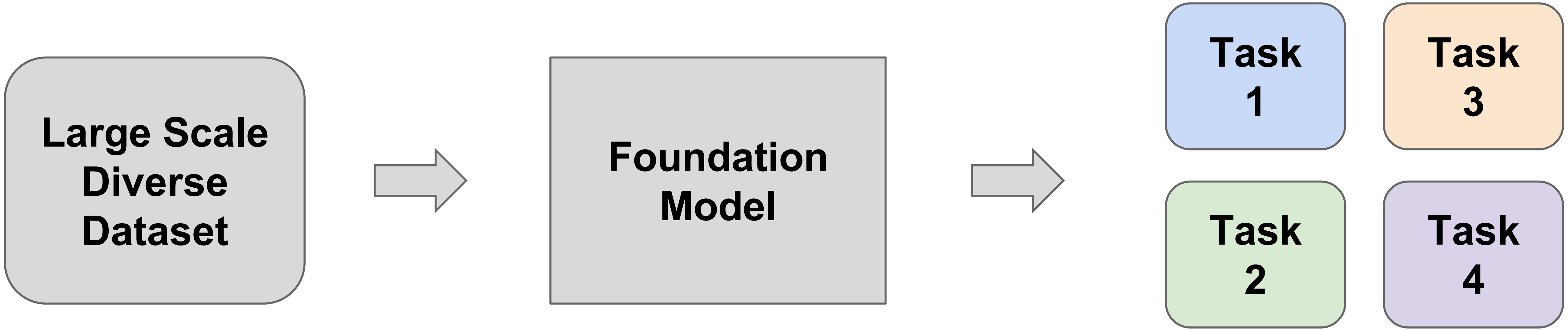
Instruction: Replace desert with lush green grass

Program:

```
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='desert', category=None)
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt='lush green grass')
RESULT=IMAGE0
```

Gupta et al "Visual Programming: Compositional visual reasoning without training". 2023.

Summary



Large Scale Diverse Dataset

Foundation Model

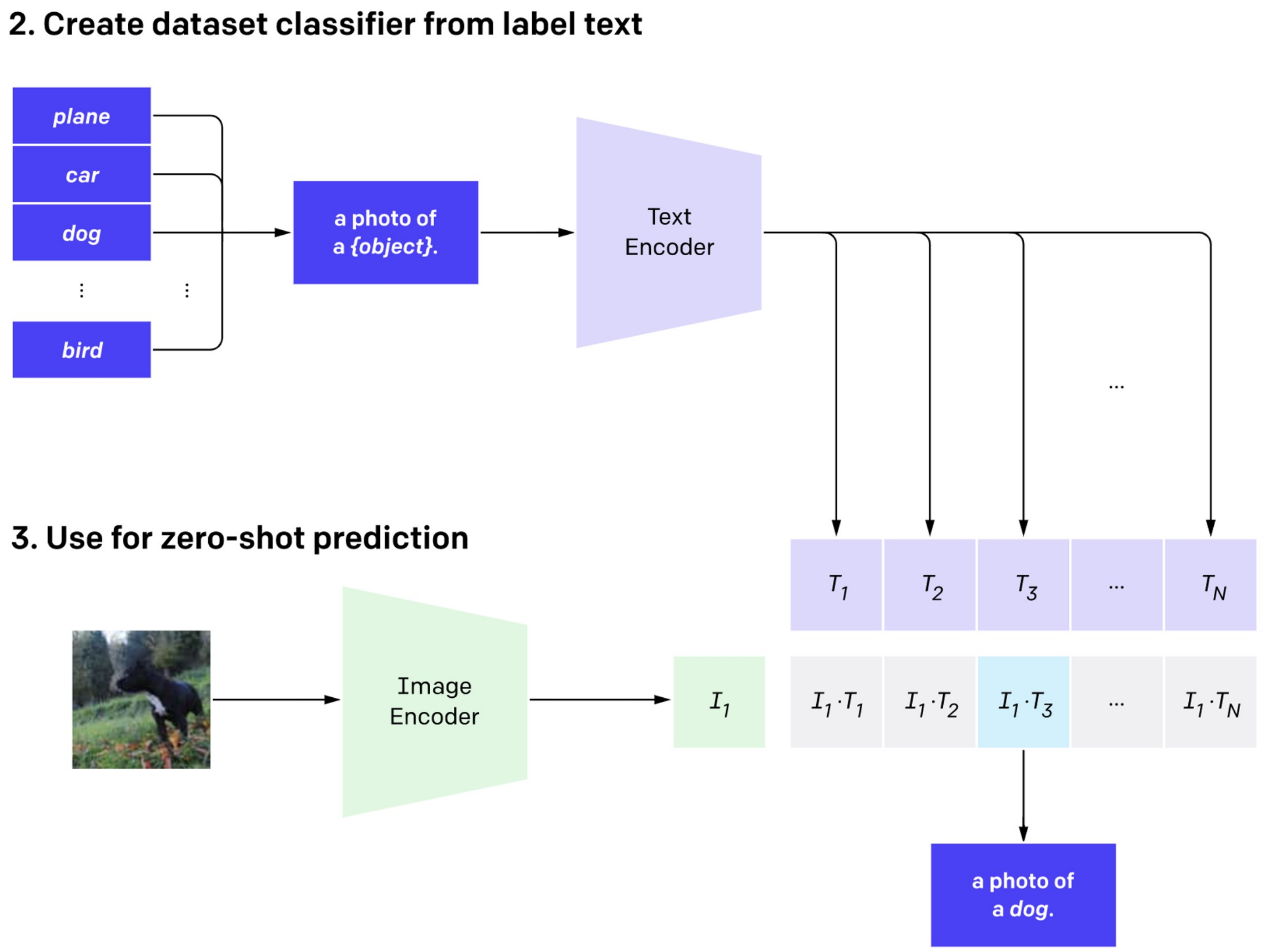
Task 1

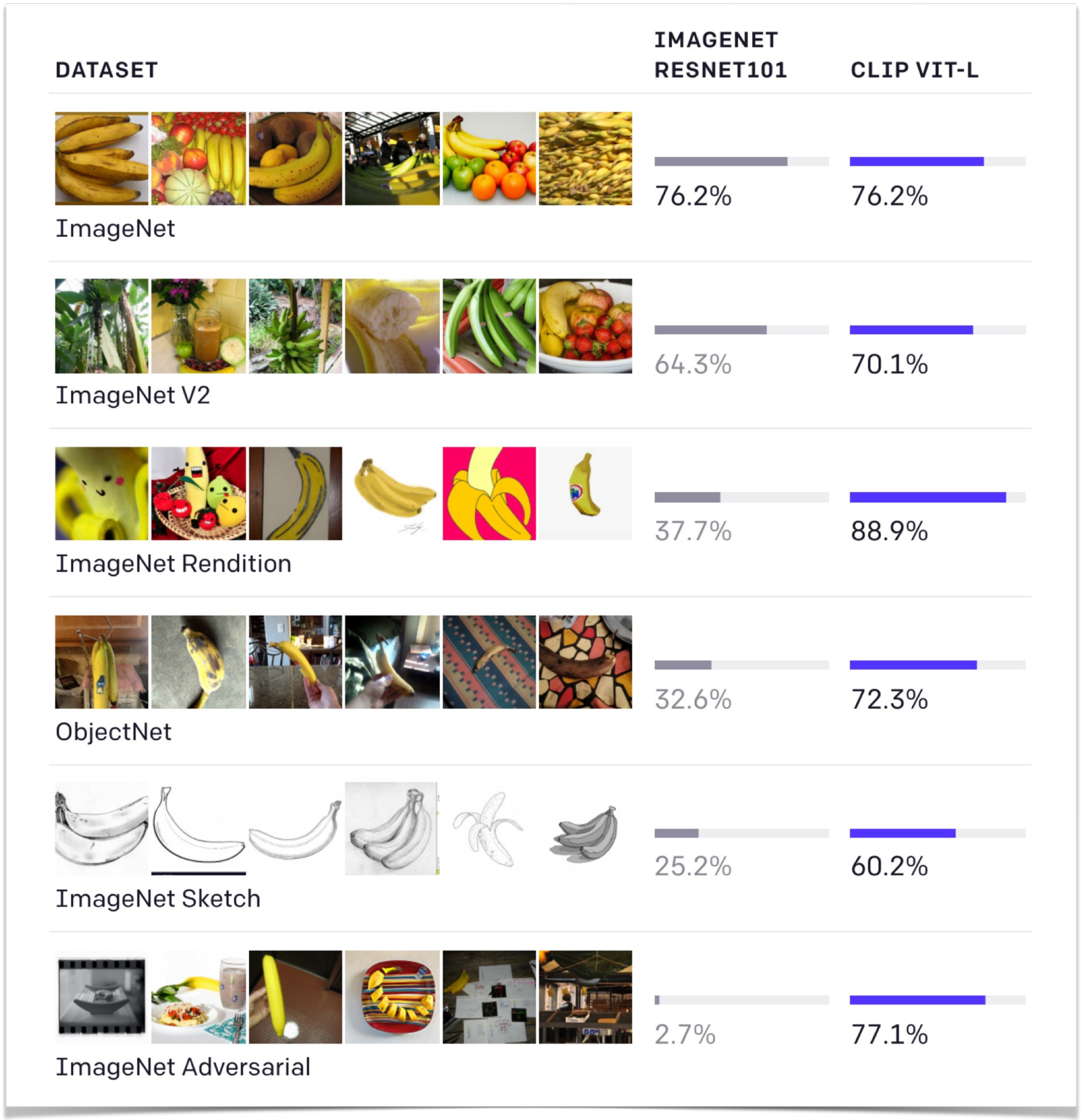
Task 3

Task 2

Task 4

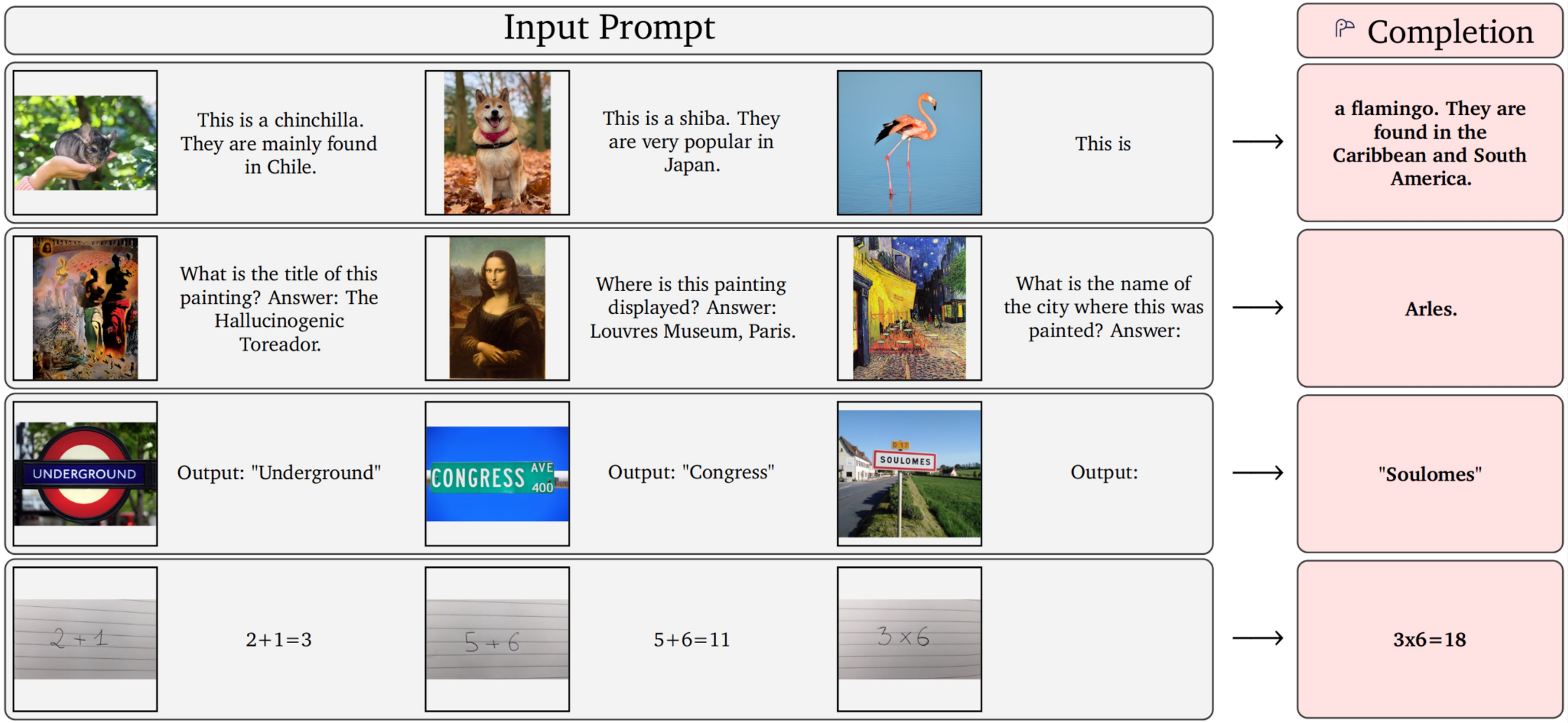
Summary





DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%
ImageNet V2	64.3%	70.1%
ImageNet Rendition	37.7%	88.9%
ObjectNet	32.6%	72.3%
ImageNet Sketch	25.2%	60.2%
ImageNet Adversarial	2.7%	77.1%

Summary



Input Prompt

Completion

This is a chinchilla. They are mainly found in Chile.

This is a shiba. They are very popular in Japan.

This is → a flamingo. They are found in the Caribbean and South America.

What is the title of this painting? Answer: The Hallucinogenic Toreador.

Where is this painting displayed? Answer: Louvres Museum, Paris.

What is the name of the city where this was painted? Answer: → Arles.

Output: "Underground"

Output: "Congress"

Output: → "Soulomes"

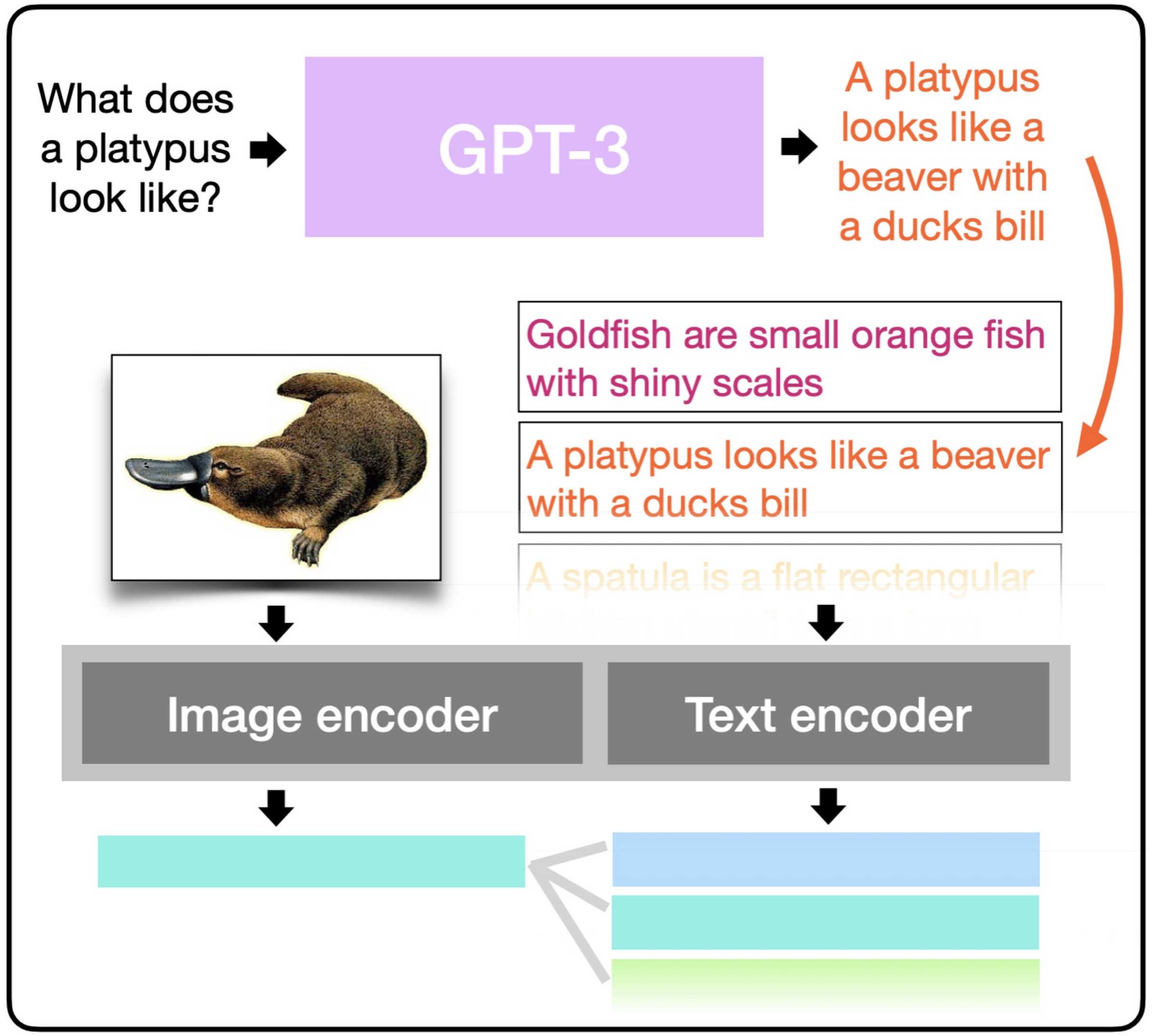
2+1=3

5+6=11

→ 3x6=18

Summary

Summary



What does a platypus look like?

GPT-3

A platypus looks like a beaver with a ducks bill

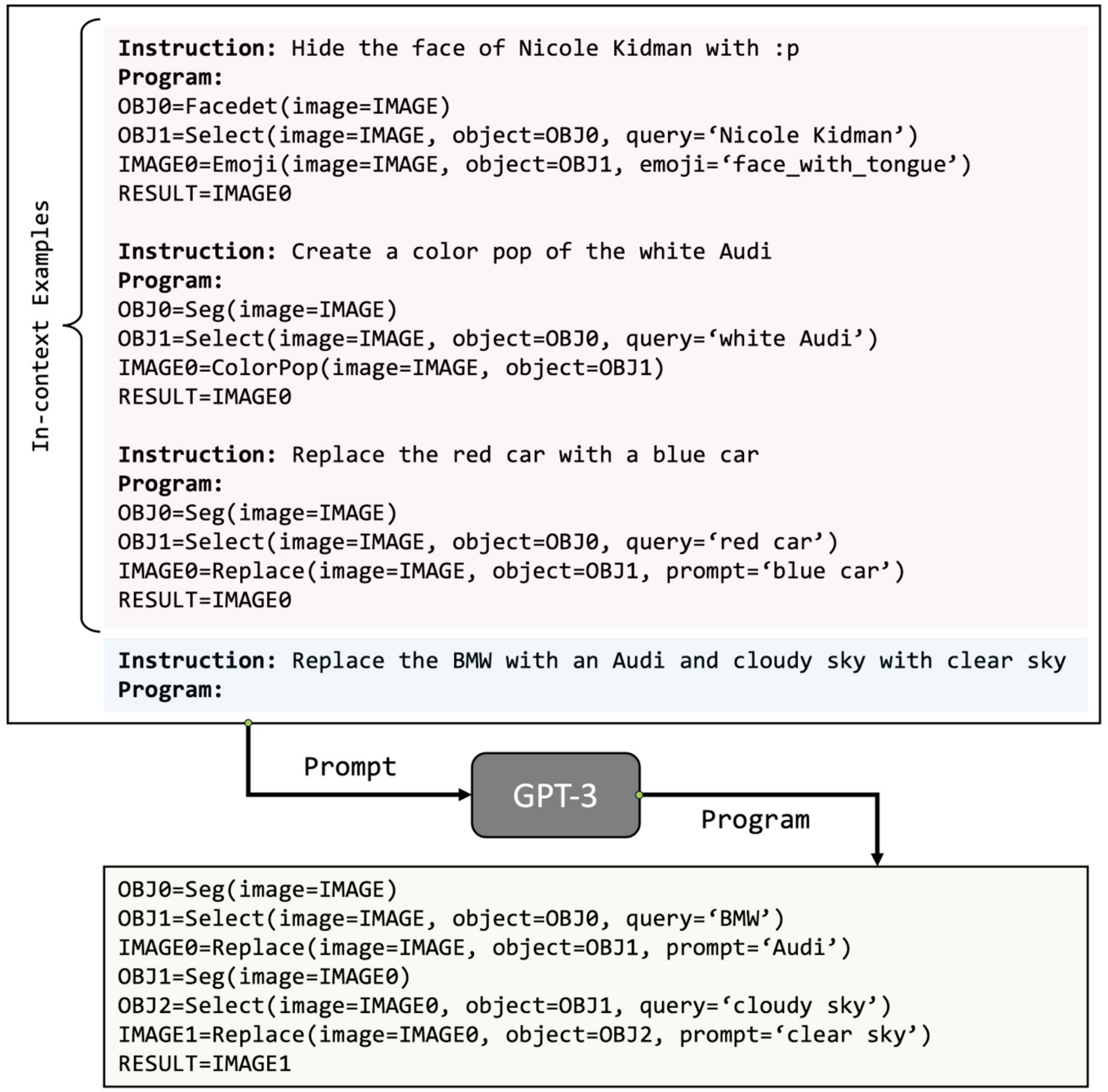
Goldfish are small orange fish with shiny scales

A platypus looks like a beaver with a ducks bill

A spatula is a flat rectangular

Image encoder

Text encoder



In-context Examples

```
Instruction: Hide the face of Nicole Kidman with :p
Program:
OBJ0=Facedet(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='Nicole Kidman')
IMAGE0=Emoji(image=IMAGE, object=OBJ1, emoji='face_with_tongue')
RESULT=IMAGE0

Instruction: Create a color pop of the white Audi
Program:
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='white Audi')
IMAGE0=ColorPop(image=IMAGE, object=OBJ1)
RESULT=IMAGE0

Instruction: Replace the red car with a blue car
Program:
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='red car')
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt='blue car')
RESULT=IMAGE0

Instruction: Replace the BMW with an Audi and cloudy sky with clear sky
Program:
```

Prompt

GPT-3

Program

```
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='BMW')
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt='Audi')
OBJ1=Seg(image=IMAGE0)
OBJ2=Select(image=IMAGE0, object=OBJ1, query='cloudy sky')
IMAGE1=Replace(image=IMAGE0, object=OBJ2, prompt='clear sky')
RESULT=IMAGE1
```

Next time: Robot Learning