Illustration by Google Deepmind

COMP541
DEEP LEARNING

Lecture #13 – Multimodal Models

KOÇ UNIVERSITY

Aykut Erdem // Koç University // Fall 2025

Previously on COMP541

- fine-tuning and fine-tuning methods
- instruction tuning
- learning from human feedback
- reasoning models

Illustration by Google Deepmind

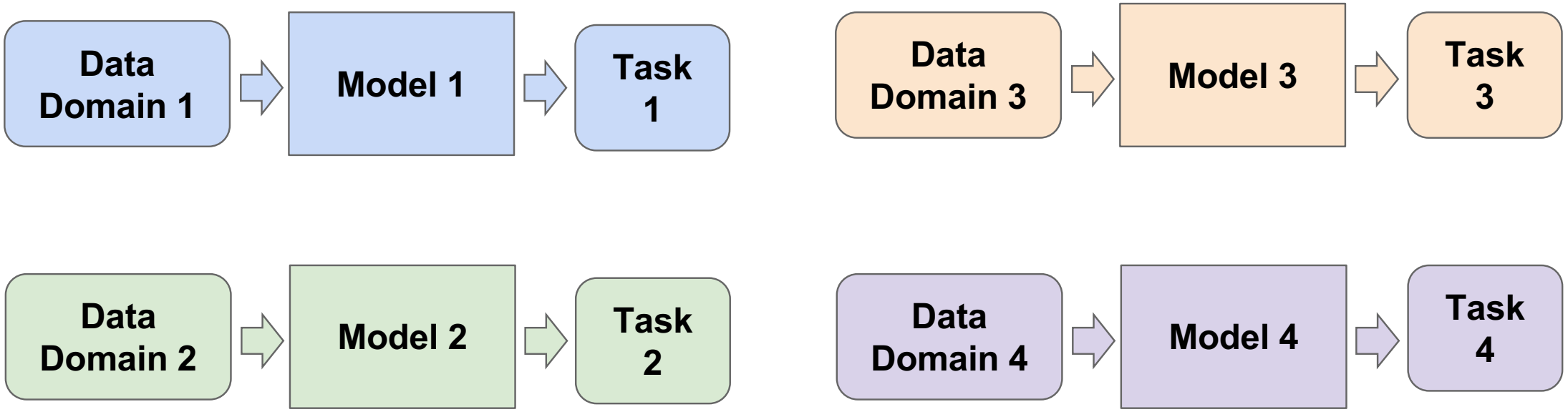
Lecture overview

- multimodal foundation models
- classification: CLIP, CoCa
- language modeling and vision: LLaVa, Flamingo, Molmo
- segment anything
- chaining models

Disclaimer: Much of the material and slides for this lecture were borrowed from —Ranjay Krishna's Stanford 231n lecture on Multimodal Foundation Models

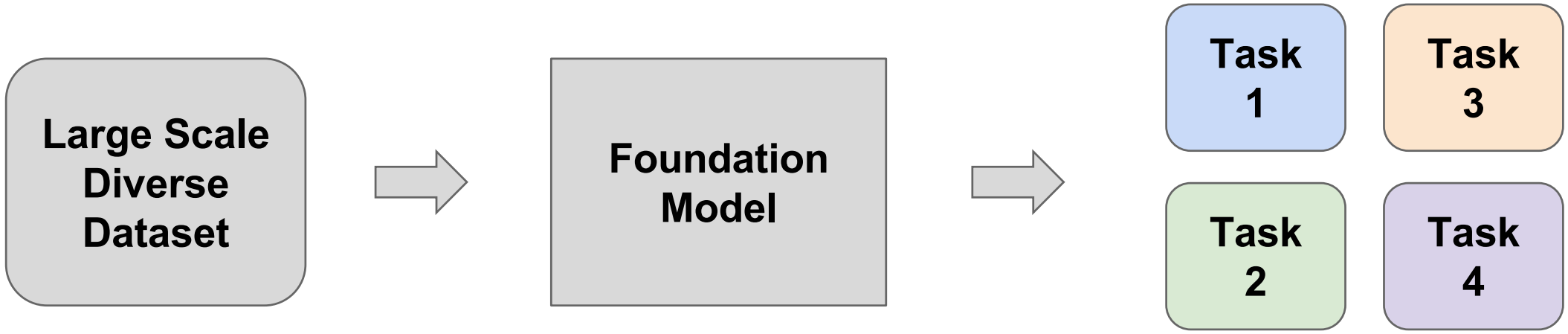
How have we been thinking about models in this class so far?

- Train a specialized model for each task



Now, we build **Foundation Models**

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



Now, we build Foundation Models

- Pre-train one model or many different tasks

Large Scale Diverse Dataset

Task 1 **Task 3** **Task 2** **Task 4**

Pre-tra shot / few-shot

On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh
Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman
Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt
Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain
Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani
Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi
Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent
Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning
Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan
Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan
Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech
Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren
Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh
Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin
Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu
Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia
Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou
Percy Liang*[1]

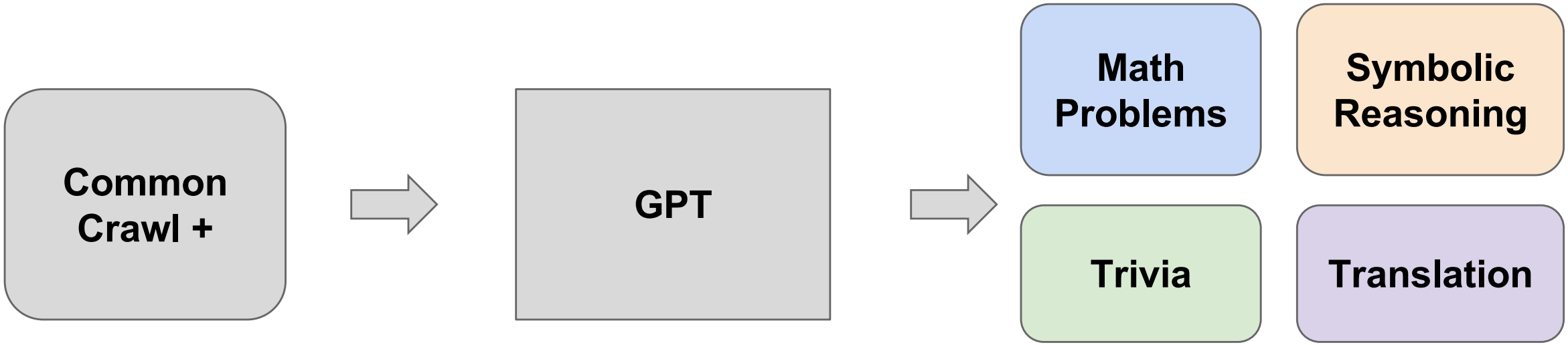
Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) trained on broad data (generally using self-supervision at scale) that can be adapted to a wide range of downstream tasks. We call these models foundation models *to underscore their critically central yet incomplete character. This report provides a thorough account of the opportunities and risks of foundation models, ranging from their capabilities (e.g., language, vision, robotic manipulation, reasoning, human interaction) and technical principles (e.g., model architectures, training procedures, data, systems, security, evaluation, theory) to their applications (e.g., law, healthcare, education) and societal impact (e.g., inequity, misuse, economic and environmental impact, legal and ethical considerations). Though foundation models are based on standard deep learning and transfer learning, their scale results in new emergent capabilities, and their effectiveness across so many tasks incentivizes homogenization. Homogenization provides powerful leverage but demands caution, as the defects of the foundation model are inherited by all the adapted models downstream. Despite the impending widespread deployment of foundation models, we currently lack a clear understanding of how they work, when they fail, and what they are even capable of due to their emergent properties. To tackle these questions, we believe much of the critical research on foundation models will require deep interdisciplinary collaboration commensurate with their fundamentally sociotechnical nature.*

[1]Corresponding author: pliang@cs.stanford.edu *Equal contribution.

Foundation Models

Language



Pre-training Fine-tuning / zero-shot / few-shot

Foundation Models

There are many classes of Foundation Models

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

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

Foundation Models

There are many classes of Foundation Models

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

Covered before

Foundation Models

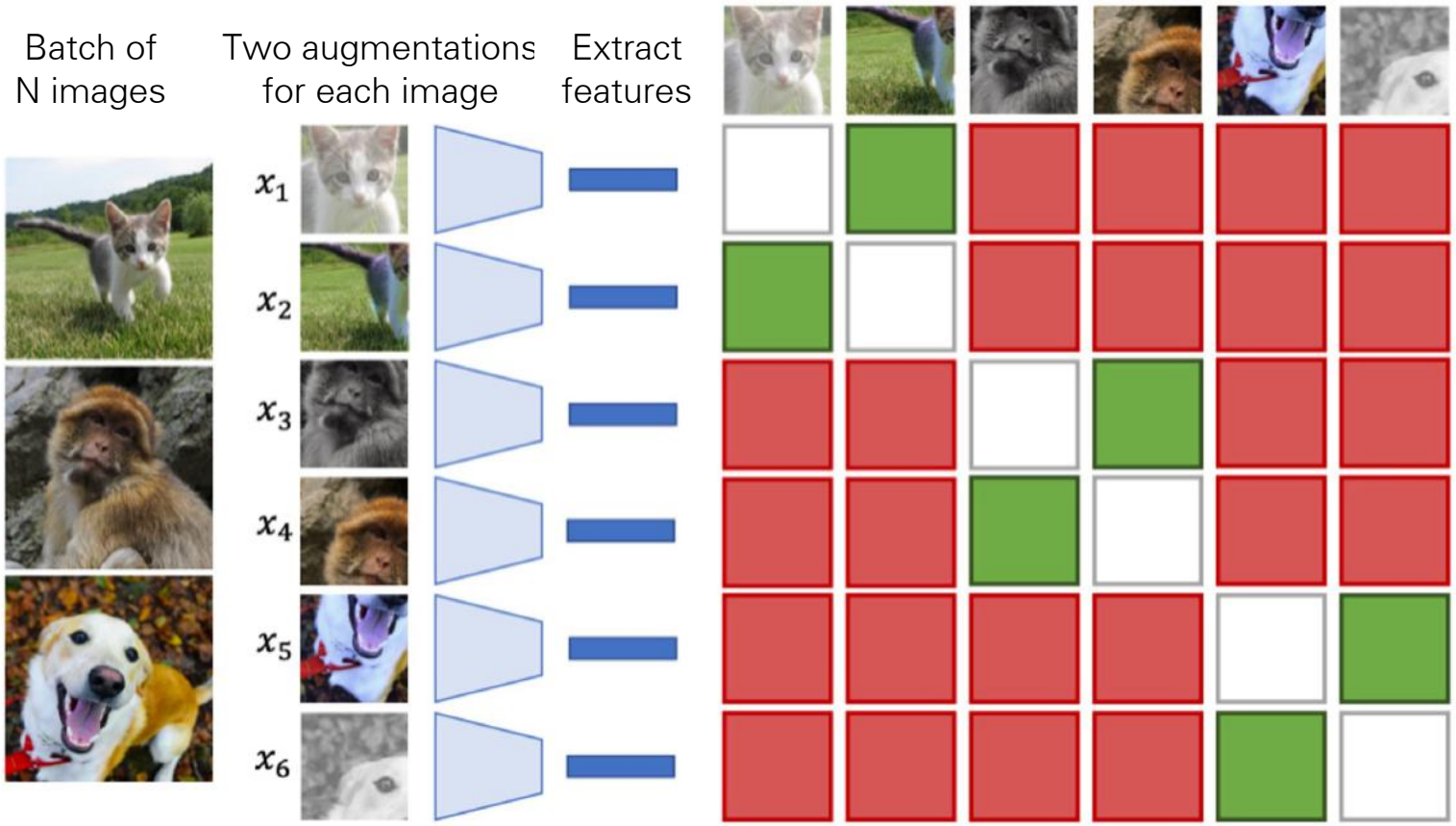
There are many classes of Foundation Models

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	

Will cover now!

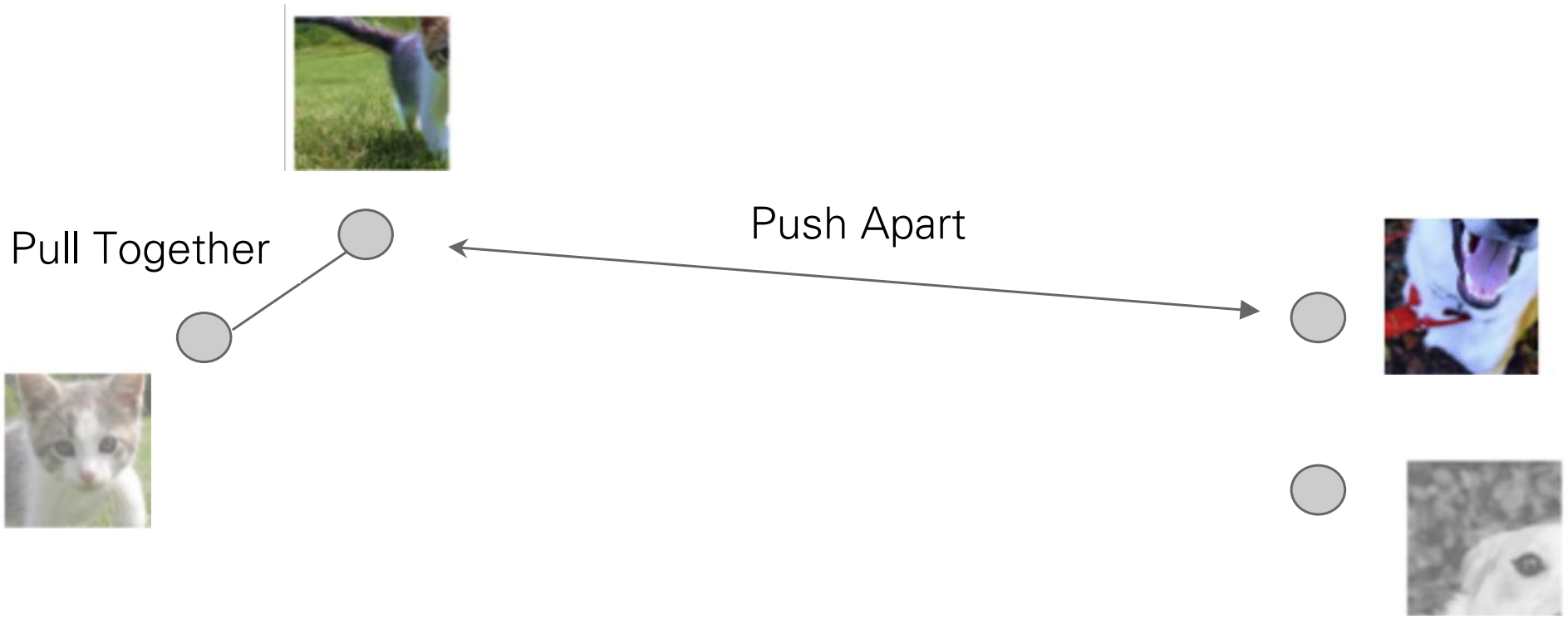
Foundation Models for Classification

Self-supervised objective from SimCLR

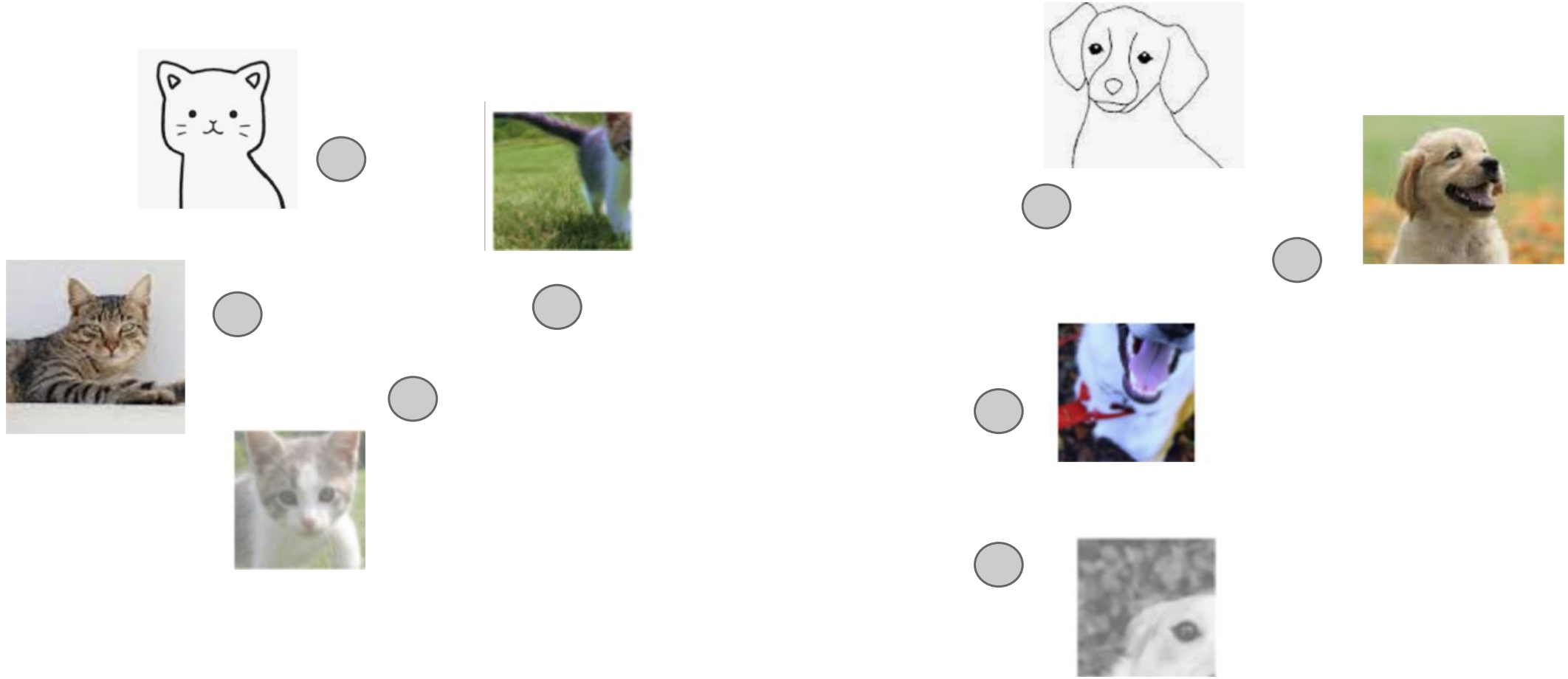


- Use Self Supervised learning to learn good image features
- Can train small classifiers on top of these features using supervised learning

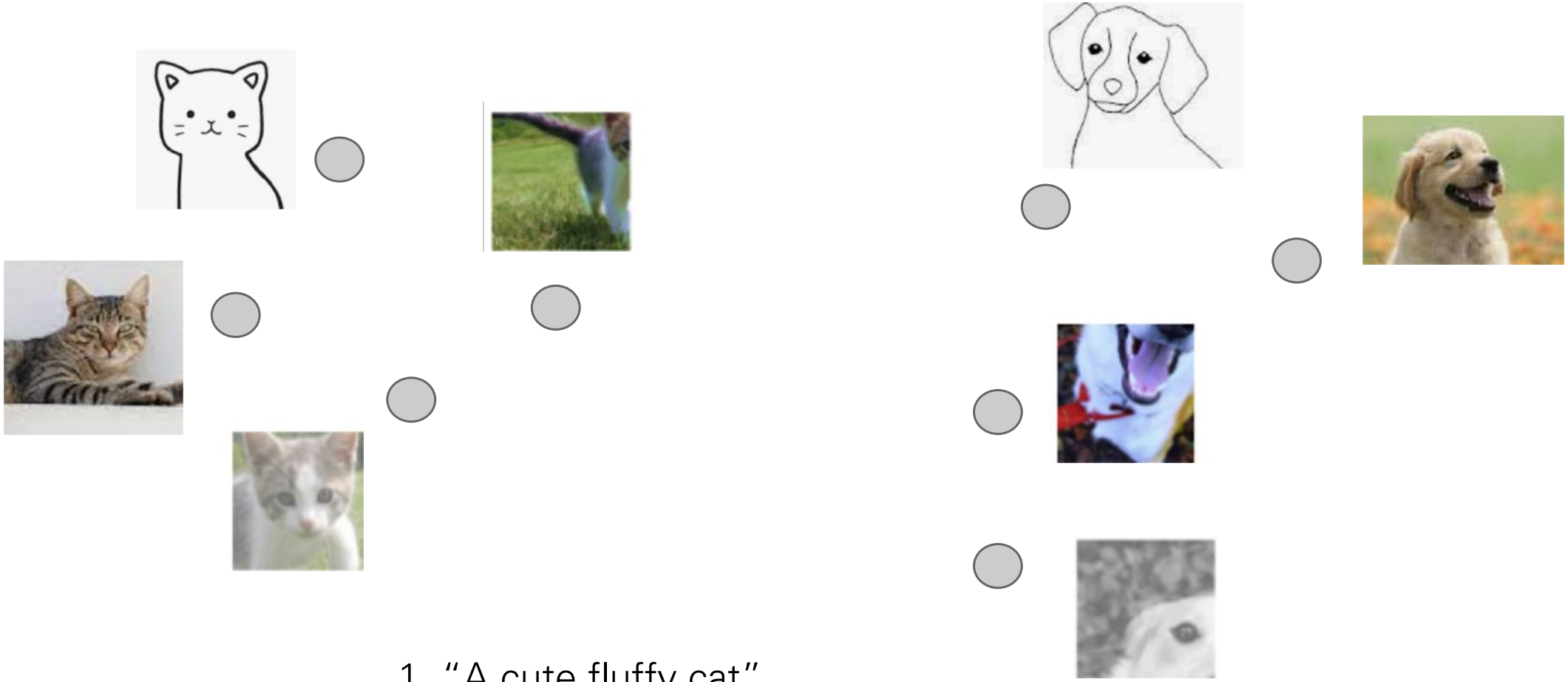
The main idea was to learning concepts without labels → a self-supervised pretraining objective



The hope is that the learned representations generalize to new instances



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



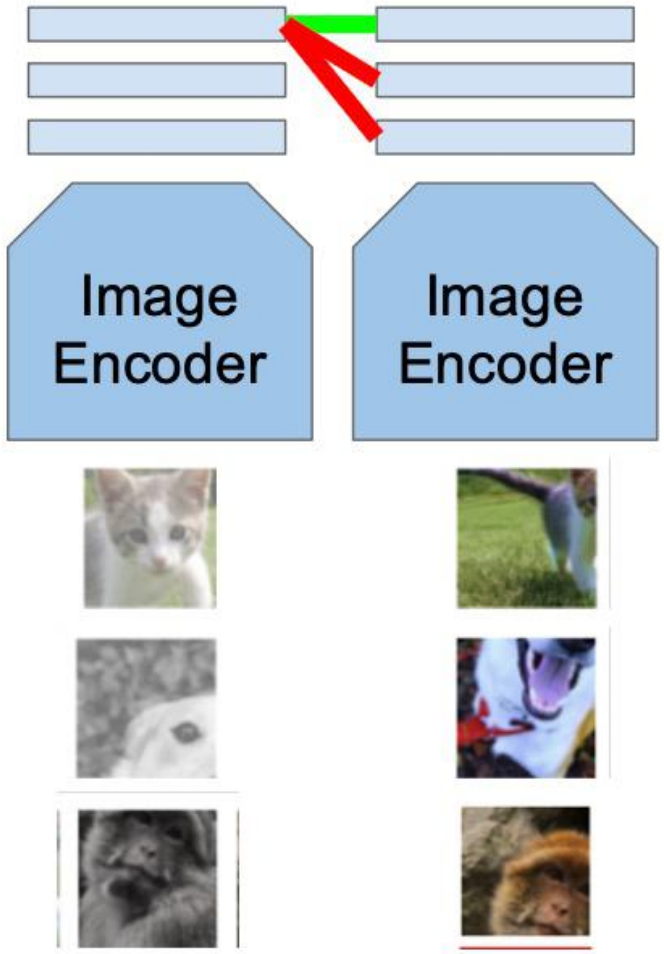
1. “A cute fluffy cat”
2. “My favorite dog is a golden retriever”

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

"My favorite dog is a golden retriever"

"A cute fluffy cat"

SimCLR



SimCLR

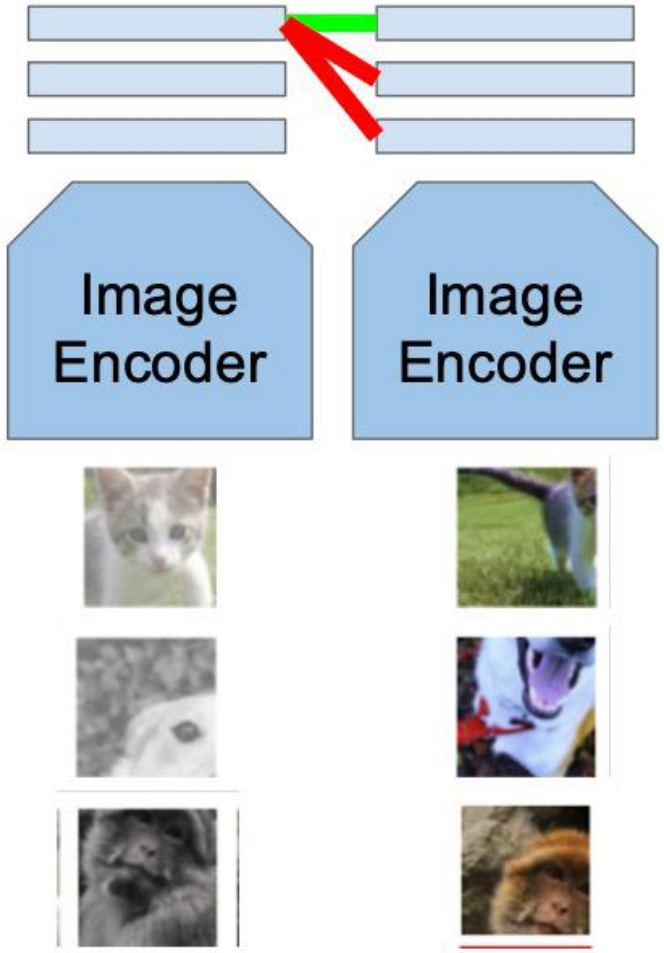


Image Encoder
Image Encoder

CLIP

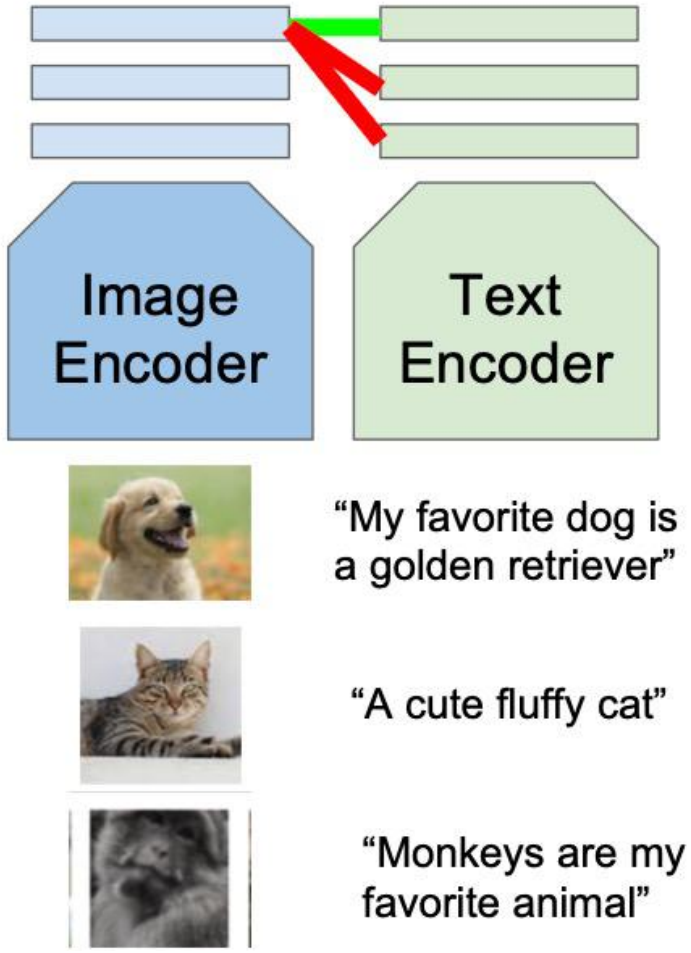
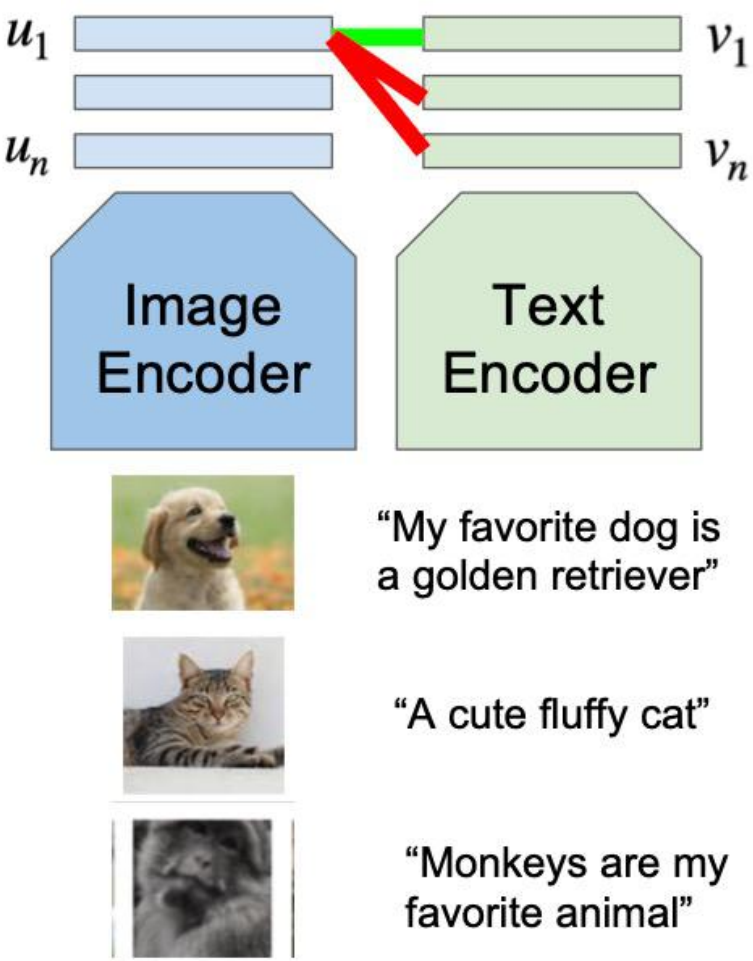


Image Encoder
Text Encoder
"My favorite dog is a golden retriever"
"A cute fluffy cat"
"Monkeys are my favorite animal"

CLIP is trained with the same contrastive objective

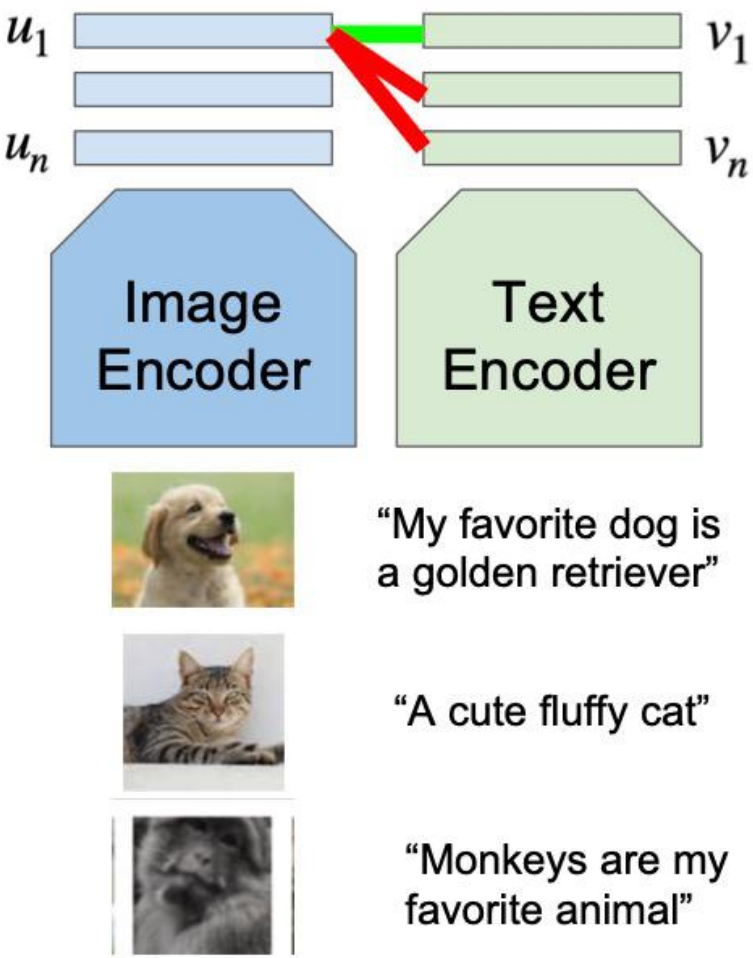
$$\sum_{i=1}^{n} -\log\left(\frac{e^{\langle u_i, v_i\rangle}}{\sum_{j=1}^{n} e^{\langle u_i, v_j\rangle}}\right)$$



CLIP is trained with the same contrastive objective

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

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



Lots of image-text data can be found online

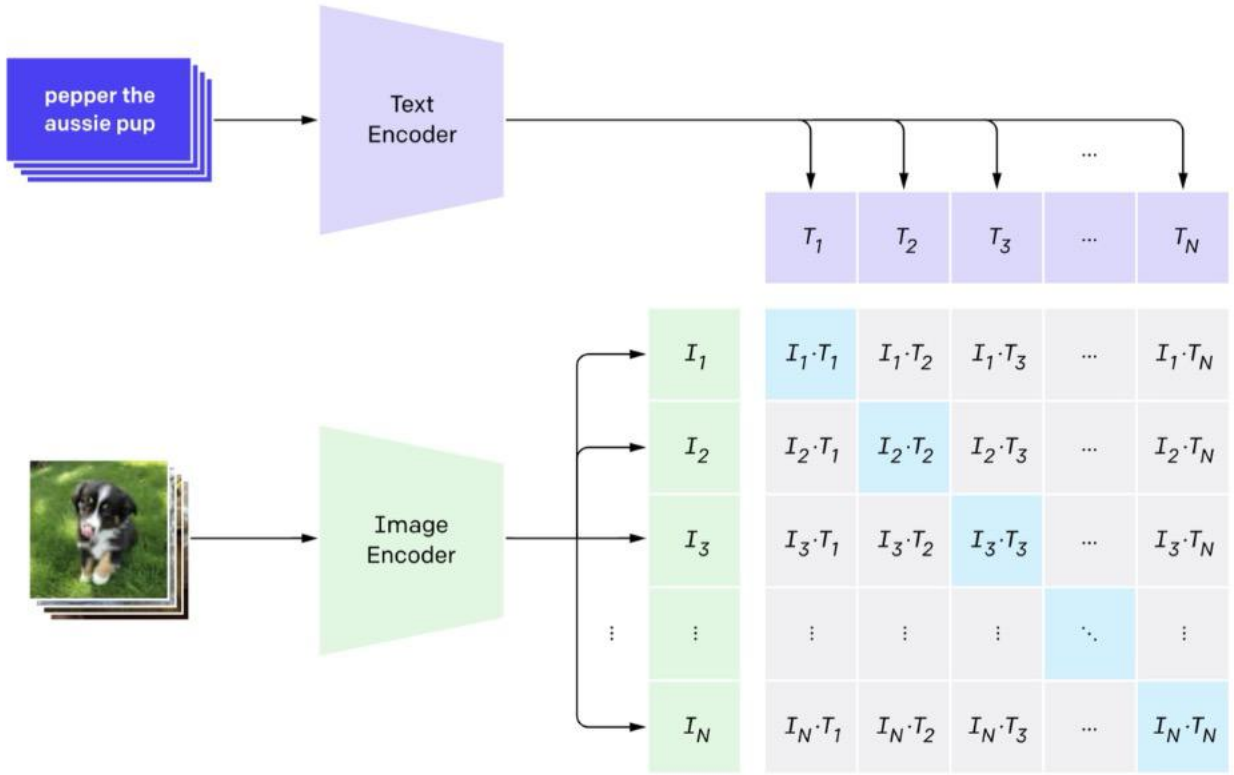
Mount Rainier's northwestern slope viewed aerially just before sunset on September 6, 2020

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

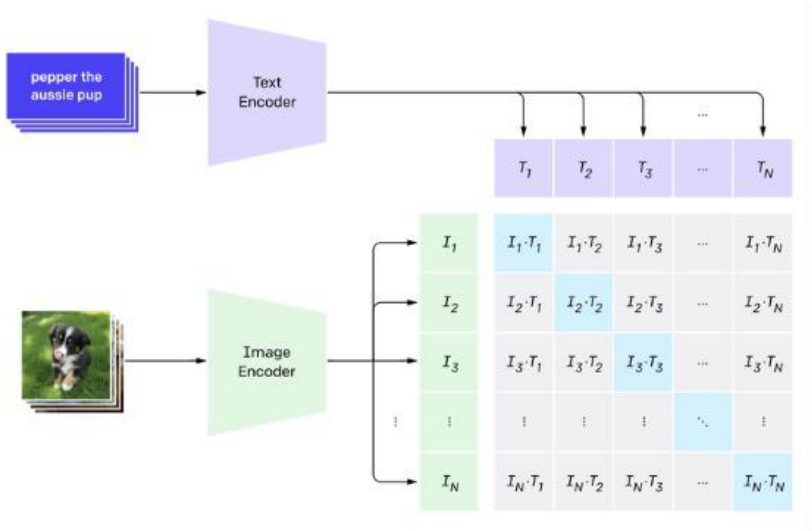
1. Contrastive pre-training



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

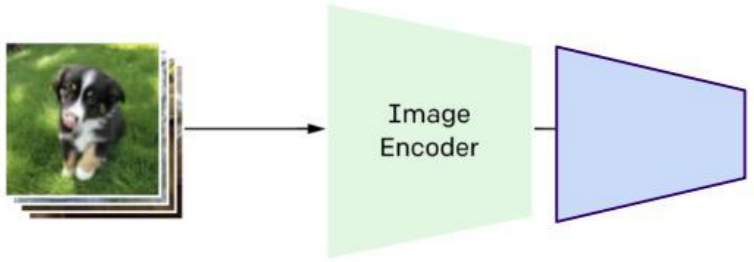
Using pre-trained models out of the bo

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



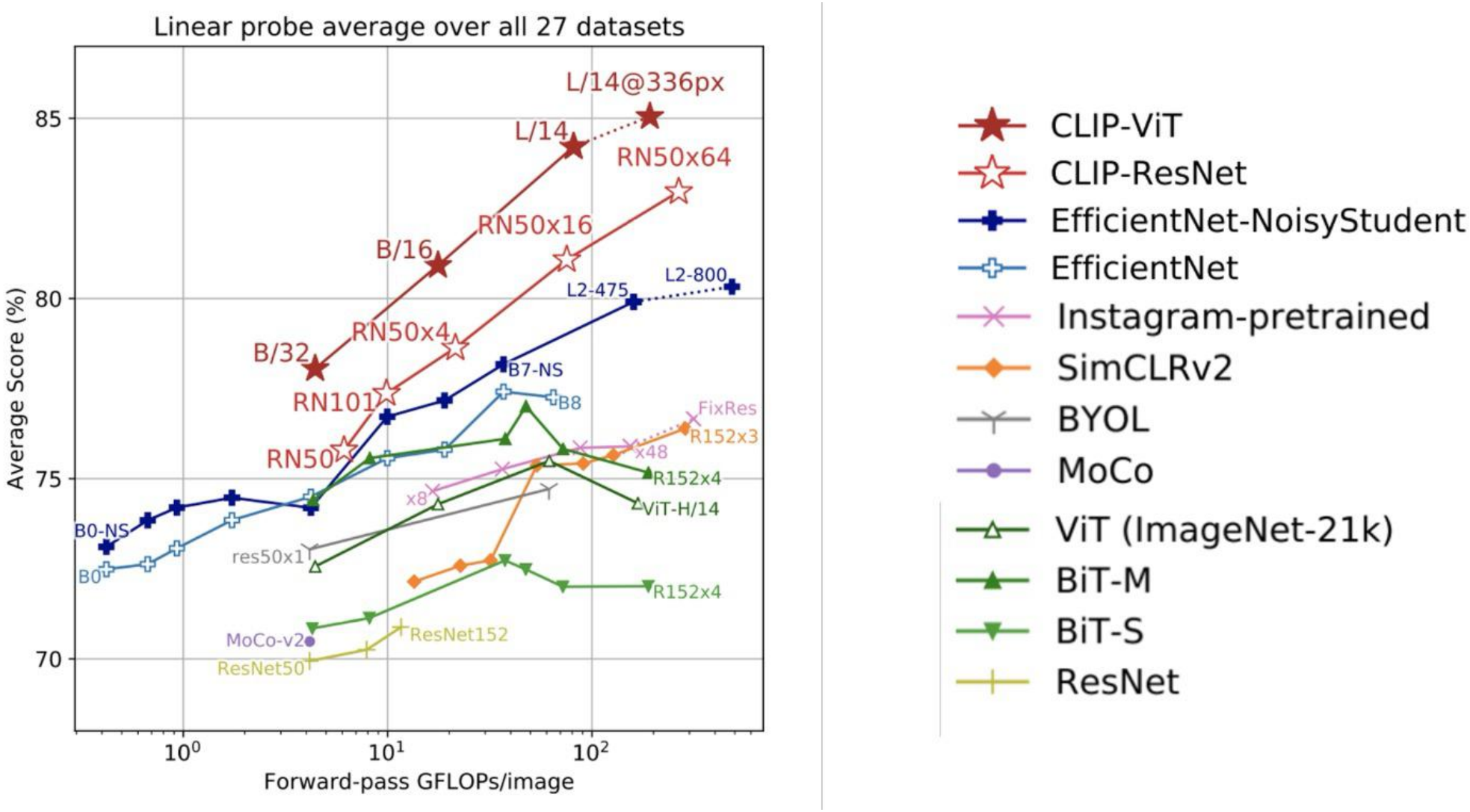
Pre-training tasks:
Contrastive Objective

Step 2: Transfer encoder to downstream tasks via linear classifiers



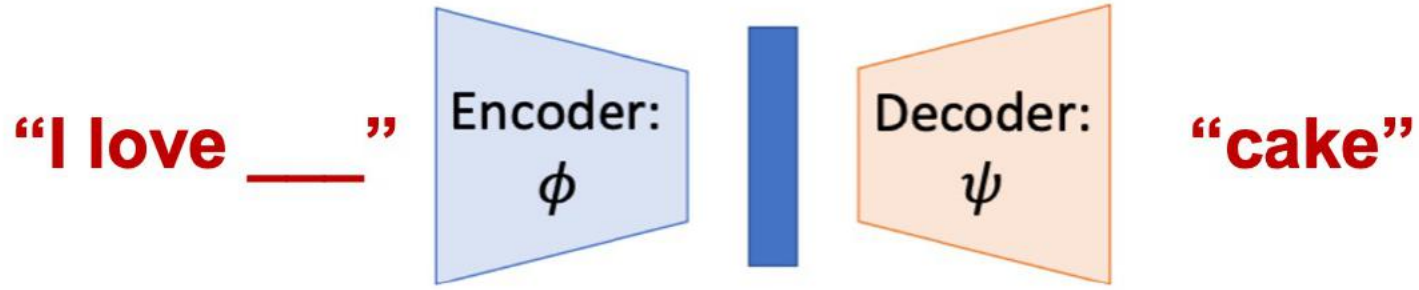
Downstream tasks:
Image classification, object detection, semantic segmentation

CLIP features w/ **linear probe** across multiple datasets

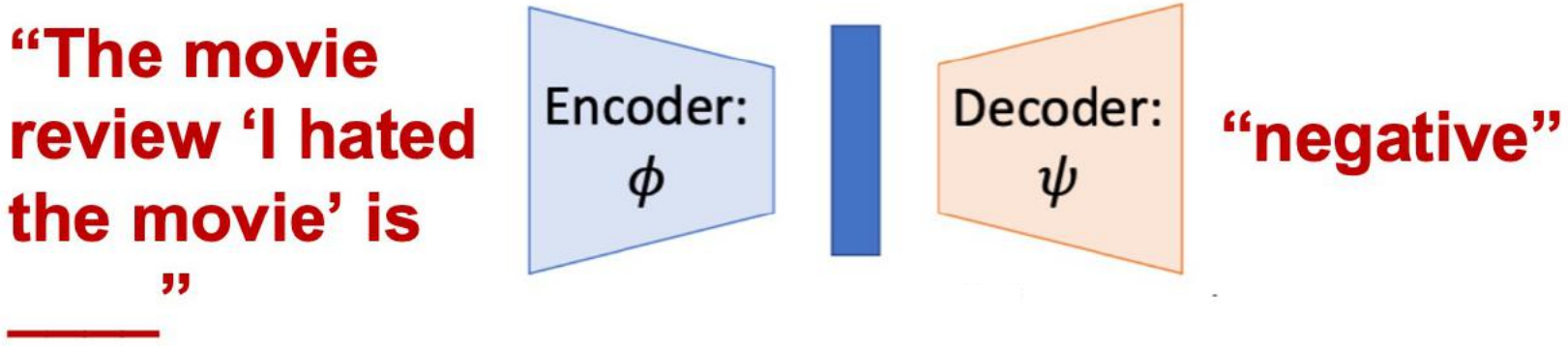


Big difference with language models: We can use LLMs **zero-shot** for new downstream tasks

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

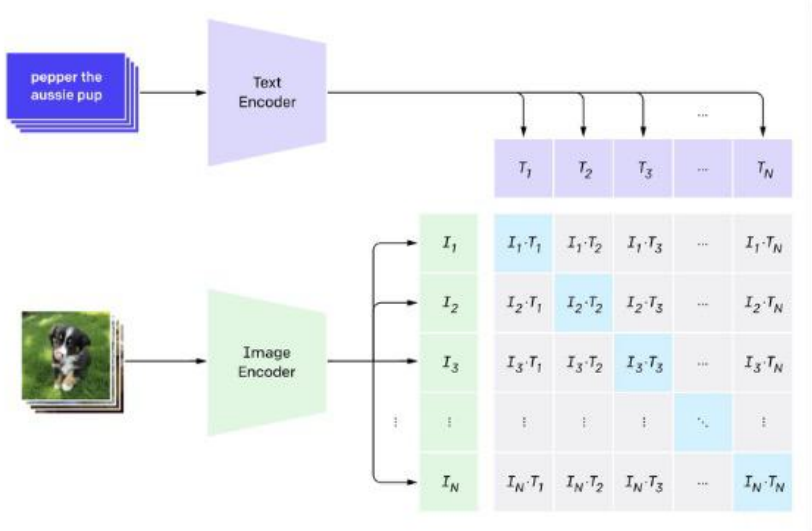


Step 2: Transfer encoder to downstream tasks via linear classifiers



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

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

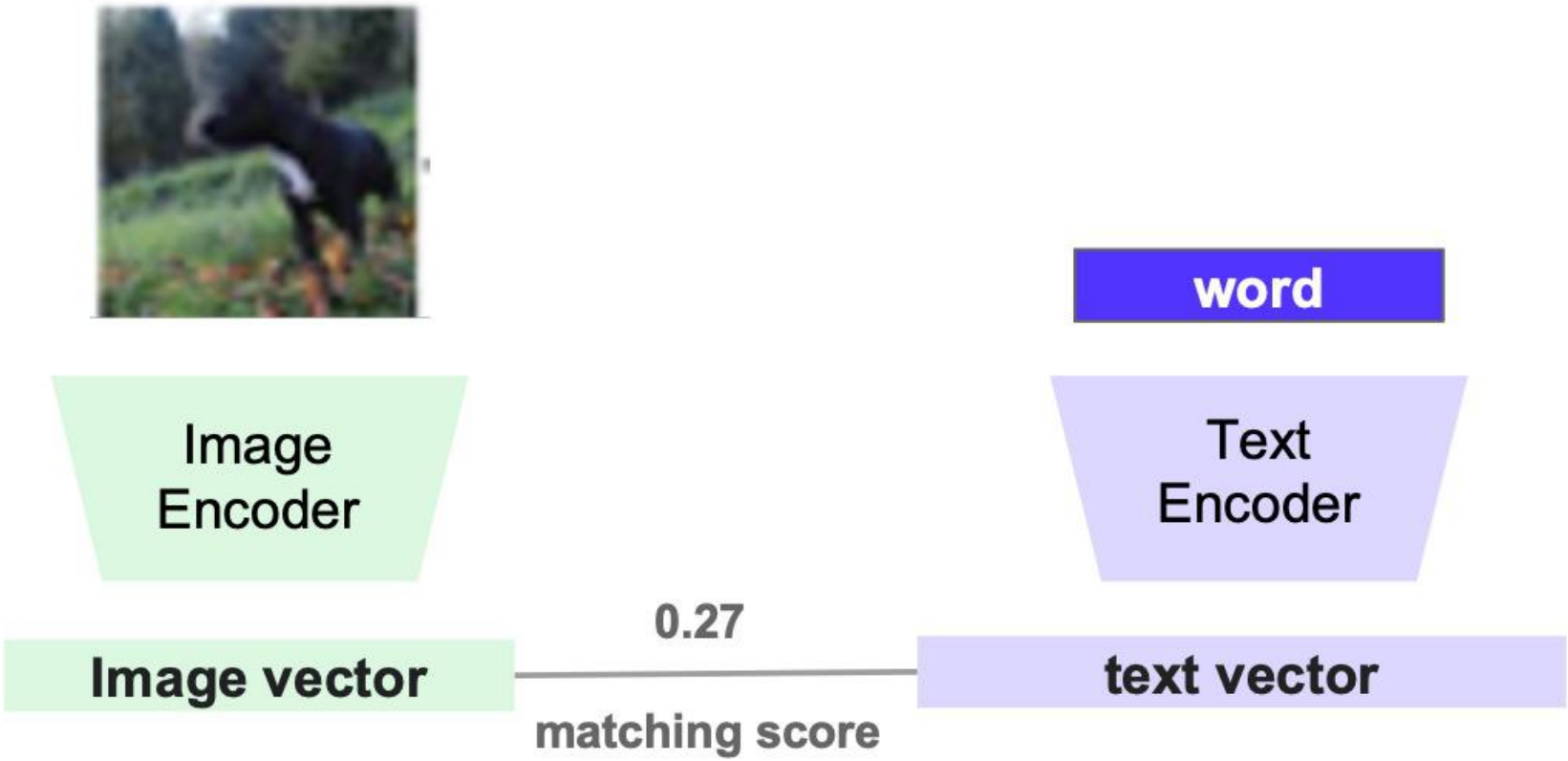


Pre-training tasks:
Contrastive Objective

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

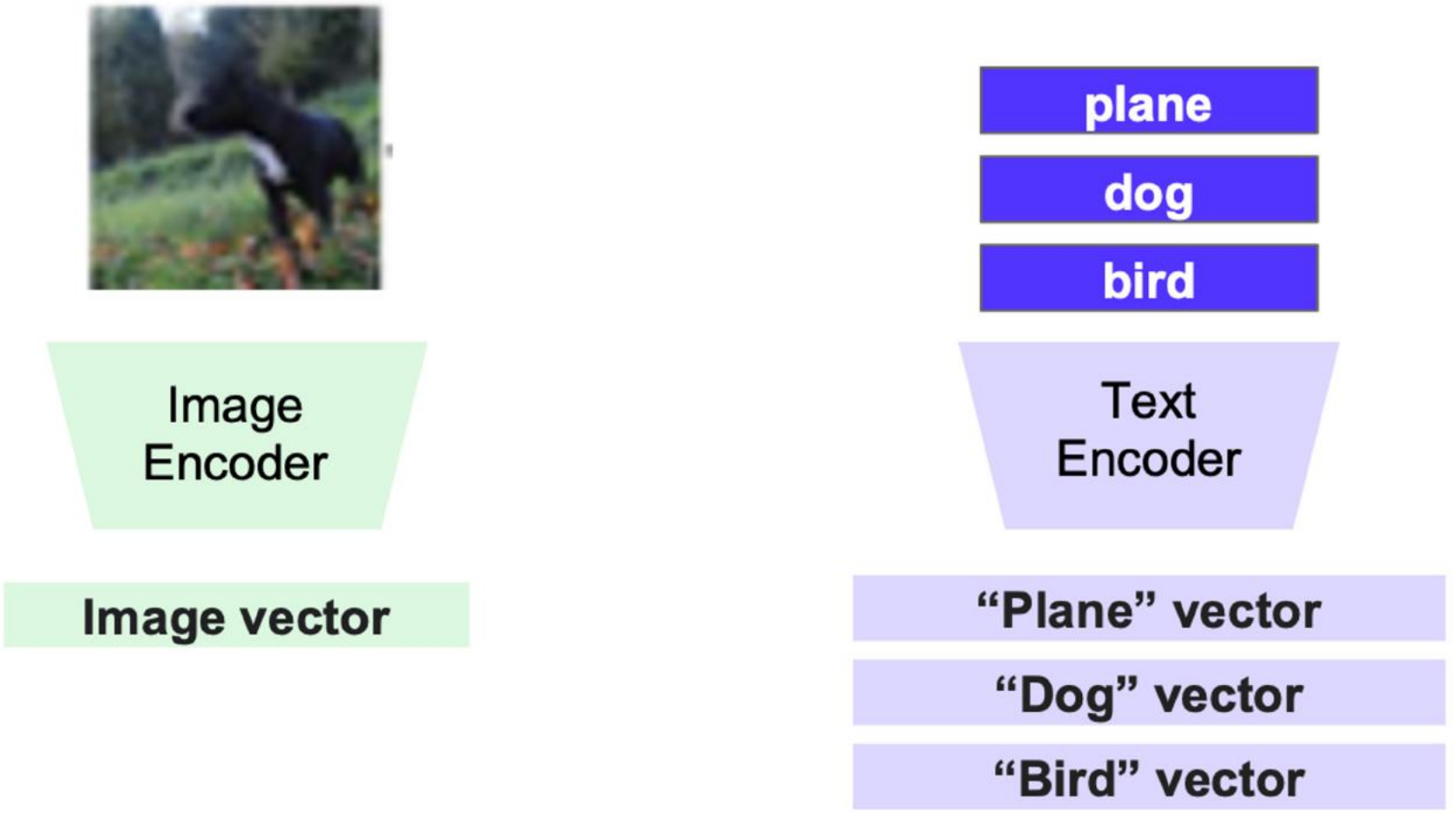
Out of the box classification (No fine-tuning)

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



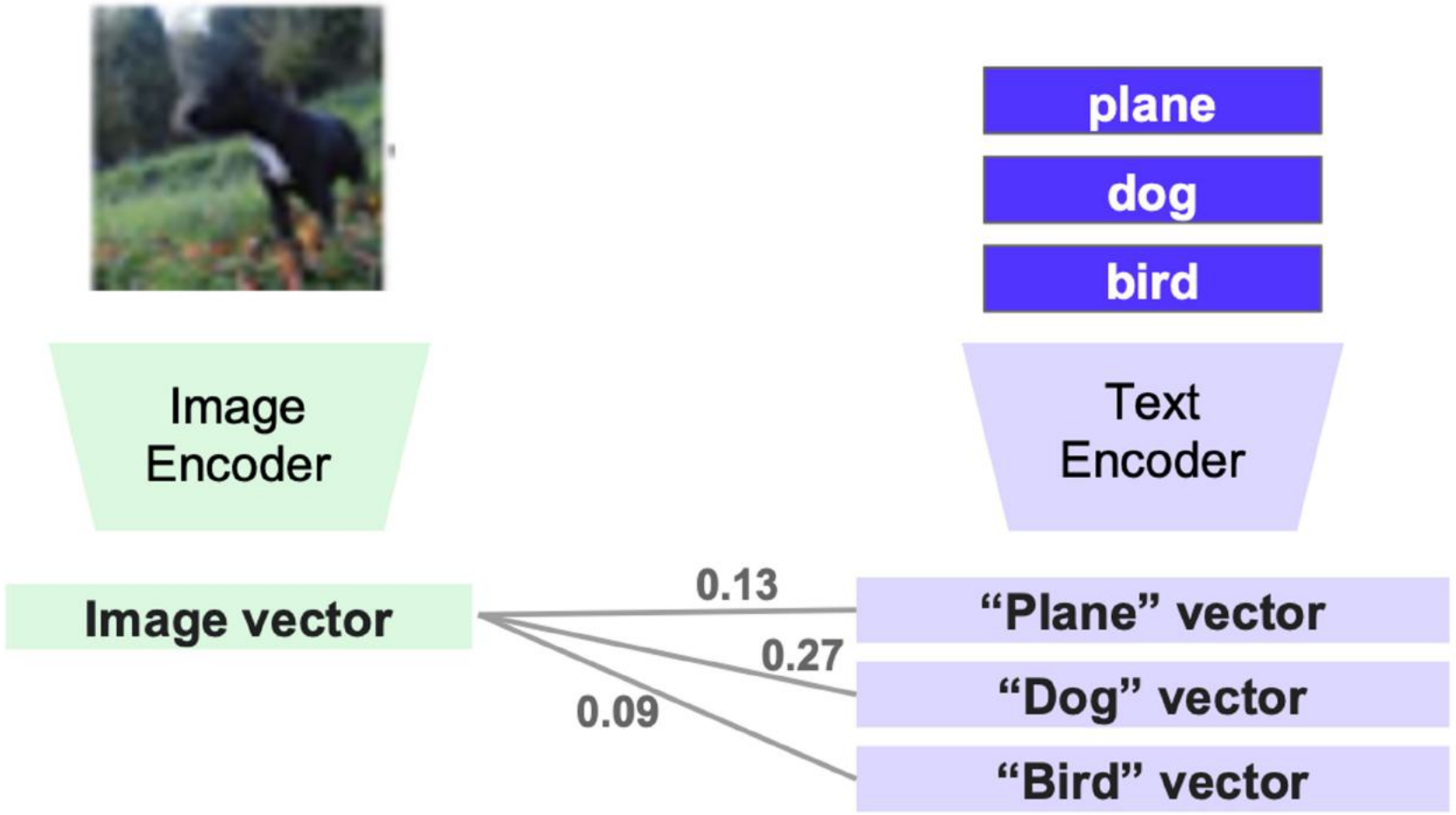
Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

Create a vector representation for *each* category!



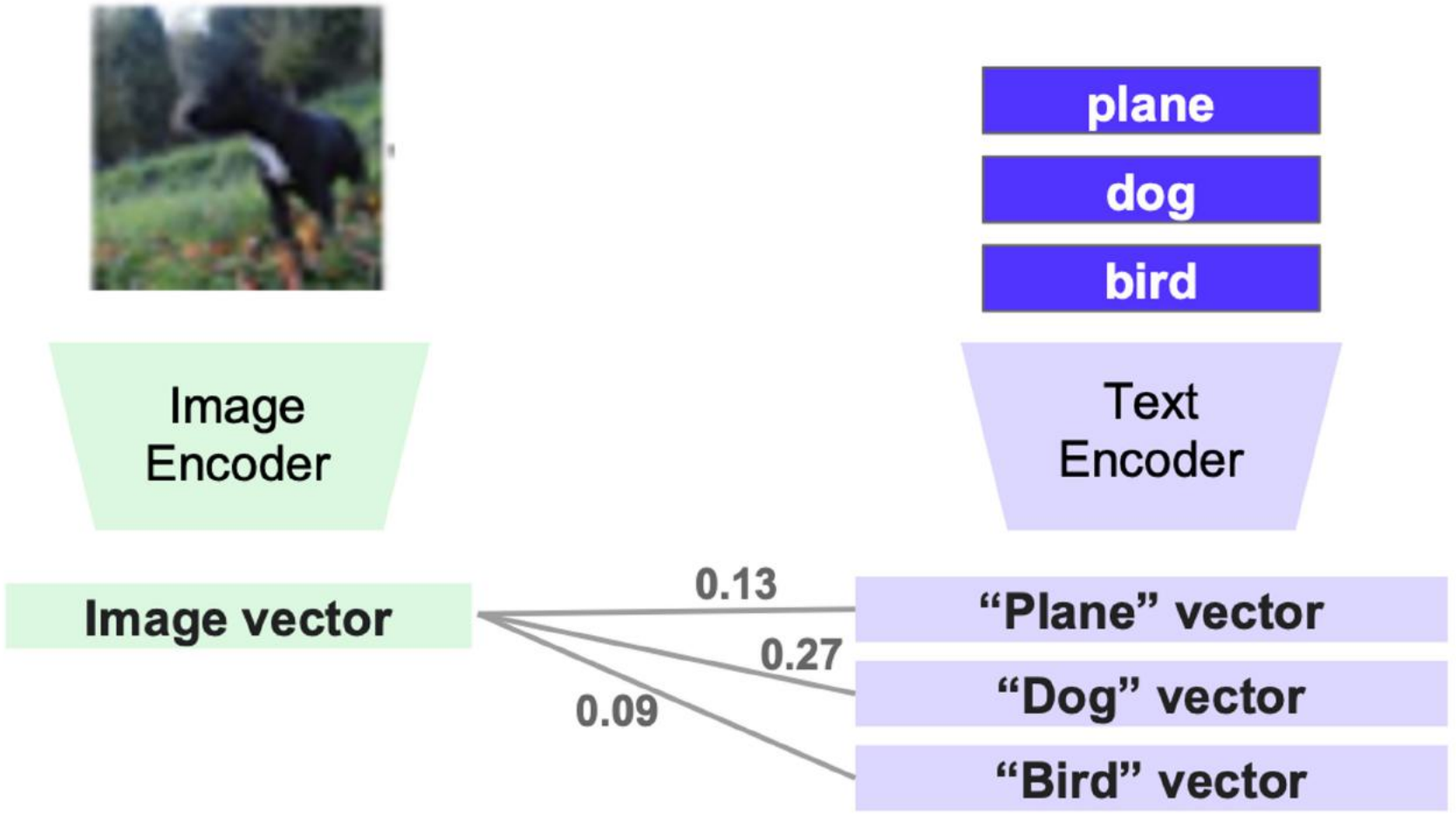
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Match a new image to the most similar vector



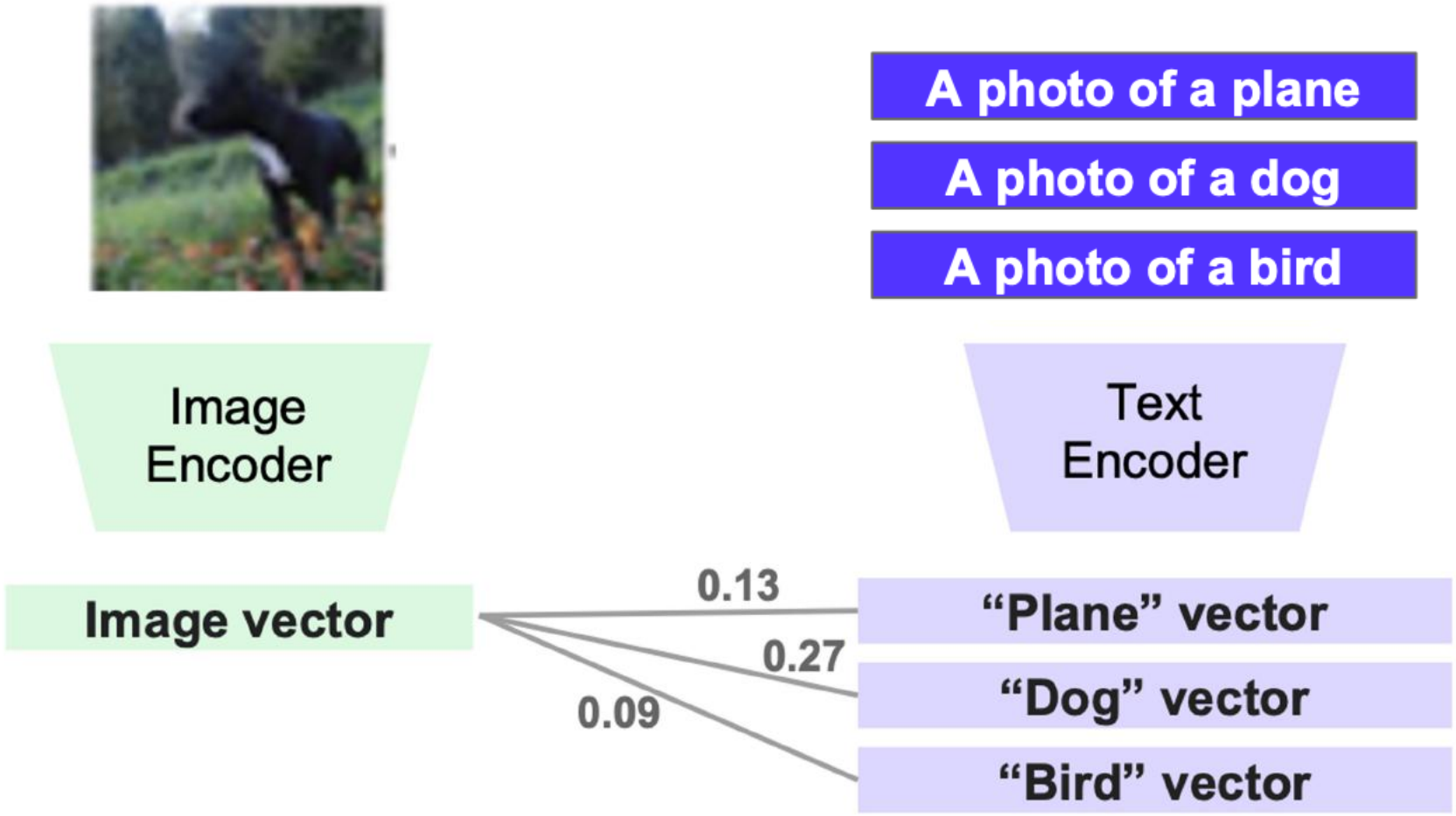
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

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



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]"



+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

Image Encoder

Image vector

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 | ...

Text Encoder

"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

+5% on ImageNet

Image vector

0.13

0.27

0.09

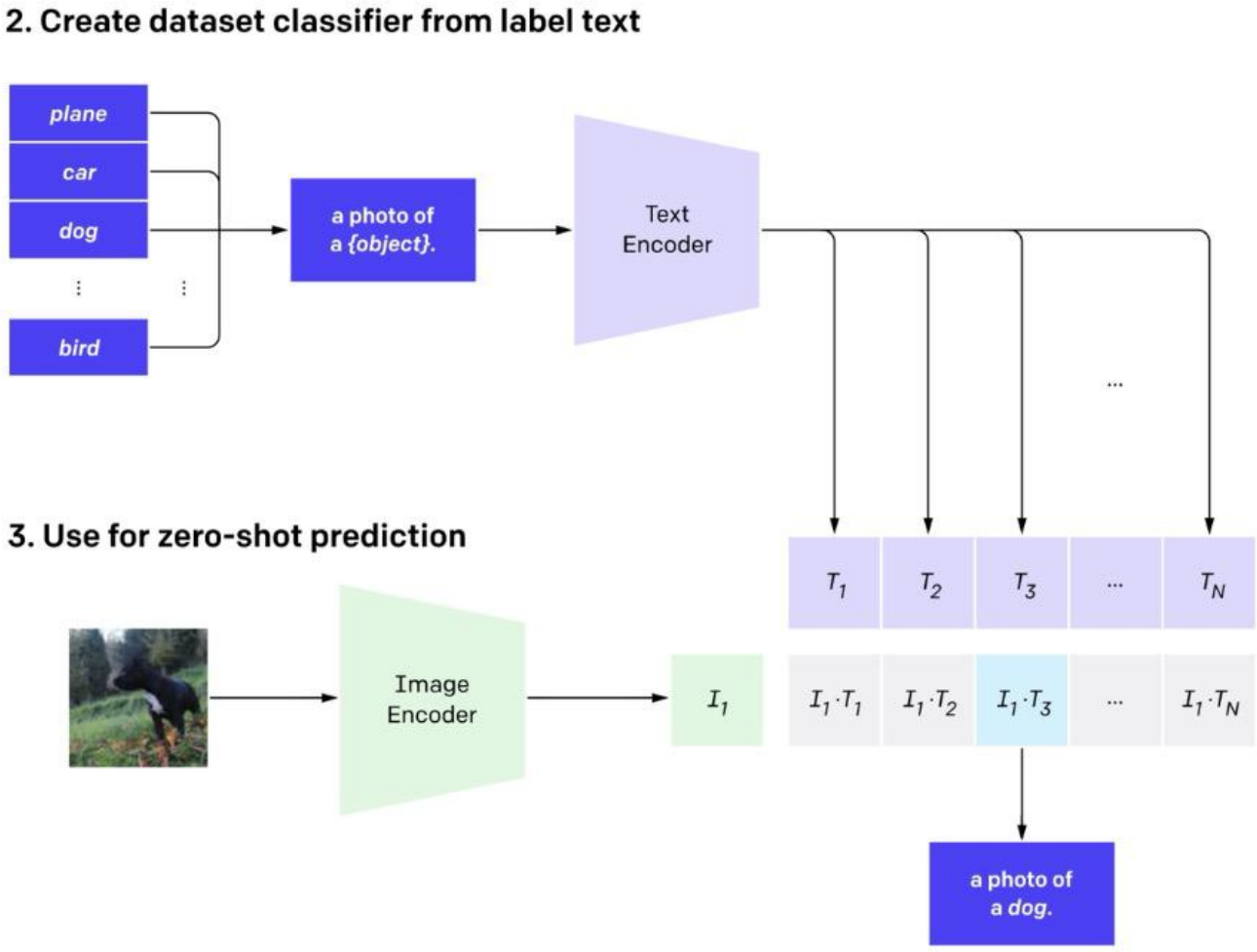
Mean "Plane" vector

Mean "Dog" vector

Mean "Bird" vector

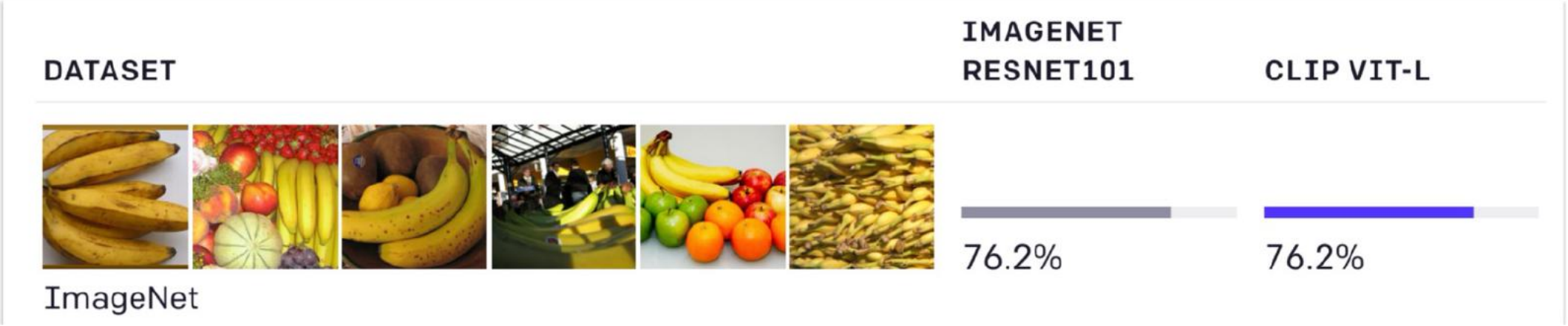
Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

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



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

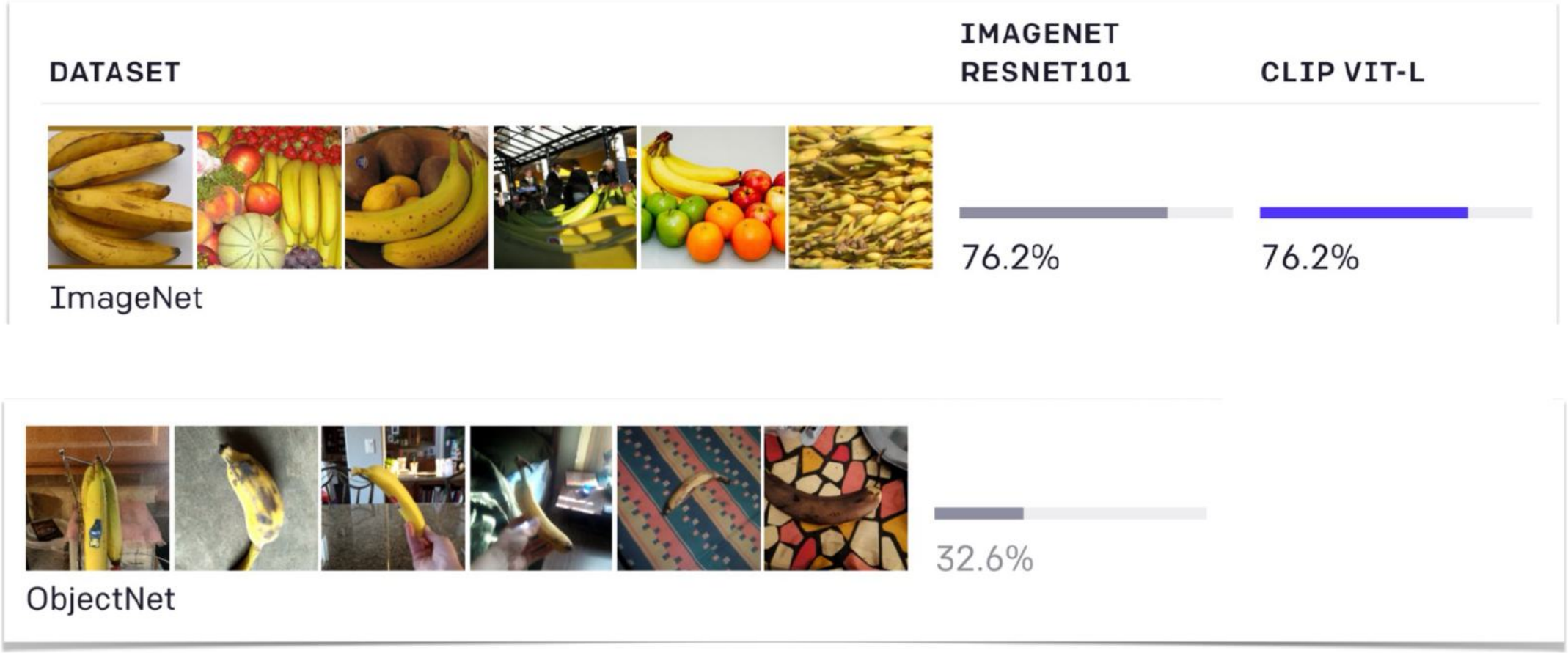
Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

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



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

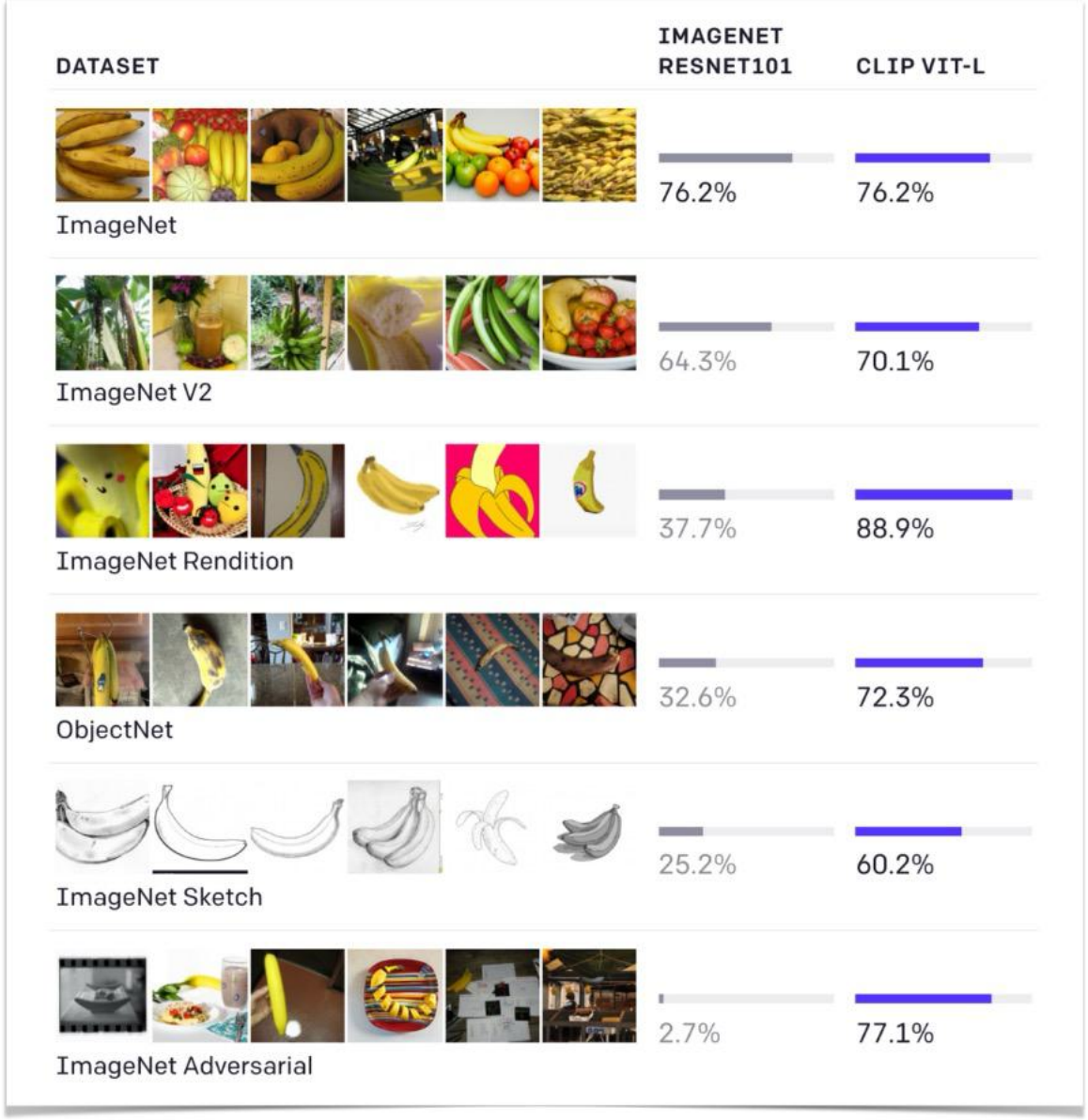
But CLIP zero-shot does so well!

Q. Why do you think that is?

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

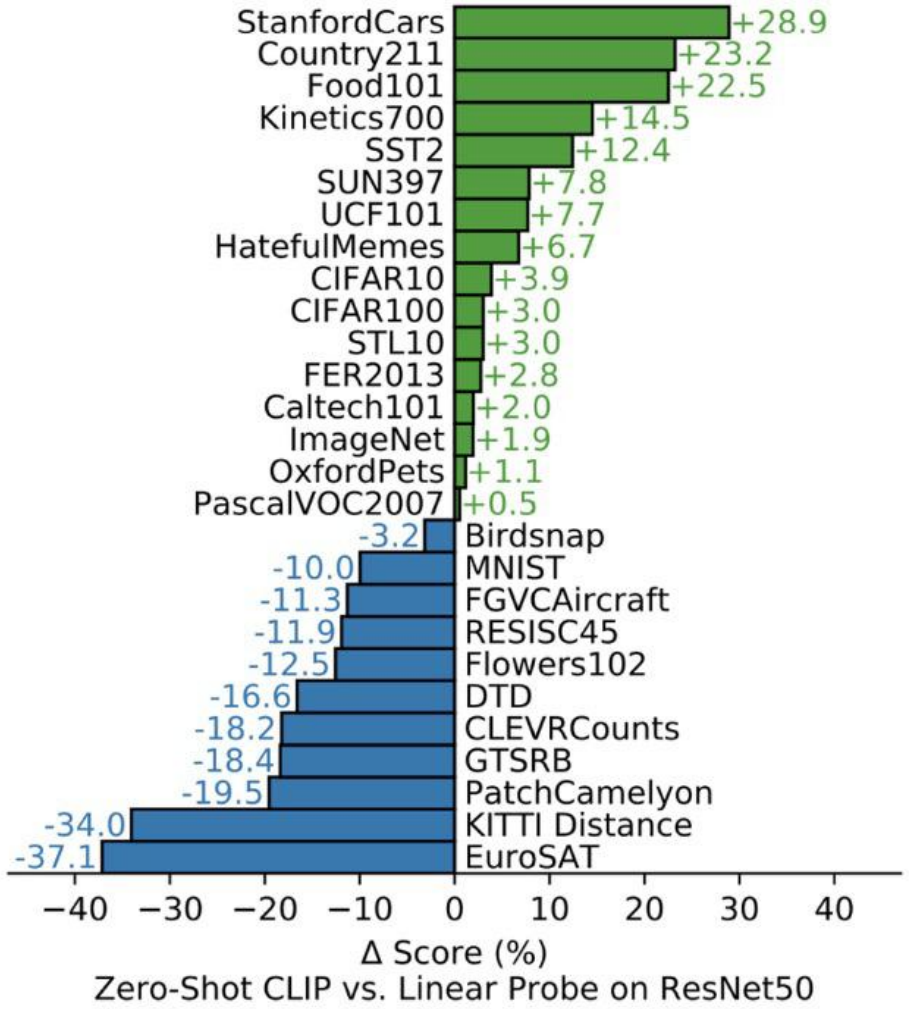
Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

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



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

Difference in performance between linear probe vs zero-shot



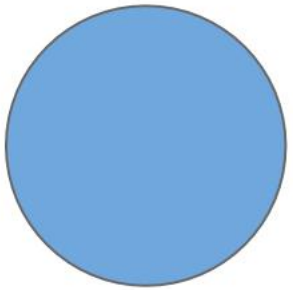
Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

Why does CLIP perform so well?

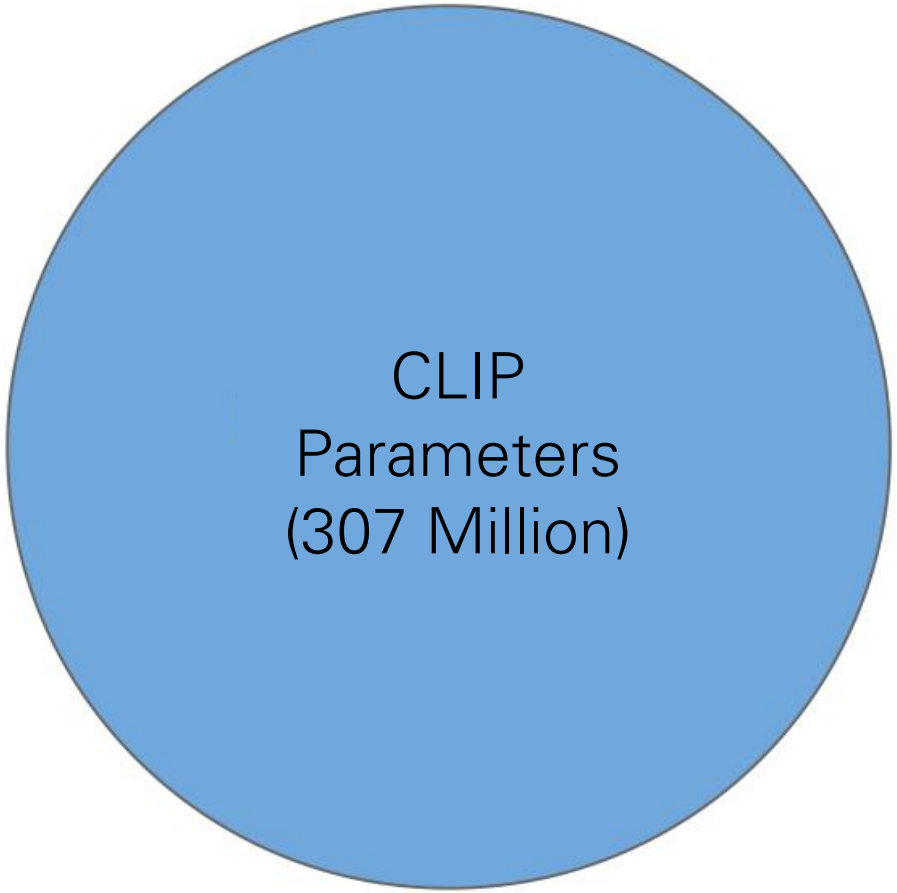
How can no labels beat labels??

Scale!

CLIP scaled up the model parameters with the Transformer architecture



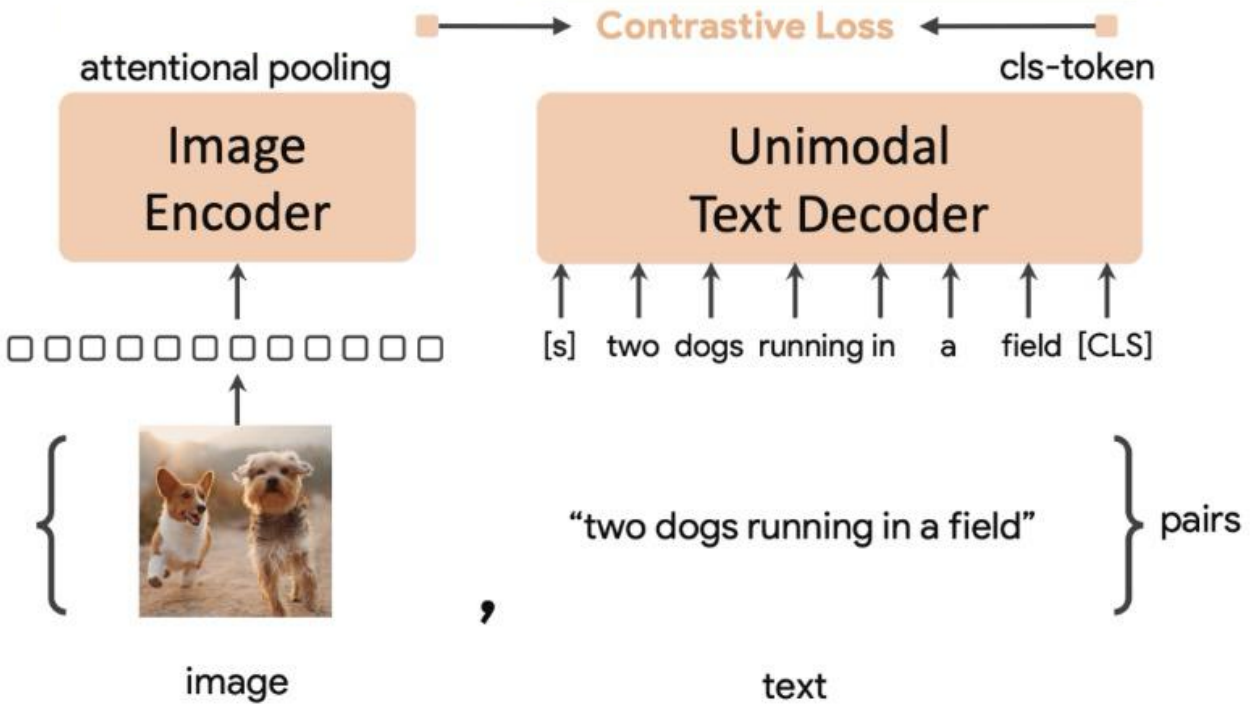
ImageNet ResNet Parameters
(44.5 Million)



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

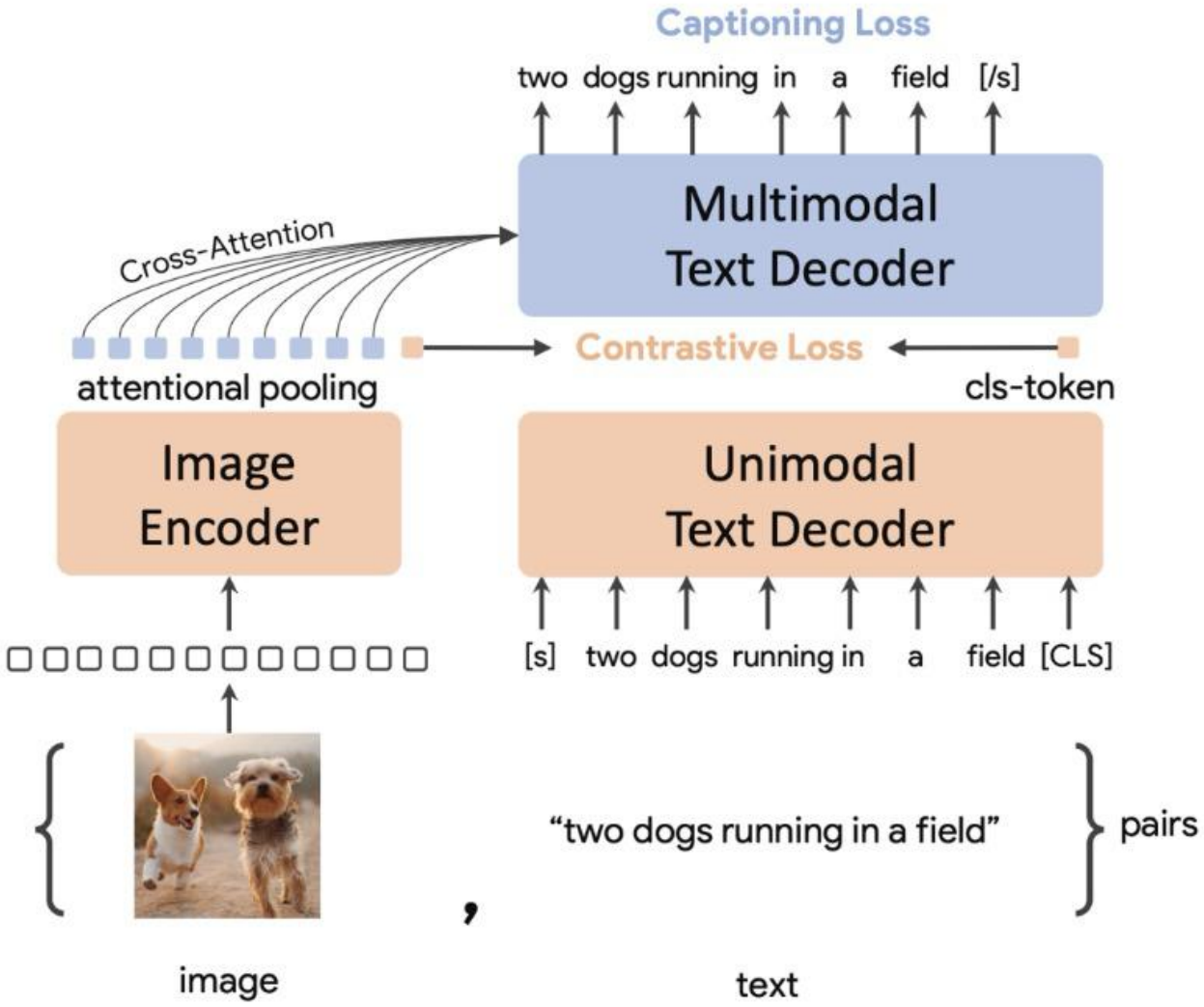
ImageNet ResNet Training Data
(1.28 Million)

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

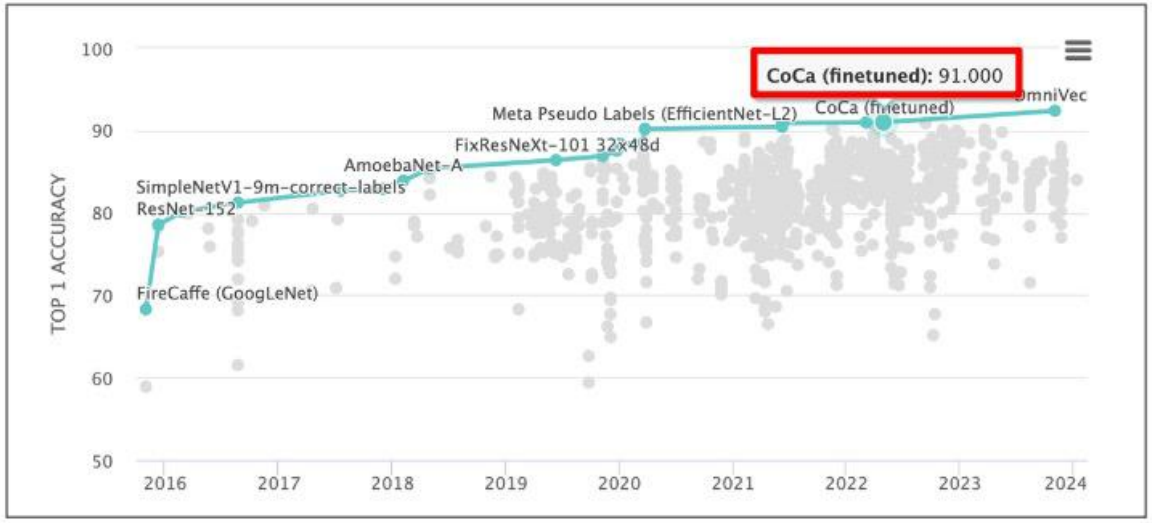
Model	ImageNet	ImageNet-A	ImageNet-R	ImageNet-V2	ImageNet-Sketch	ObjectNet	Average
CLIP [12]	76.2	77.2	88.9	70.1	60.2	72.3	74.3
ALIGN [13]	76.4	75.8	92.2	70.1	64.8	72.2	74.5
FILIP [61]	78.3	-	-	-	-	-	-
Florence [14]	83.7	-	-	-	-	-	-
LiT [32]	84.5	79.4	93.9	78.7	-	81.1	-
BASIC [33]	85.7	85.6	95.7	80.6	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].

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

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**



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

Advantages of CLIP-style models

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

- Open-vocabulary (zero-shot generalization)

- 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

BLIP

the grass is eating the horse 81%

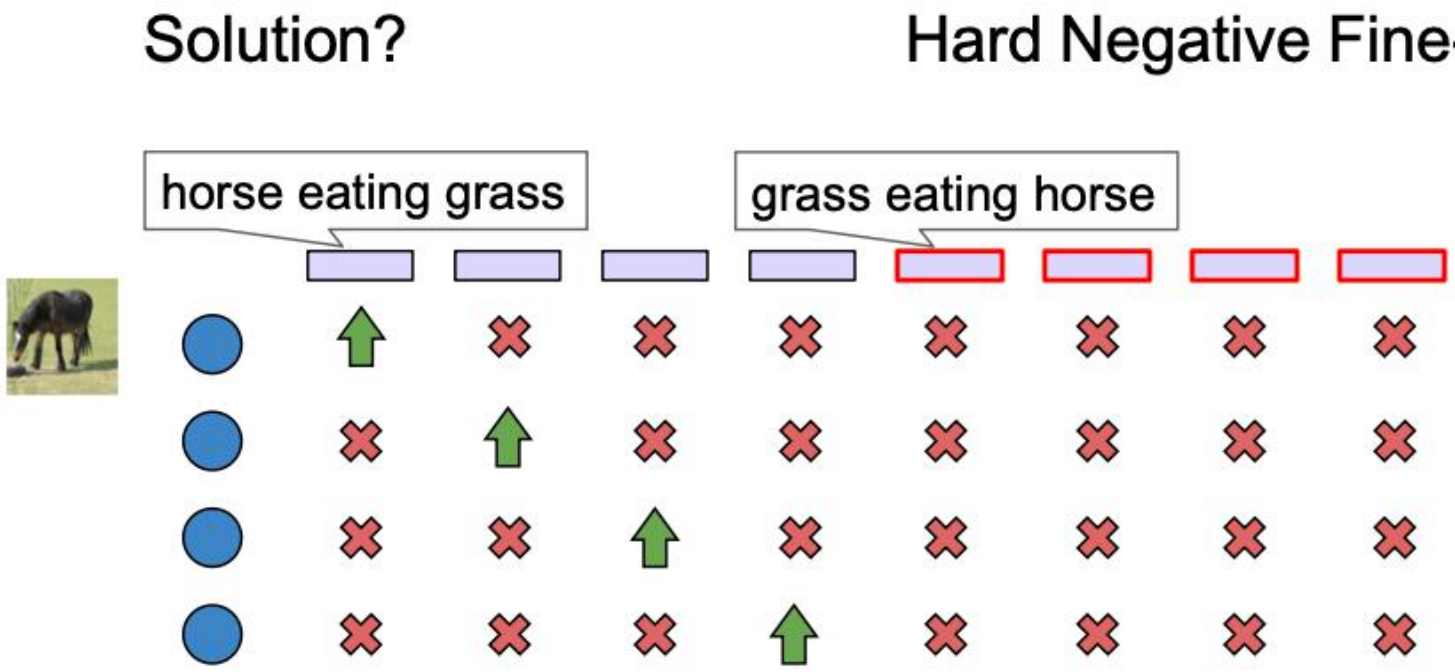
the horse is eating the grass 78%

...

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



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" ✓

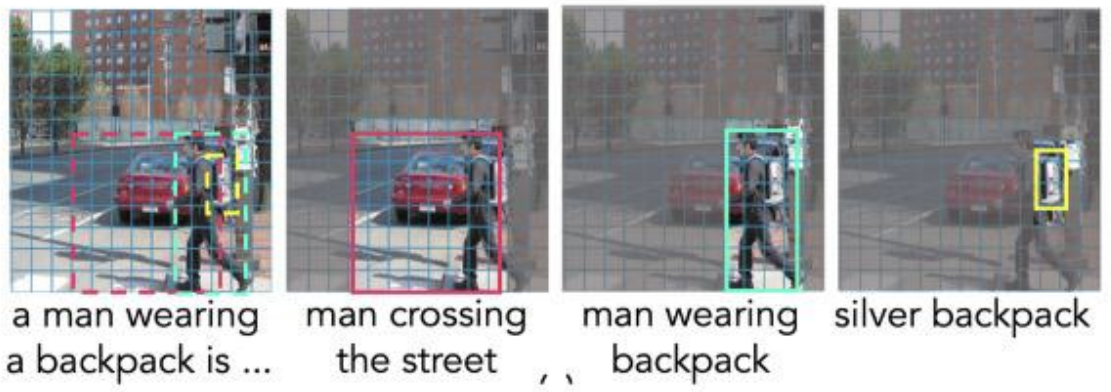
"house plants" ✗

"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

LM + Vision
Multimodal LLMs

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)

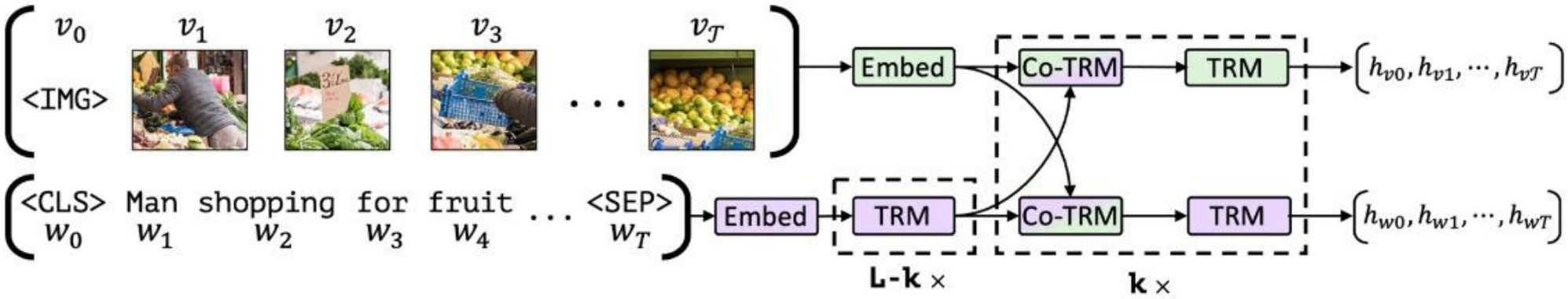
- Can we build a model that can accept images and text as input, and then output text?

  → 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

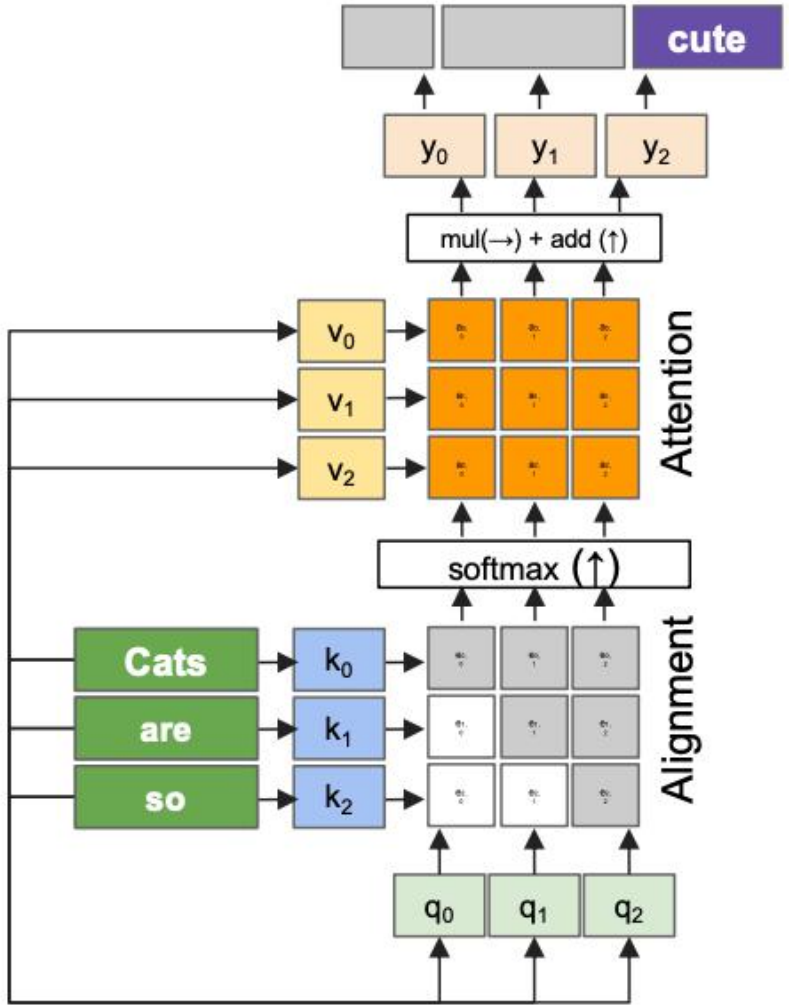
- Vision-Language Models didn't start with LLaVA!

- They go as far back as 2019 → ViLBERT

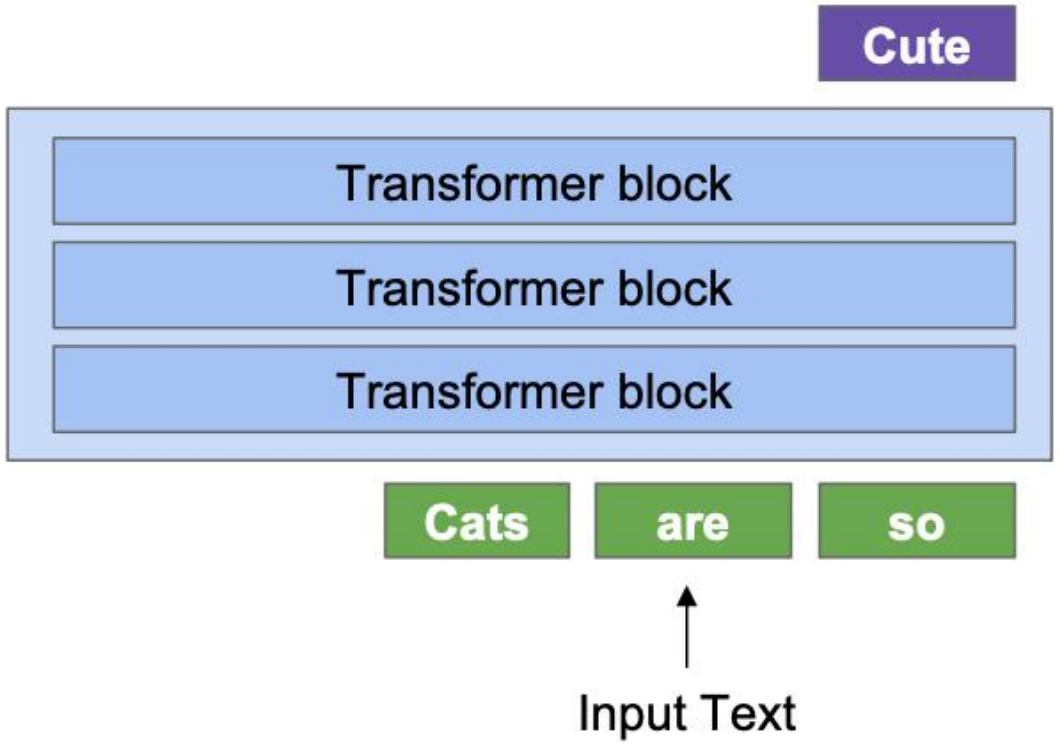
- BUT, they had to finetune for each task separately, with non-trivial task-specific methods (e.g., Mask-RCNN bounding box re-ranking for RefCOCO)

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

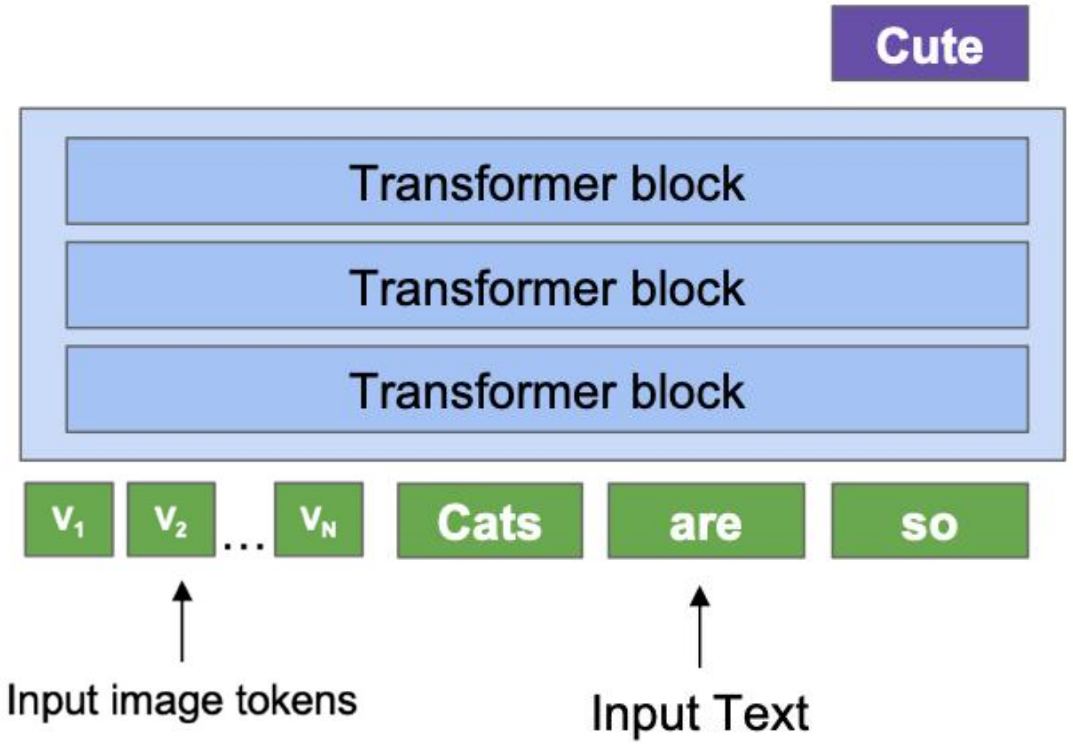
LLaVA uses the autoregressive nature of LLMs



Recall how transformers decode language



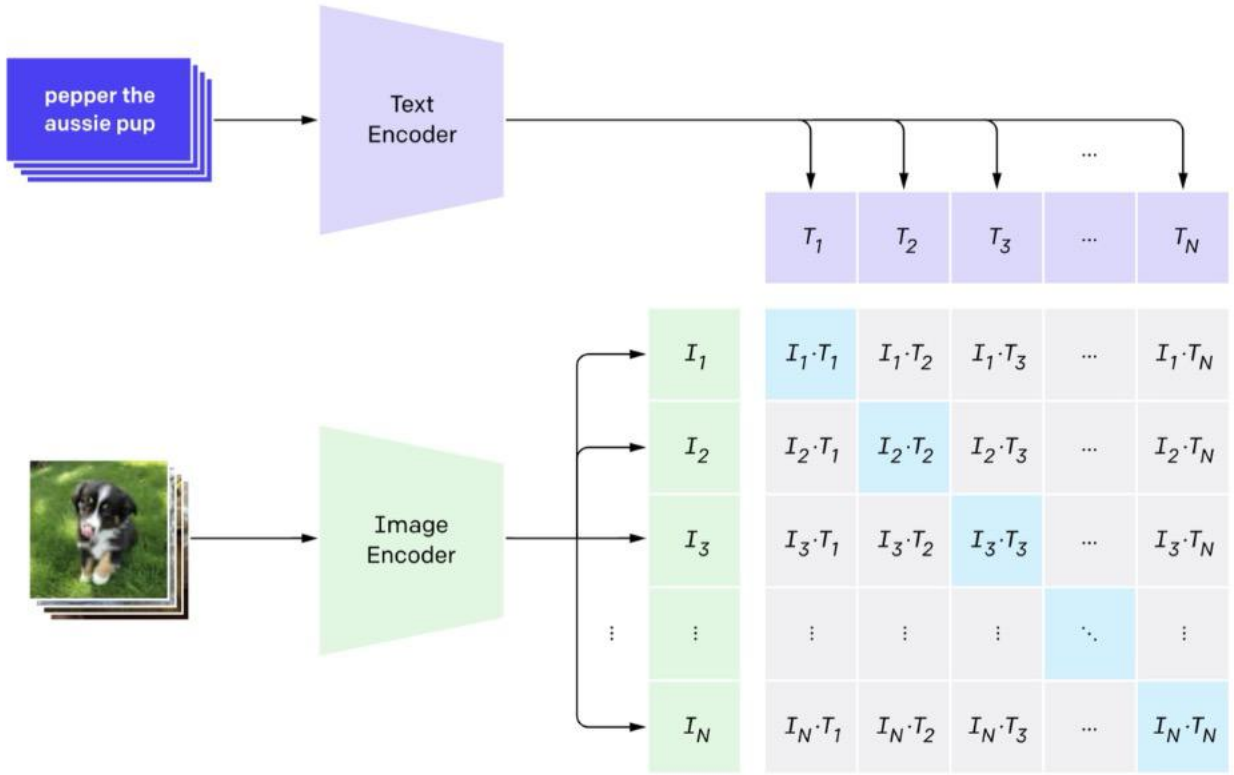
Key idea behind LLaVA – add visual information to the LLM



Which image tokens work best here?

The CLIP encoder is a good option!

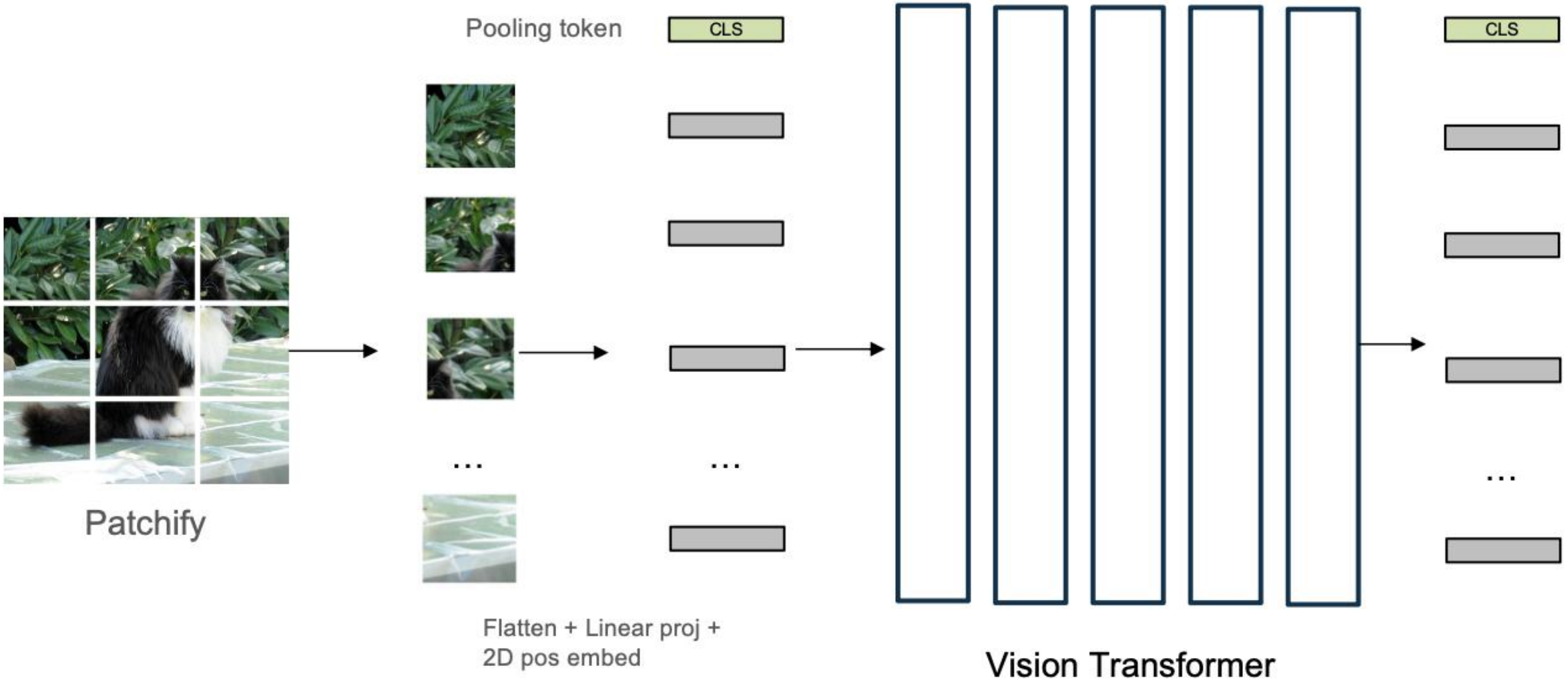
1. Contrastive pre-training



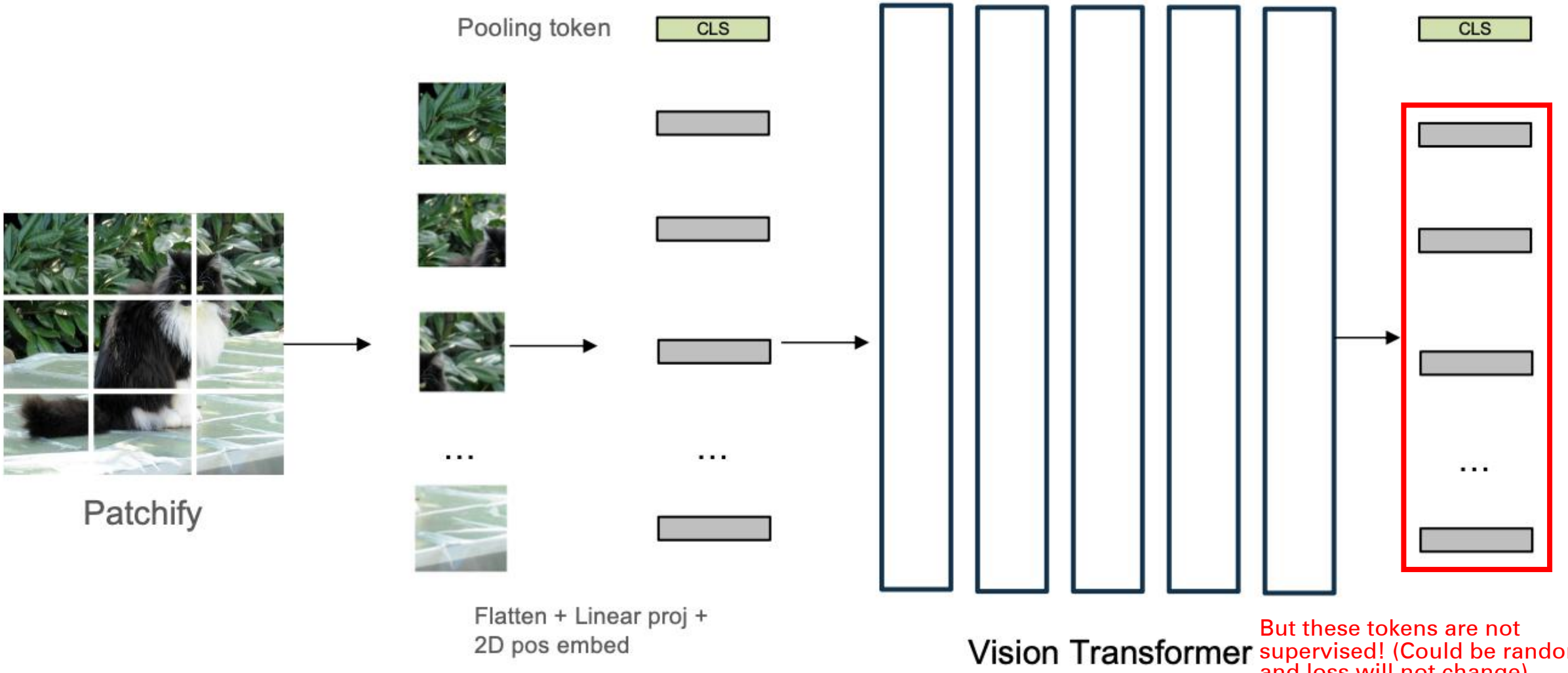
At the end of training, you have a model that will give you a similarity score between an image and a text

What features should we use from CLIP?

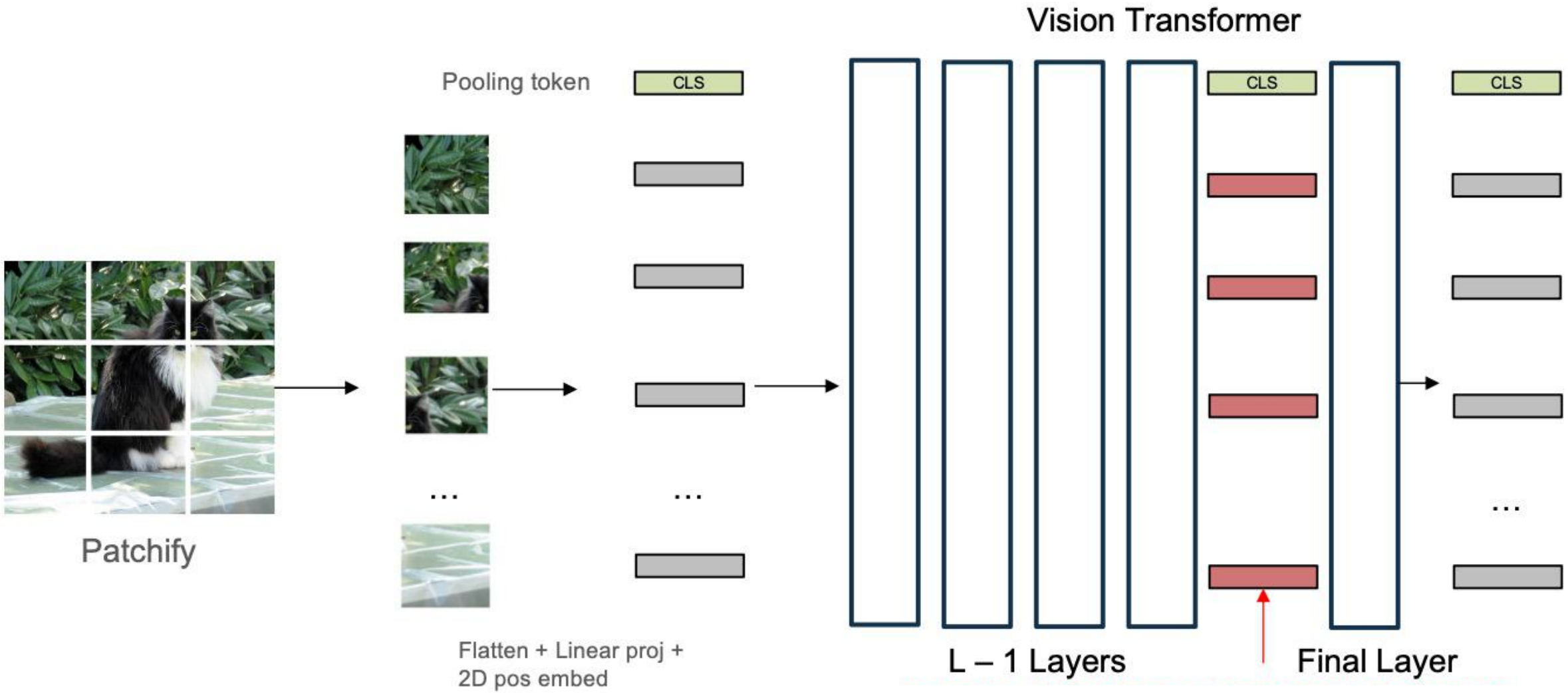
Extract Image Features from CLS token for contrastive learning



What features should we use from CLIP?

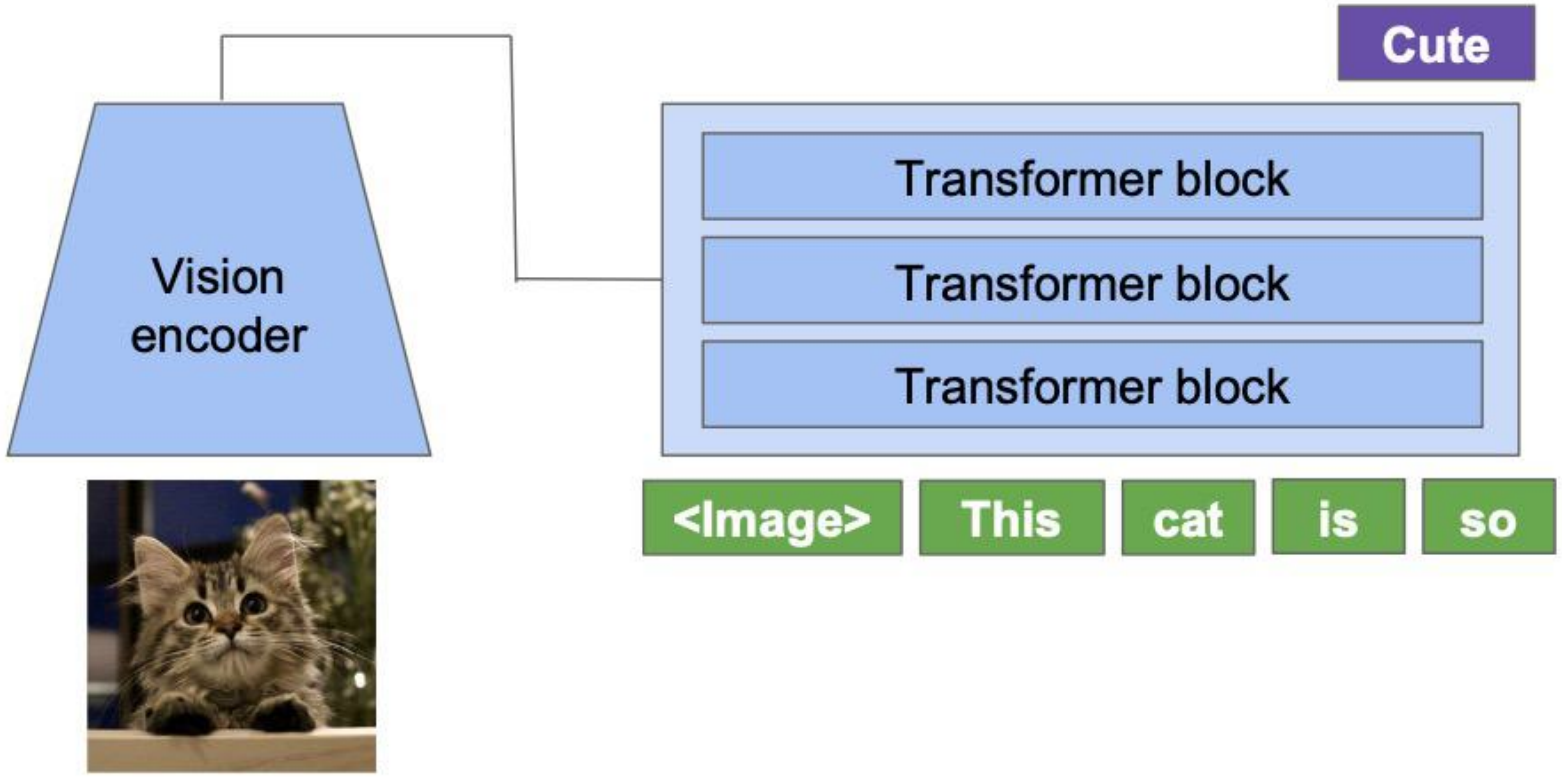


Use Penultimate Layer!

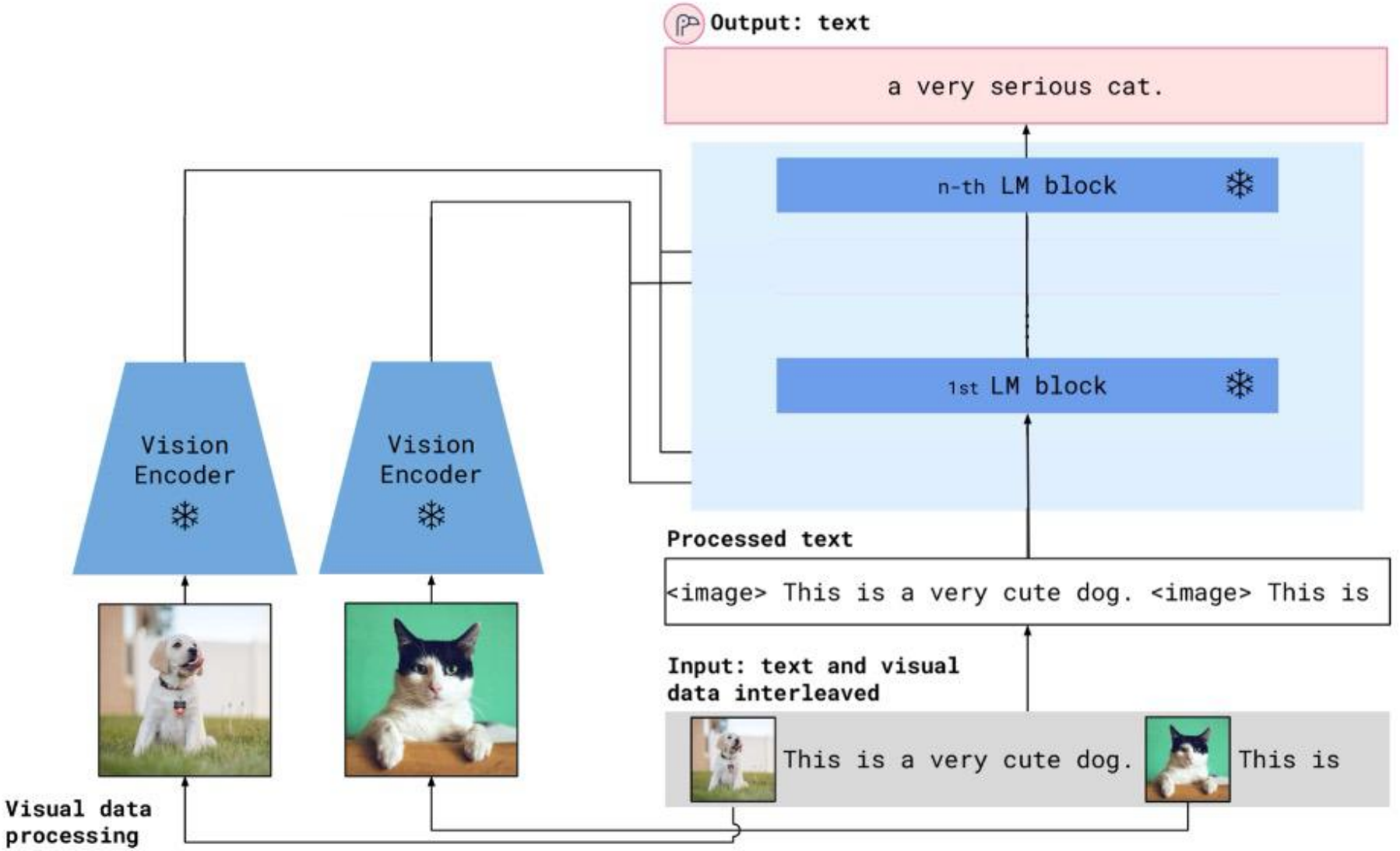


In practice, these tokens preserve spatial and linguistic information best for LLMs. Can drop CLS for slight gains.

Flamingo followed up with a new way to fuse visual features

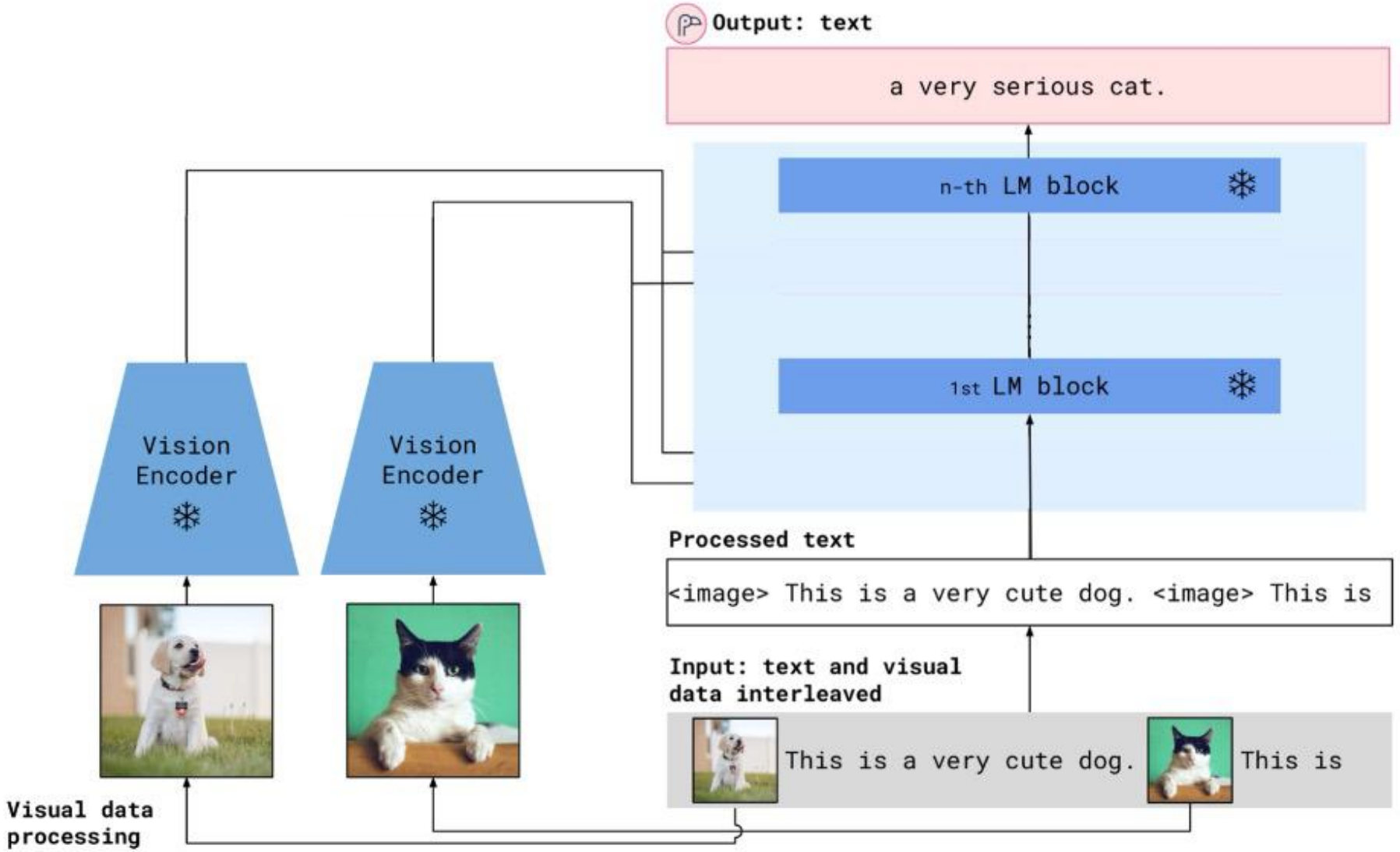


Pre-trained parts of Flamingo



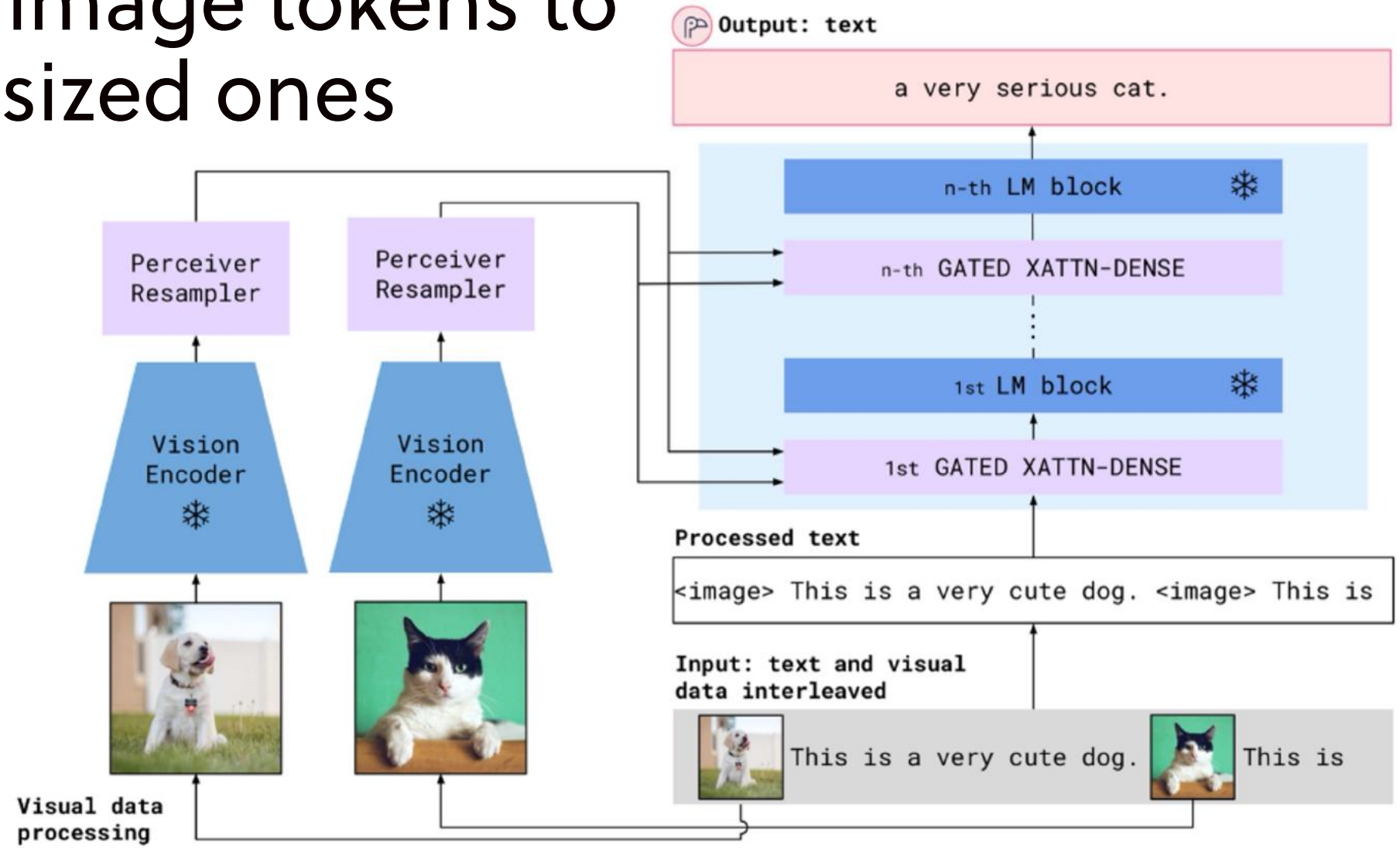
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

There are 2 learned parts in Flamingo



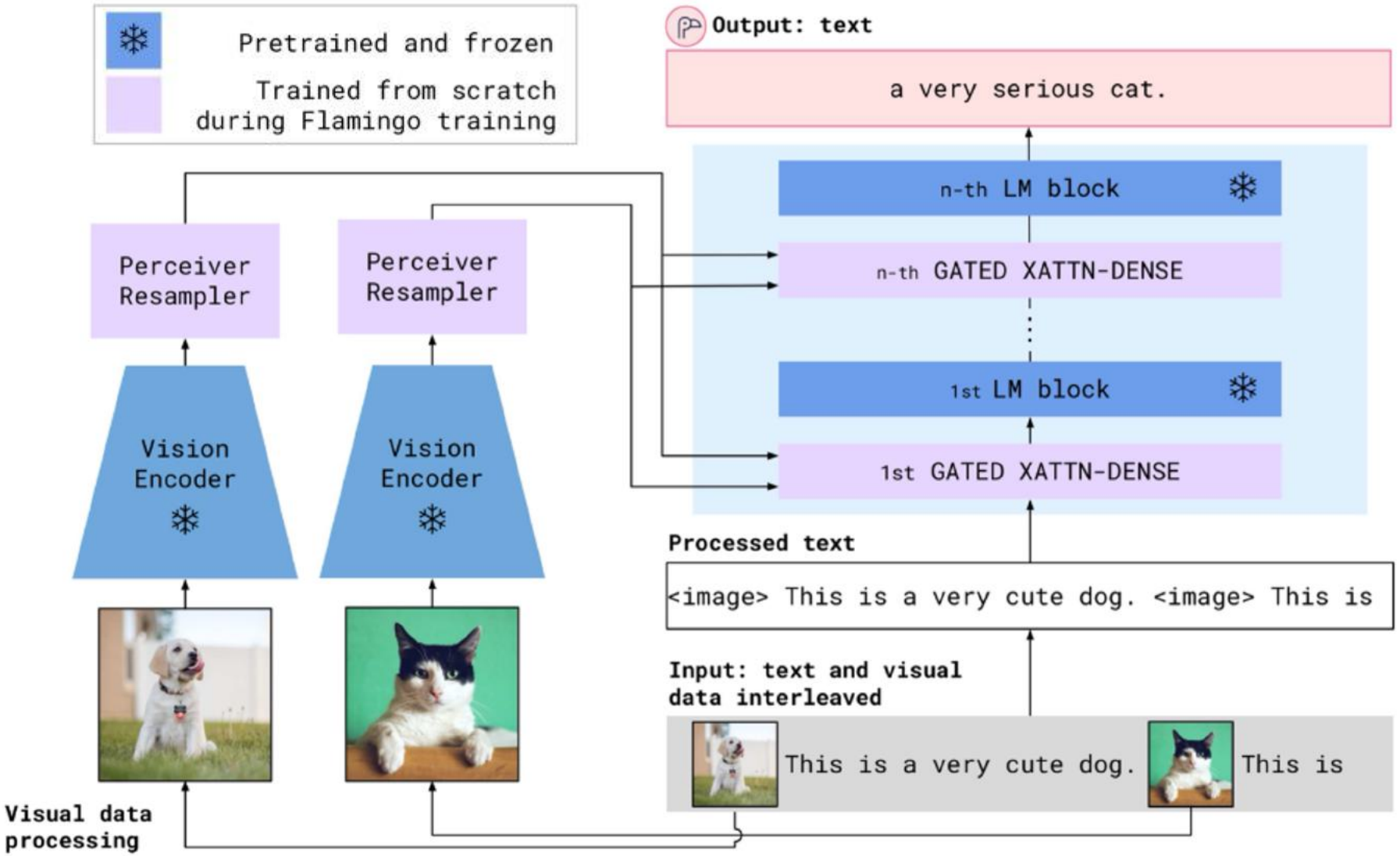
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Perceiver sampler converts variable sized image tokens to fixed sized ones



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

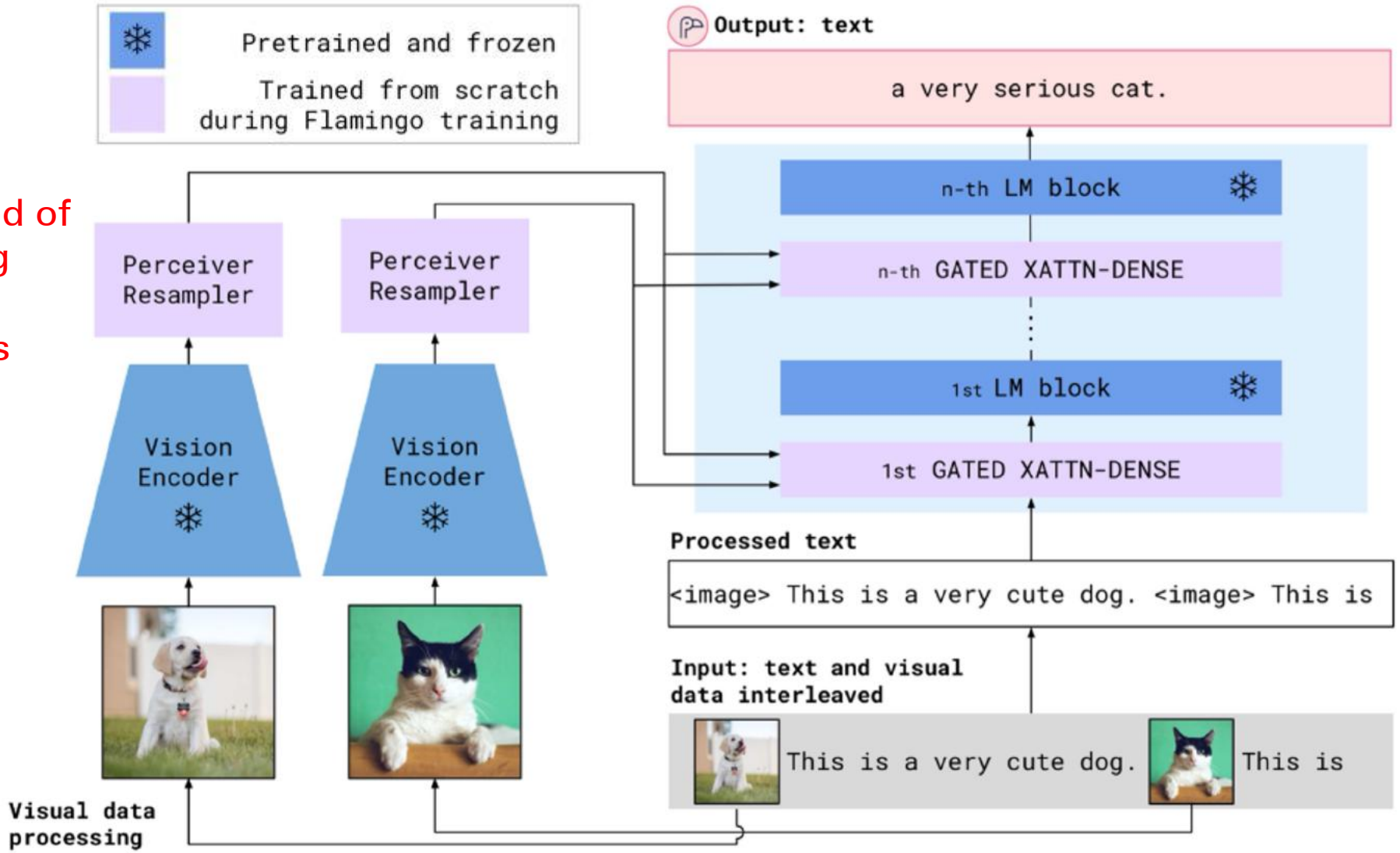
Flamingo full architecture



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

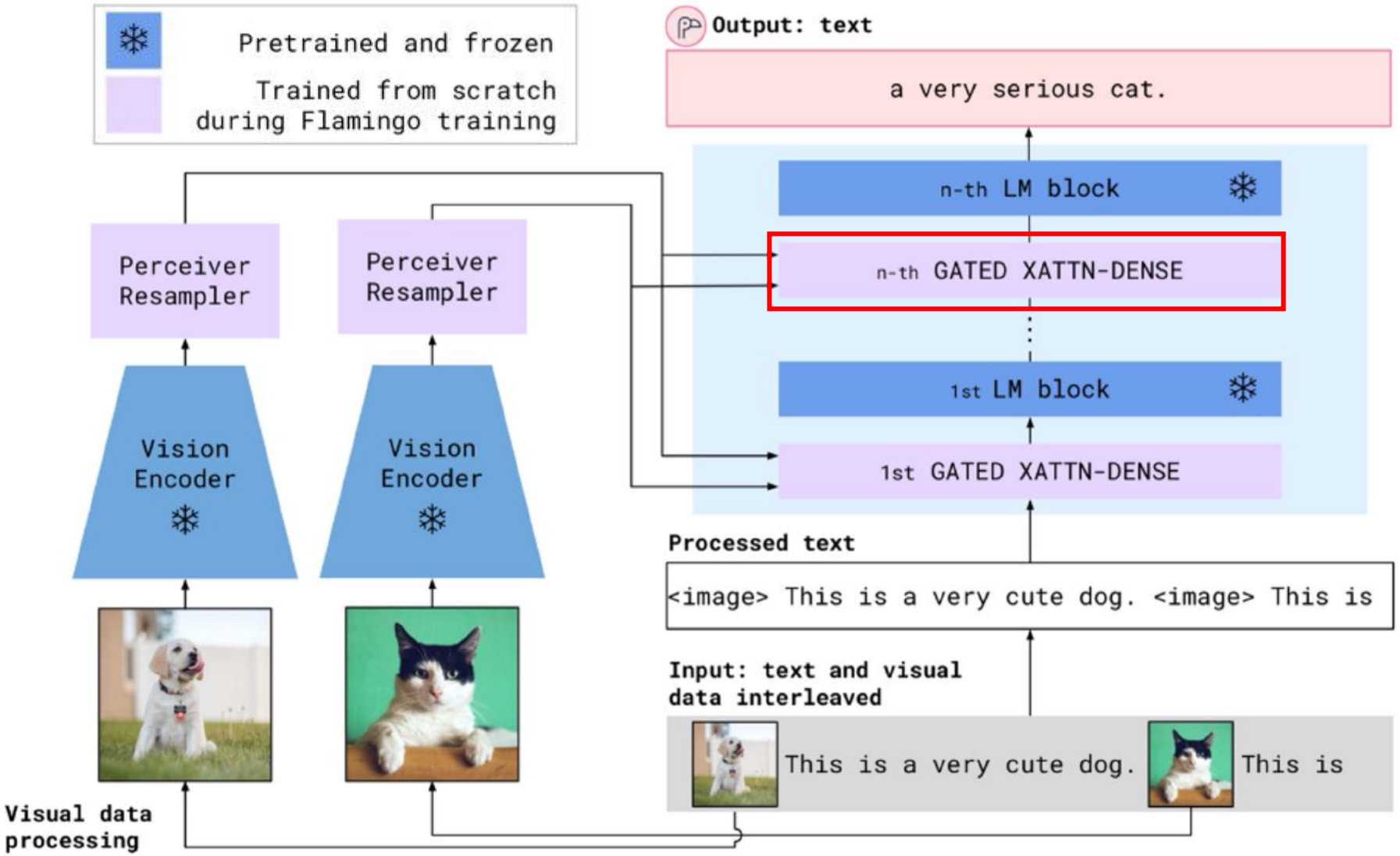
Flamingo full architecture

Learned method of down-sampling image/video representations



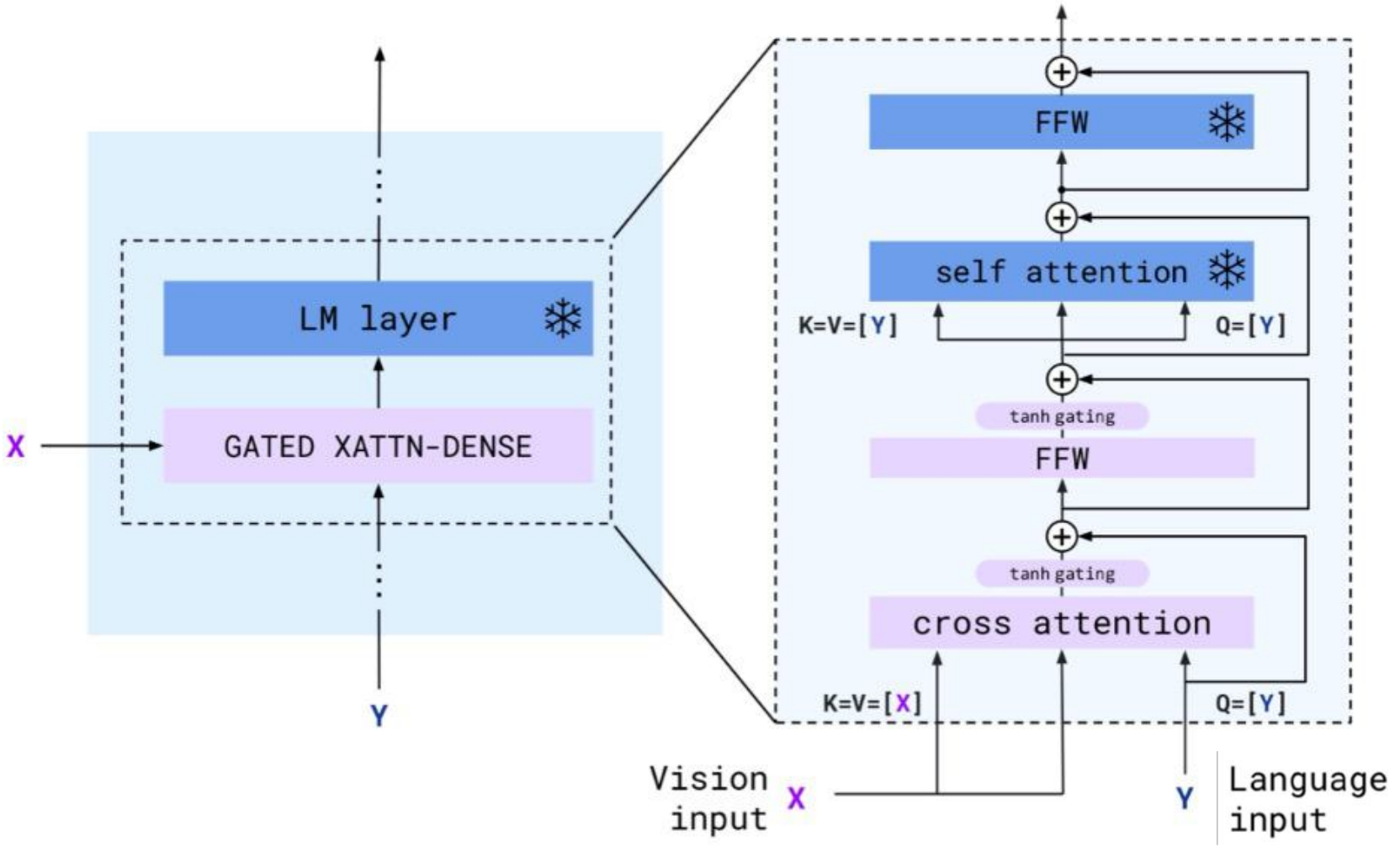
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo full architecture



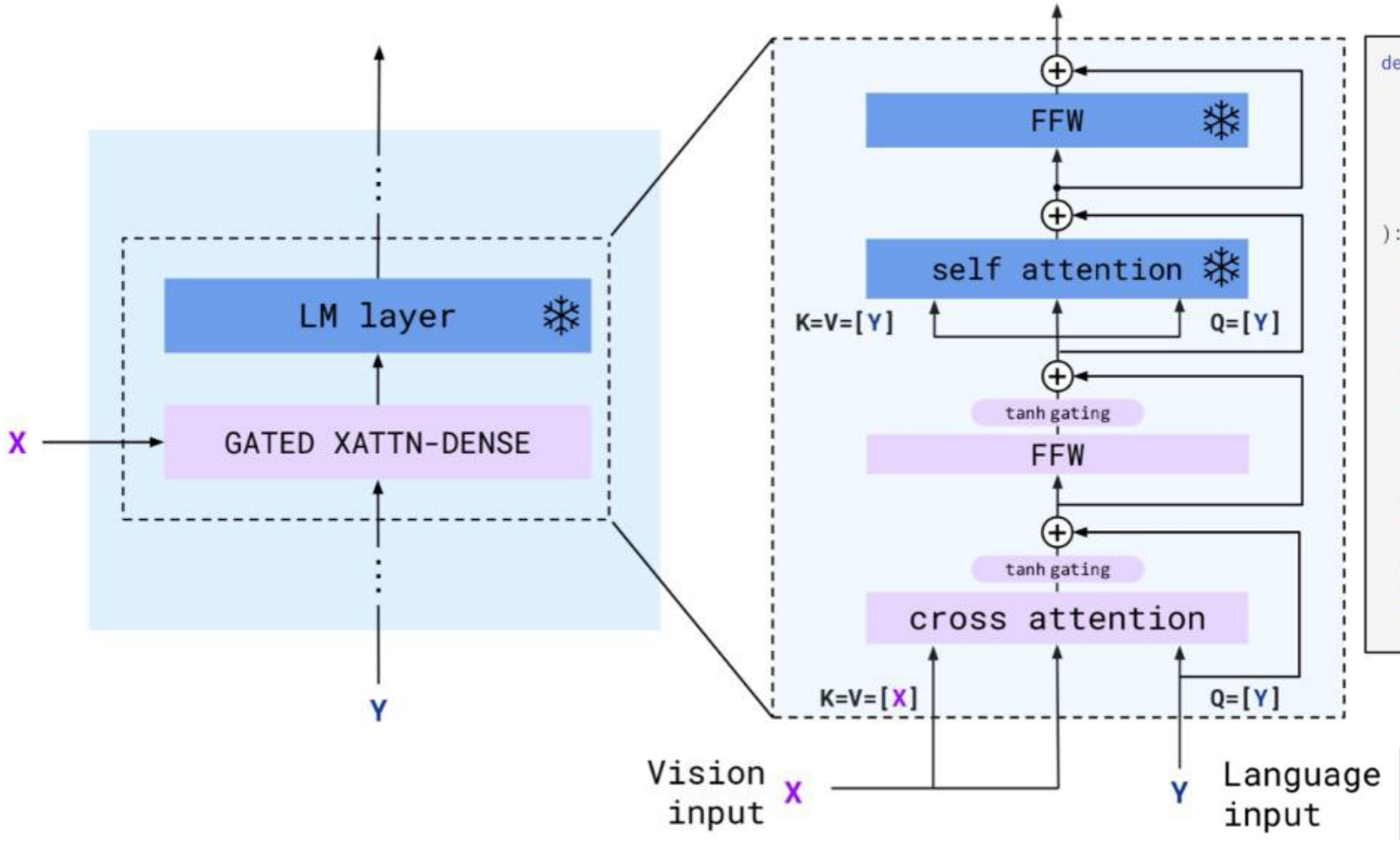
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo gated cross-attention



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

Flamingo gated cross-attention

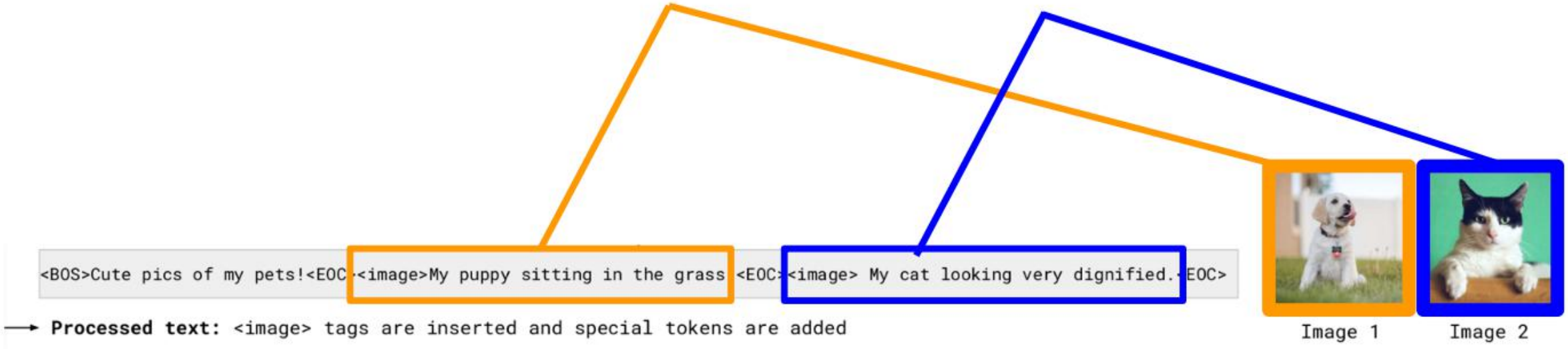



```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter - init at 0.
    alpha_dense, # ffw gating parameter - init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 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

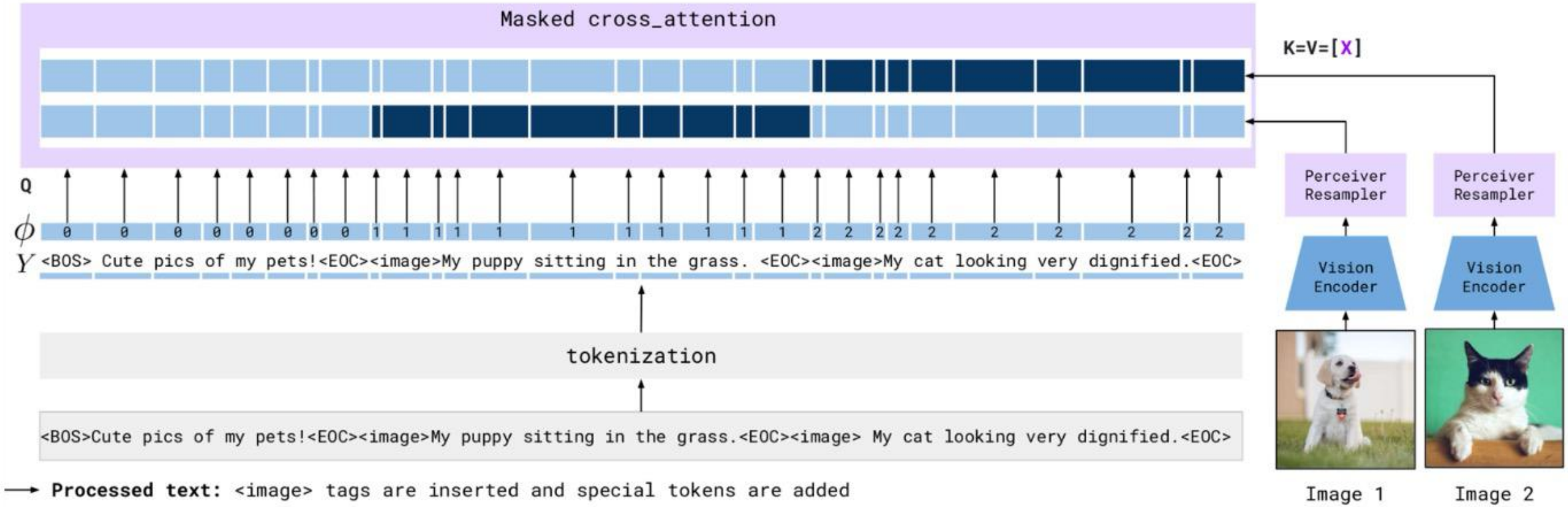
with special tags <image>, <eos> to indicate when a new image shows up or the text ends.



Flamingo masked attention

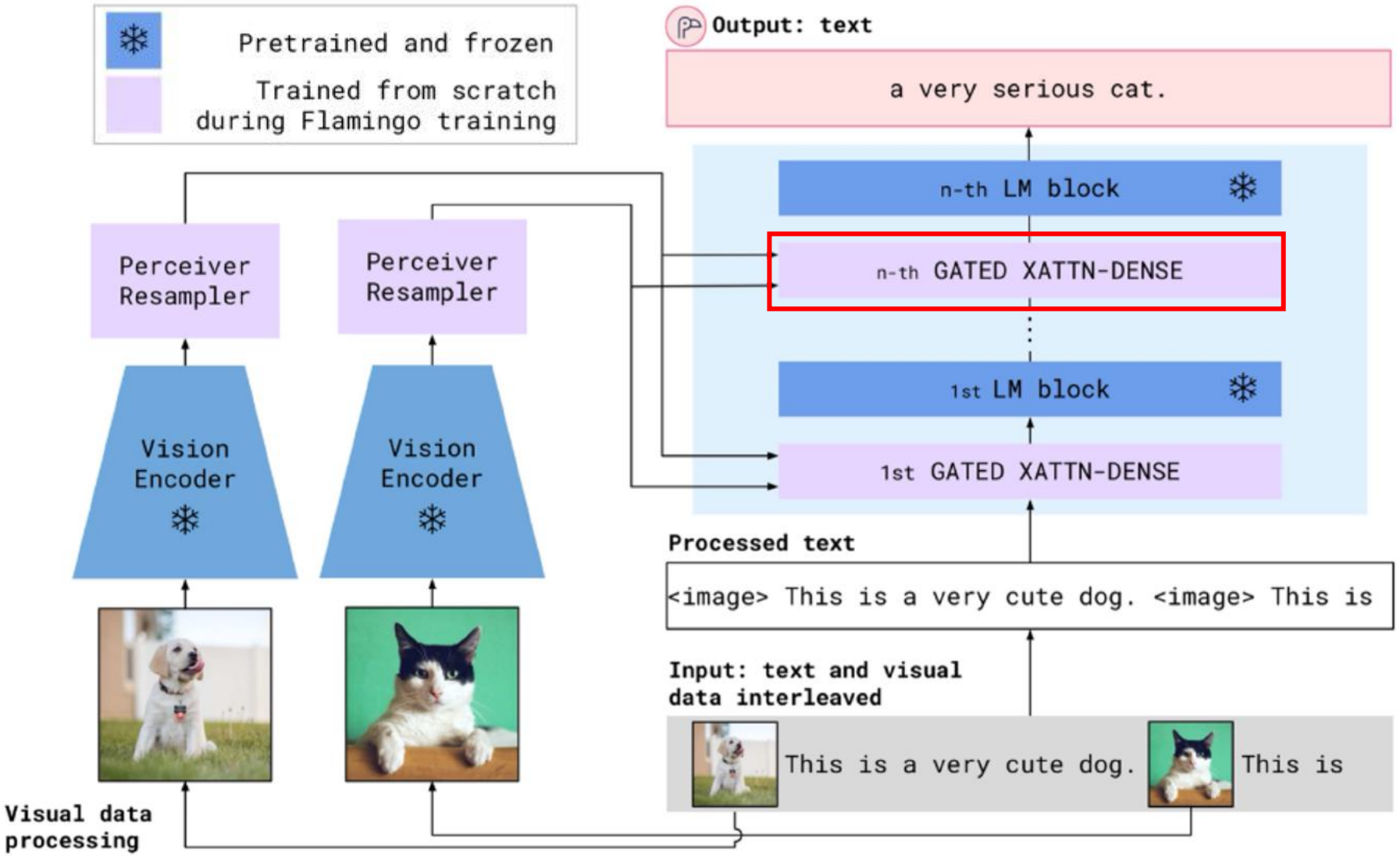
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo masked attention



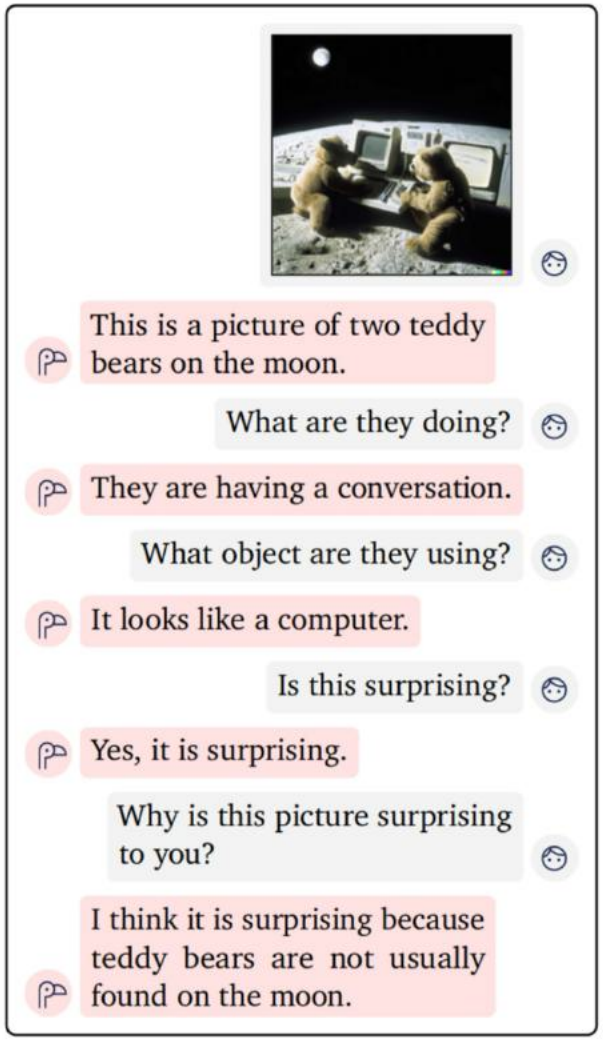
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

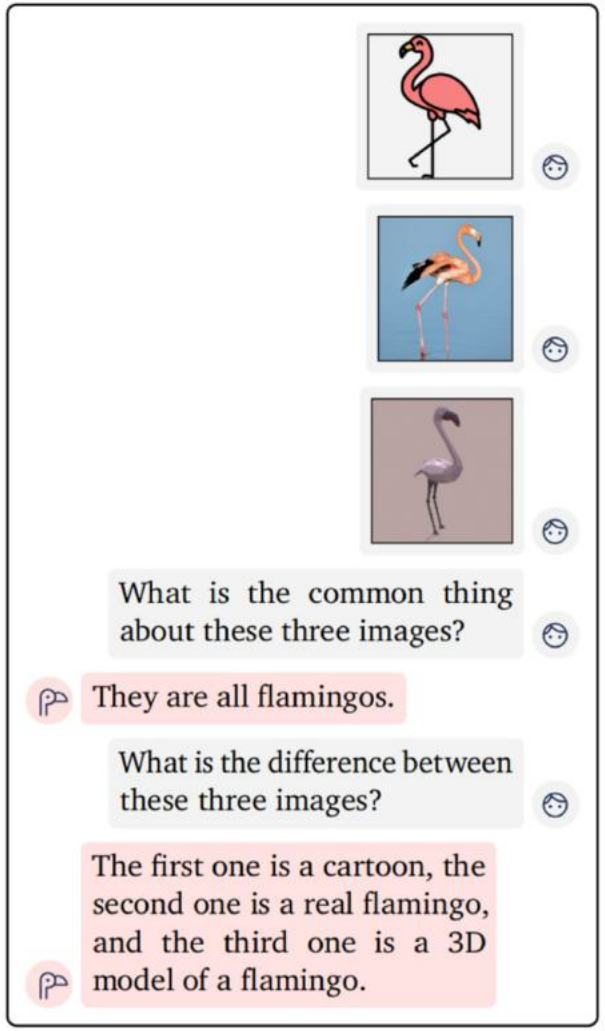
Flamingo full architecture

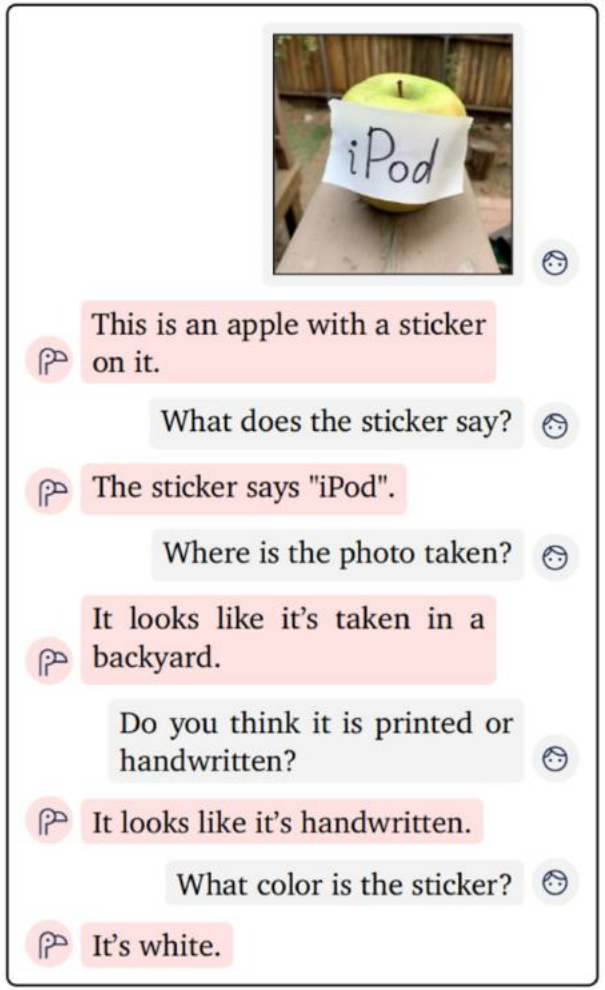


Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo results

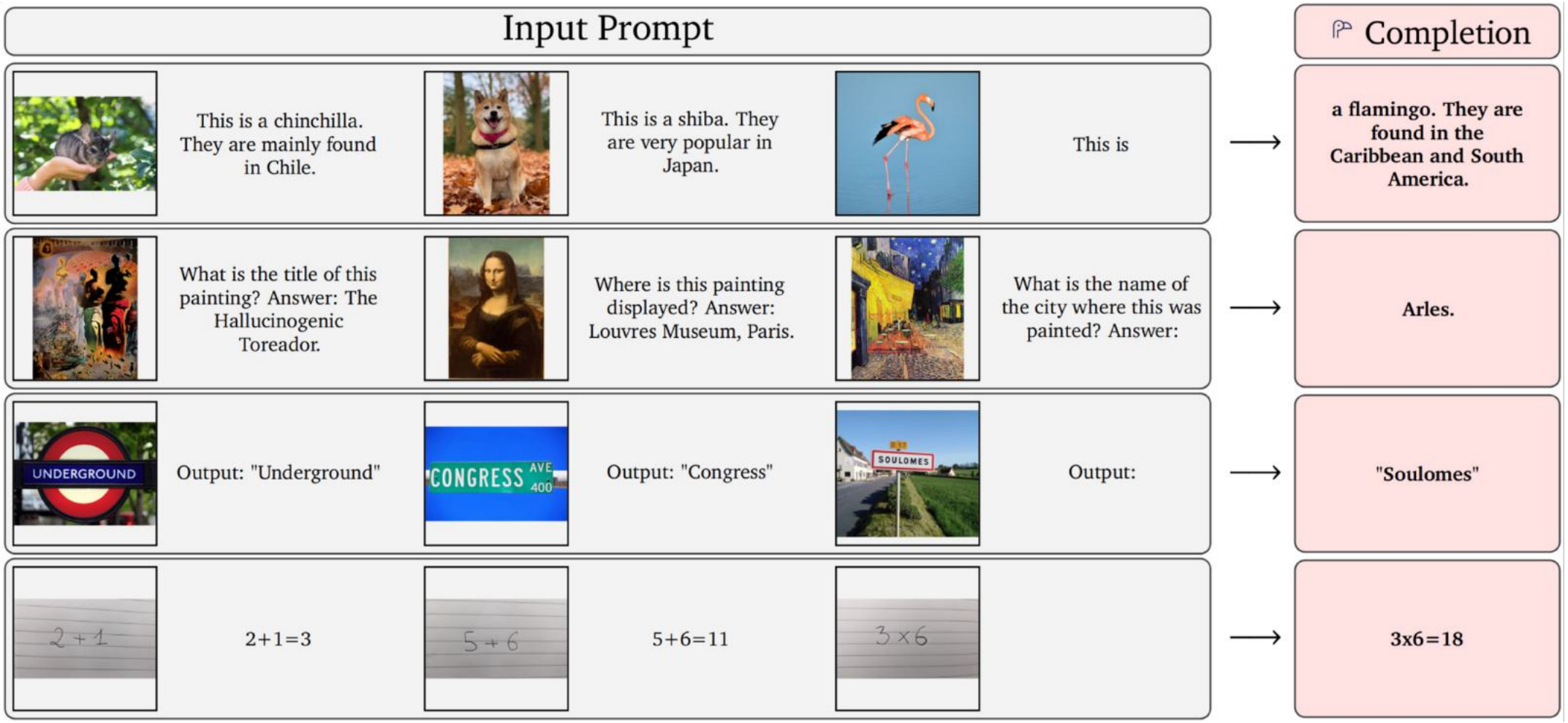






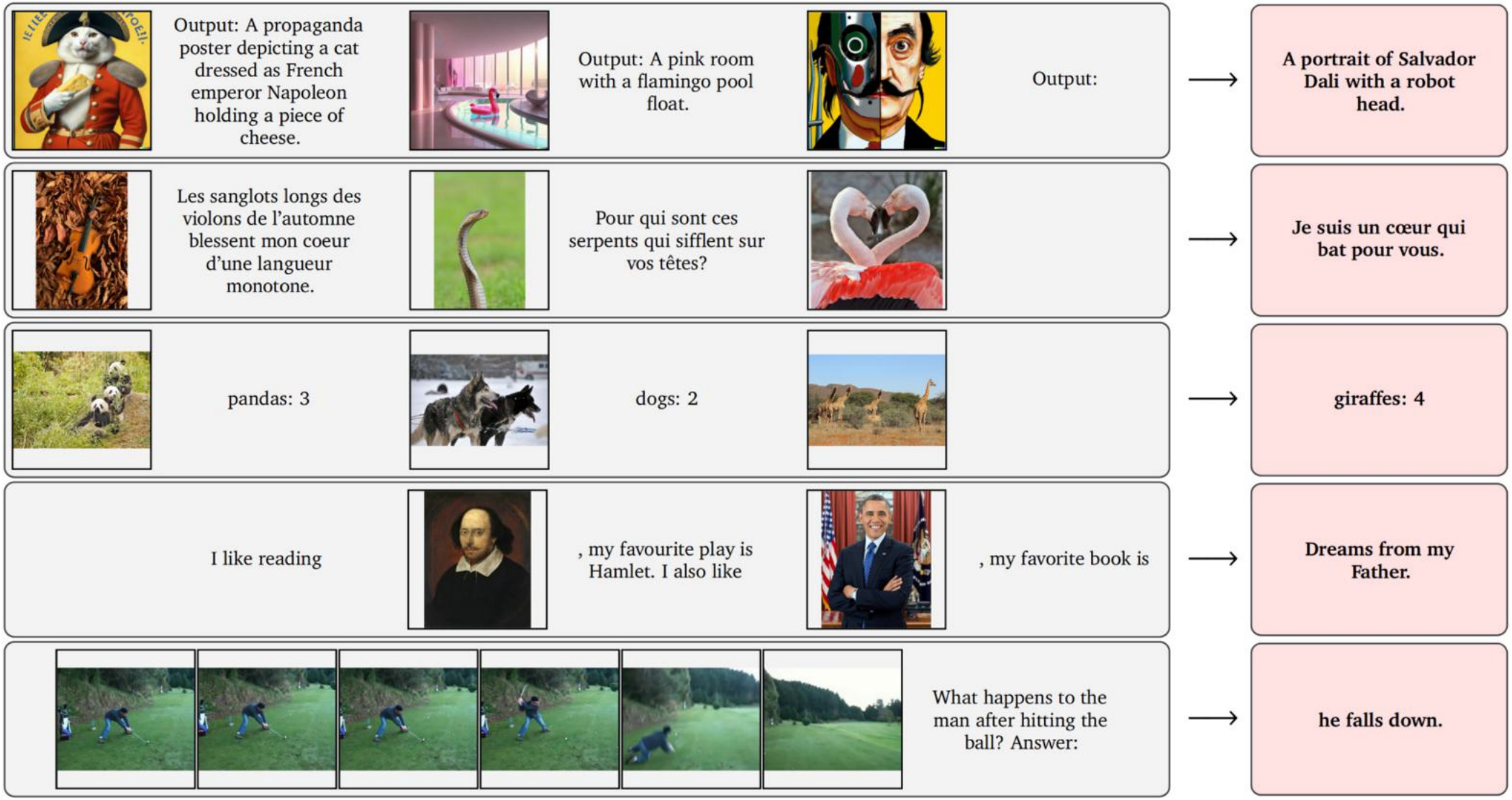
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo enables in-context learning



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

Flamingo results



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

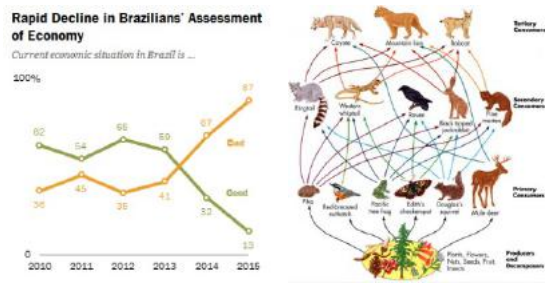
Flamingo results: Zero & few-shot

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)	-

Flamingo 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



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

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

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

Molmo

***Molmo and PixMo*:**
Open Weights and Open Data
for State-of-the-Art Vision-Language Models

Matt Deitke*†ψ Christopher Clark*† Sangho Lee† Rohun Tripathi† Yue Yang†
Jae Sung Park ψ Mohammadreza Salehiψ Niklas Muennighoff† Kyle Lo† Luca Soldaini†
Jiasen Lu† Taira Anderson† Erin Bransom† Kiana Ehsani† Huong Ngo†
YenSung Chen† Ajay Patel† Mark Yatskar† Chris Callison-Burch† Andrew Head†
Rose Hendrix† Favyen Bastani† Eli VanderBilt† Nathan Lambert† Yvonne Chou†
Arnavi Chheda† Jenna Sparks† Sam Skjonsberg† Michael Schmitz† Aaron Sarnat†
Byron Bischoff† Pete Walsh† Chris Newell† Piper Wolters† Tanmay Gupta† Kuo-Hao Zeng†
Jon Borchardt† Dirk Groeneveld† Crystal Nam† Sophie Lebrecht† Caitlin Wittlif†
Carissa Schoenick† Oscar Michel† Ranjay Krishna†ψ Luca Weihs†
Noah A. Smith†ψ Hannaneh Hajishirzi†ψ Ross Girshick†ψ Ali Farhadi†ψ Aniruddha Kembhavi†ψ

†Allen Institute for AI ψUniversity of Washington

Abstract

Today's most advanced vision-language models (VLMs) remain proprietary. The strongest open-weight models rely heavily on synthetic data from proprietary VLMs to achieve good performance, effectively distilling these closed VLMs into open ones. As a result, the community has been missing foundational knowledge about how to build performant VLMs from scratch. We present ***Molmo****, a new family of VLMs that are state-of-the-art in their class of openness. Our key contribution is a collection of new datasets called* ***PixMo****, including a dataset of highly detailed image captions for pre-training, a free-form image Q&A dataset for fine-tuning, and an innovative 2D pointing dataset, all collected without the use of external VLMs. The success of our approach relies on careful modeling choices, a well-tuned training pipeline, and, most critically, the quality of our newly collected datasets. Our best-in-class 72B model not only outperforms others in the class of open weight and data models, but also outperforms larger proprietary models including Claude 3.5 Sonnet, and Gemini 1.5 Pro and Flash, second only to GPT-4o based on both academic benchmarks and on a large human evaluation. Our model weights, new datasets, and source code are available at* https://molmo.allenai.org/blog.

1. Introduction

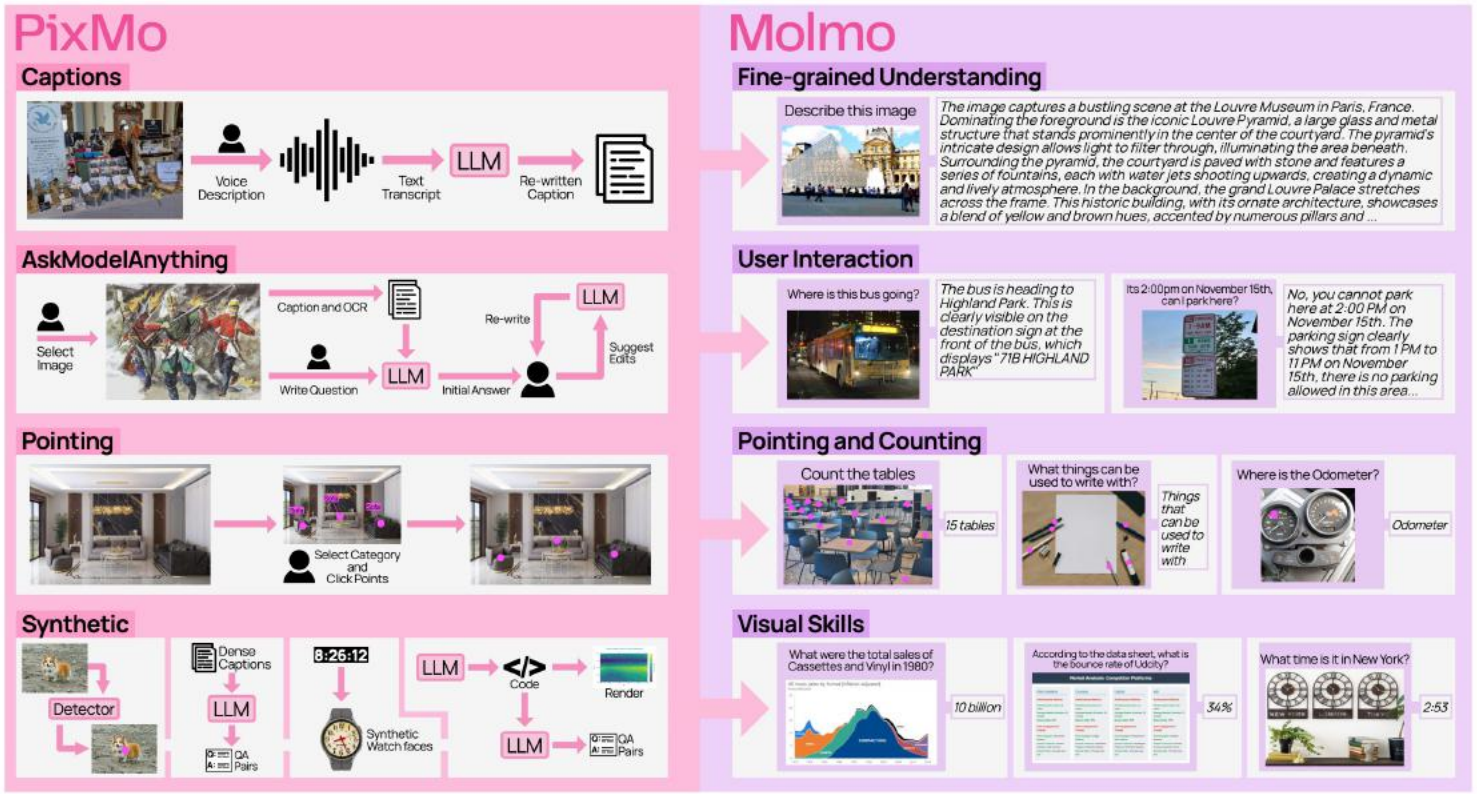
Large multimodal models are used ubiquitously today. Proprietary models—GPT-4o, Gemini-1.5 Pro, Claude 3.5 Sonnet—produce comprehensive image descriptions and accurately answer complex visual questions. Unfortunately, the most performant of these vision-language models (VLMs) remain proprietary with neither model weights, data, nor code being publicly released.

To foster scientific exploration, numerous research efforts have attempted to reproduce similar capabilities in *open* models. Early works, exemplified by LLaVA [69], produced fully open weights and training data but now lag significantly behind the state-of-the-art. More recent, stronger open-weight models have trended towards less open data: the training data may either be proprietary (*e.g.*, [10]) or, in cases where it is released, there is a heavy reliance on *synthetic* data generated by proprietary systems, *e.g.*, models are trained on datasets like ShareGPT4V [15] which uses GPT-4V [88] to generate a large set of detailed image captions. The resulting VLMs, therefore, are effectively *distillations* of proprietary VLMs. As it stands, the scientific community is missing foundational knowledge about how to build performant VLMs *from scratch* (more discussion about this and related work are in the Appendix).

In this work, we present the **Molmo** (Multimodal **O**pen **L**anguage **Mo**del) family of state-of-the-art open VLMs with released model weights *and* released vision-language training data without any reliance on synthetic data from other VLMs, including proprietary ones. The success of our approach relies on careful model design choices, a well-tuned training pipeline, and most critically, the quality of our new datasets, collectively named **PixMo** (**Pix**els for **Mo**lmo), which are fully open.

High-quality multimodal data, both for pre-training and

*Equal contribution



Molmo

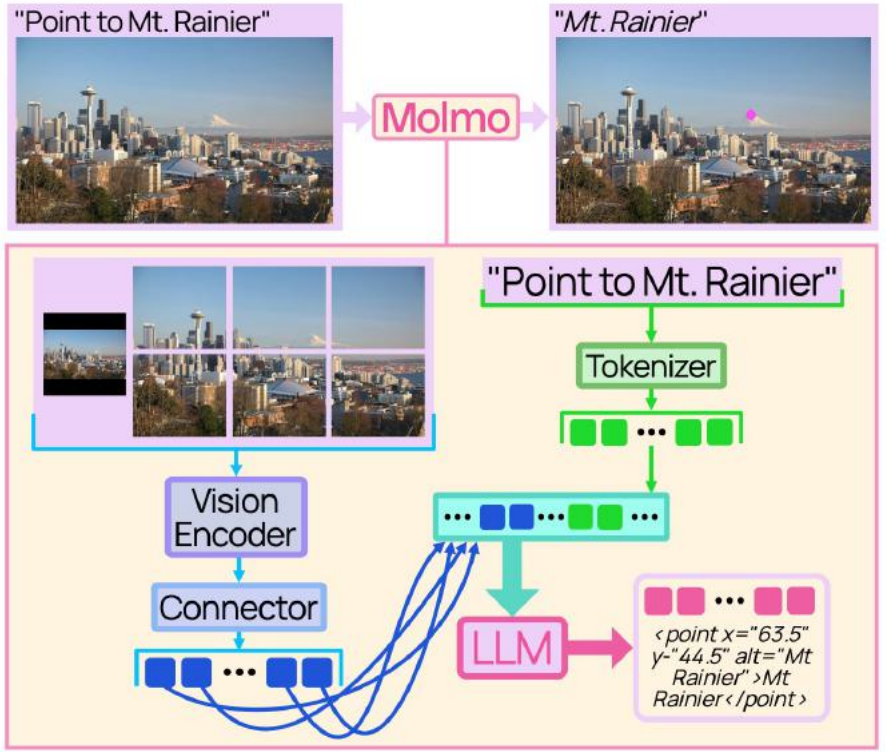


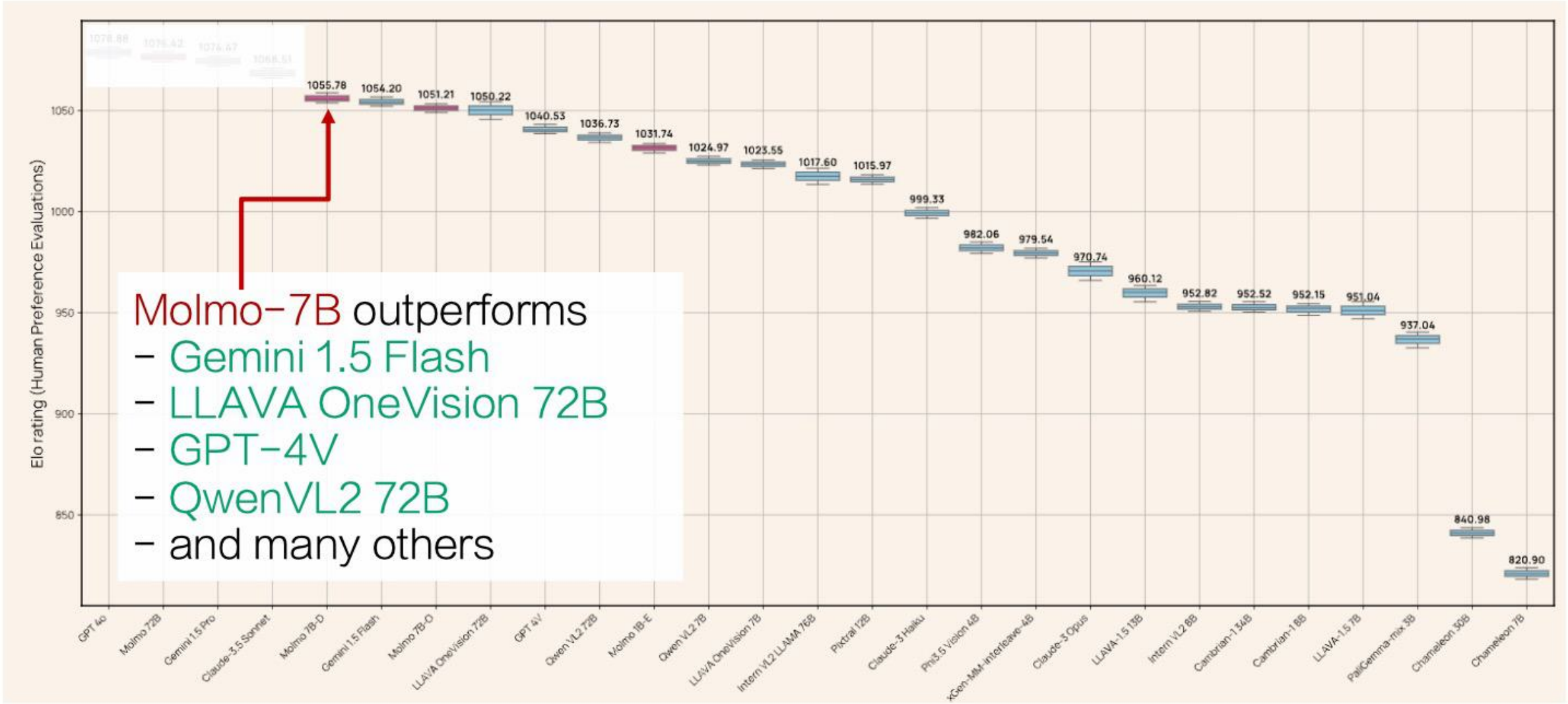
Figure 2. **Molmo** follows the simple and standard design of connecting a vision encoder and a language model.

Molmo results

model	AI2D test [49]	ChartQA test [82]	VQA v2.0 testdev [36]	DocVQA test [83]	InfoQA test [84]	TextVQA val [100]	RealWorldQA [116]	MMMU val [129]	MathVista testmini [78]	CountBenchQA [10]	PixMo-Count test	Average	Elo score	Elo rank
API call only														
GPT-4V [88]	89.4	78.1	77.2	87.2	75.1	78.0	61.4	63.1	58.1	69.9	45.0	71.1	1041	10
GPT-4o-0513 [90]	94.2	85.7	78.7	92.8	79.2	77.4	75.4	**69.1**	63.8	87.9	59.6	78.5	**1079**	**1**
Gemini 1.5 Flash [103]	91.7	85.4	80.1	89.9	75.3	78.7	67.5	56.1	58.4	81.6	61.1	75.1	1054	7
Gemini 1.5 Pro [103]	94.4	87.2	80.2	93.1	81.0	78.7	70.4	62.2	63.9	85.8	64.3	78.3	1074	3
Claude-3 Haiku [7]	86.7	81.7	68.4	88.8	56.1	67.3	45.5	50.2	46.4	83.0	43.9	65.3	999	18
Claude-3 Opus [7]	88.1	80.8	66.3	89.3	55.6	67.5	49.8	59.4	50.5	83.6	43.3	66.7	971	21
Claude-3.5 Sonnet [7]	94.7	**90.8**	70.7	95.2	74.3	74.1	60.1	68.3	**67.7**	89.7	58.3	76.7	1069	4
Open weights only														
PaliGemma-mix-3B [10]	72.3	33.7	76.3	31.3	21.4	56.0	55.2	34.9	28.7	80.6	60.0	50.0	937	27
Phi3.5-Vision-4B [1]	78.1	81.8	75.7	69.3	36.6	72.0	53.6	43.0	43.9	64.6	38.3	59.7	982	19
Qwen2-VL-7B [111]	83.0	83.0	82.9	94.5	76.5	84.3	70.1	54.1	58.2	76.5	48.0	73.7	1025	14
Qwen2-VL-72B [111]	88.1	88.3	81.9	**96.5**	84.5	**85.5**	**77.8**	64.5	70.5	80.4	55.7	79.4	1037	12
InternVL2-8B [104]	83.8	83.3	76.7	91.6	74.8	77.4	64.2	51.2	58.3	57.8	43.9	69.4	953	23
InternVL2-Llama-3-76B [104]	87.6	88.4	85.6	94.1	**82.0**	84.4	72.7	58.2	65.5	74.7	54.6	77.1	1018	16
Pixtral-12B [3]	79.0	81.8	80.2	90.7	50.8	75.7	65.4	52.5	58.0	78.8	51.7	69.5	1016	17
Llama-3.2V-11B-Instruct [5]	91.1	83.4	75.2	88.4	63.6	79.7	64.1	50.7	51.5	73.1	47.4	69.8	1040	11
Llama-3.2V-90B-Instruct [5]	92.3	85.5	78.1	90.1	67.2	82.3	69.8	60.3	57.3	78.5	58.5	74.5	1063	5
Open weights + data († distilled)														
LLaVA-1.5-7B [69]	55.5	17.8	78.5	28.1	25.8	58.2	54.8	35.7	25.6	40.1	27.6	40.7	951	26
LLaVA-1.5-13B [69]	61.1	18.2	80.0	30.3	29.4	61.3	55.3	37.0	27.7	47.1	35.2	43.9	960	22
xGen-MM-interleave-4B† [119]	74.2	60.0	81.5	61.4	31.5	71.0	61.2	41.1	40.5	81.9	50.2	59.5	979	20
Cambrian-1-8B† [106]	73.0	73.3	81.2	77.8	41.6	71.7	64.2	42.7	49.0	76.4	46.6	63.4	952	25
Cambrian-1-34B† [106]	79.7	75.6	83.8	75.5	46.0	76.7	67.8	49.7	53.2	75.6	50.7	66.8	953	24
LLaVA OneVision-7B† [59]	81.4	80.0	84.0	87.5	68.8	78.3	66.3	48.8	63.2	78.8	54.4	72.0	1024	15
LLaVA OneVision-72B† [59]	85.6	83.7	85.2	91.3	74.9	80.5	71.9	56.8	67.5	84.3	60.7	76.6	1051	8
The Molmo family: Open weights, Open data, Open training code, Open evaluations														
MolmoE-1B	86.4	78.0	83.9	77.7	53.9	78.8	60.4	34.9	34.0	87.2	79.6	68.6	1032	13
Molmo-7B-O	90.7	80.4	85.3	90.8	70.0	80.4	67.5	39.3	44.5	89.0	83.3	74.6	1051	9
Molmo-7B-D	93.2	84.1	85.6	92.2	72.6	81.7	70.7	45.3	51.6	88.5	84.8	77.3	1056	6
Molmo-72B	**96.3**	87.3	**86.5**	93.5	81.9	83.1	75.2	54.1	58.6	**91.2**	**85.2**	**81.2**	1077	2

Table 1. We present academic benchmark results for 10 common datasets, plus a new counting benchmark, PixMo-Count, which features more challenging natural images than CountBenchQA. We categorize models into four groups: (top) proprietary models accessible only via API calls, (upper middle) models with released weights but closed data, (lower middle) models with released weights and training data (noting some of these use distillation (†) from proprietary VLMs via synthetic data), and (bottom) the Molmo family of models.

Molmo results



with 325k pairwise comparisons and 870 human annotators

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.

A	B	C	D	E	F	G
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

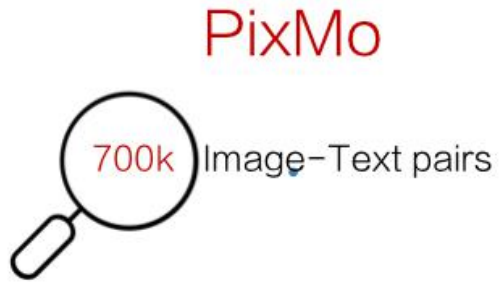
11:42 AM · Sep 25, 2024 · **45.6K** Views

Molmo grounds reasoning directly in the pixels

Example, it points when it counts

Data matters! Quality over quantity even for pretraining

Molmo is trained with



Internet data is **incidental**
Human annotated data is **intentional**

pink, japan,
aesthetic image

love this winter picture by
person

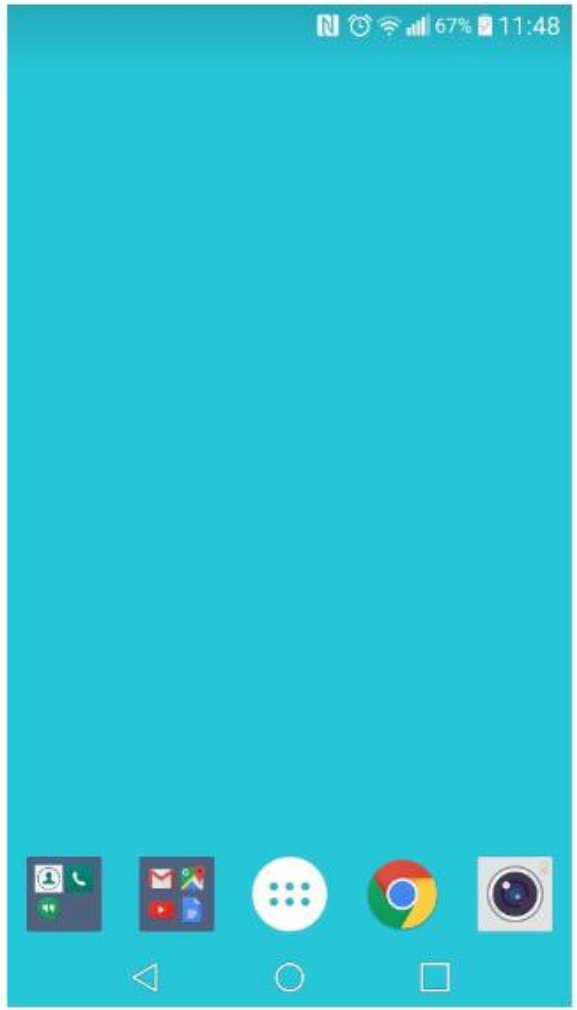
PixMo data is intentional:

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.

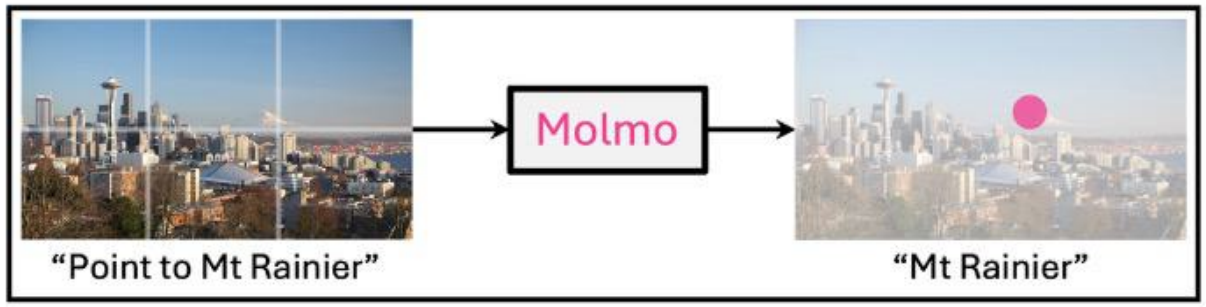
Questions designed to extract meaningful visual information from annotators

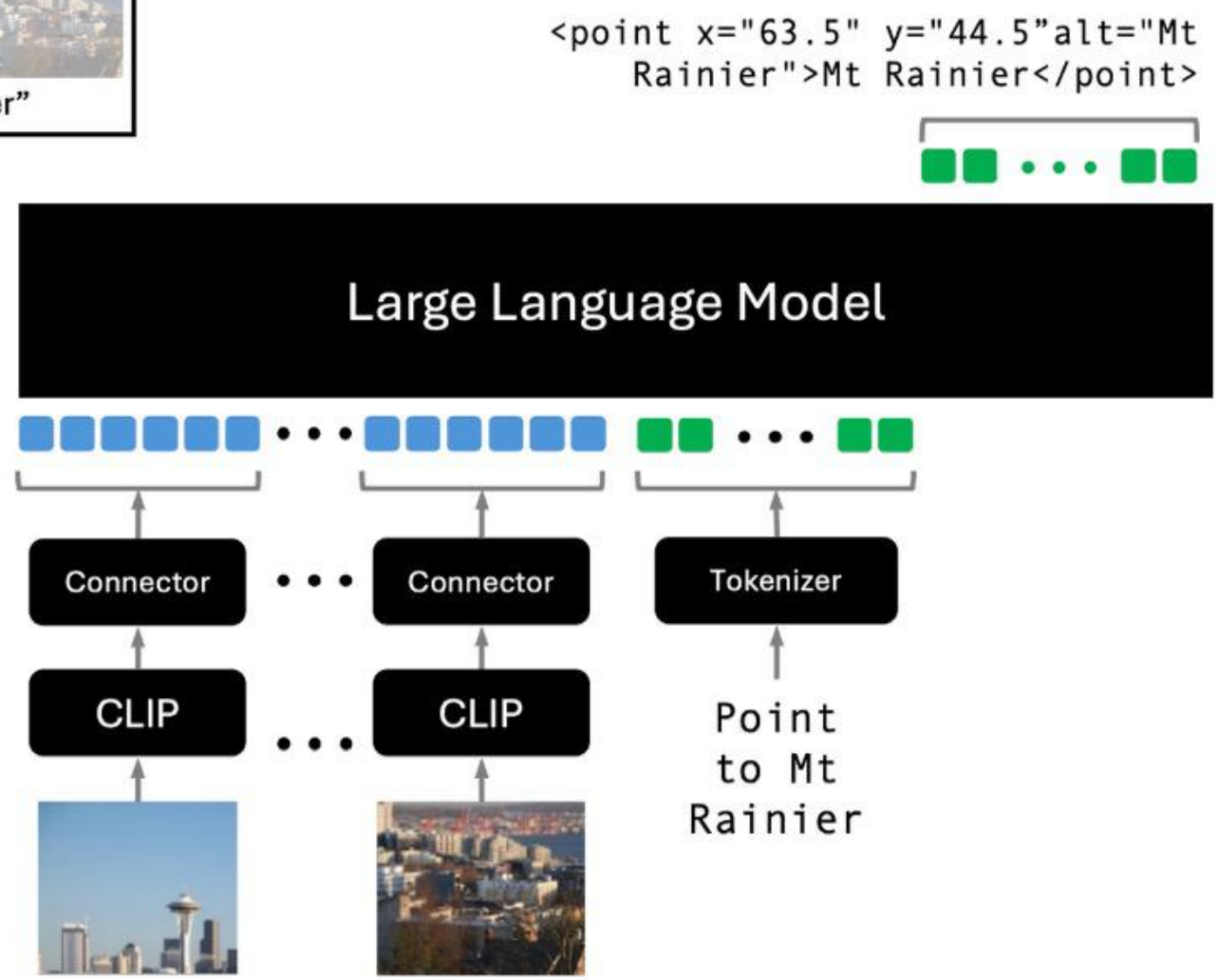
- What is the image at first glance?
- What are the objects and their counts?
- What does the text say?
- What are the positions of the objects?
- What subtle details are noticeable?
- What is in the background?
- What is the style and color?

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

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

They automatically convert speech into text for pretraining





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

Molmo2

Open Weights and Data for Vision-Language Models with Video Understanding and Grounding

Christopher Clark[♥1*] Jieyu Zhang[♥1,2*] Zixian Ma[♥1,2*] Jae Sung Park[♥1,2] Mohammadreza Salehi[♥1,2] Rohun Tripathi[♥1] Sangho Lee[♥1]

Zhongzheng Ren[1,2] Chris Dongjoo Kim[1] Yinuo Yang[2] Vincent Shao[2] Yue Yang[1] Weikai Huang[2] Ziqi Gao[1] Taira Anderson[1] Jianrui Zhang[1] Jitesh Jain[1] George Stoica[1] Winson Han[1]

Ali Farhadi[1,2] Ranjay Krishna[♥1,2]

[1]Allen Institute for AI, [2]University of Washington

[*]denotes equal contribution. ♥ marks core contributors, who were all integral to the project
See full author contributions here.

Models: Molmo2-4B Molmo2-8B Molmo2-O-7B
Data: Molmo2 Data
Code: https://github.com/allenai/molmo2
Demo: playground.allenai.org
Contact: molmo@allenai.org

Abstract

Today's strongest video-language models (VLMs) remain proprietary. The strongest open-weight models either rely on synthetic data from proprietary VLMs, effectively distilling from them, or do not disclose their training data or recipe. As a result, the open-source community lacks the foundations needed to improve on the state-of-the-art video (and image) language models. Crucially, many downstream applications require more than just high-level video understanding; they require grounding—either by pointing or by tracking in pixels. Even proprietary models lack this capability. We present Molmo2, a new family of VLMs that are state-of-the-art among open-source models and demonstrate exceptional new capabilities in point-driven grounding in single image, multi-image, and video tasks. Our key contribution is a collection of 7 new video datasets and 2 multi-image datasets, including a dataset of highly detailed video captions for pre-training, a free-form video Q&A dataset for fine-tuning, a new object tracking dataset with complex queries, and an innovative new video pointing dataset, all collected without the use of closed VLMs. We also present a training recipe for this data utilizing an efficient packing and message-tree encoding scheme, and show bi-directional attention on vision tokens and a novel token-weight strategy improves performance. Our best-in-class 8B model outperforms others in the class of open weight and data models on short videos, counting, and captioning, and is competitive on long-videos. On video-grounding Molmo2 significantly outperforms existing open-weight models like Qwen3-VL (35.5 vs 29.6 accuracy on video counting) and surpasses proprietary models like Gemini 3 Pro on some tasks (38.4 vs 20.0 F1 on video pointing and 56.2 vs 41.1 $\mathcal{J}\&\mathcal{F}$ on video tracking).

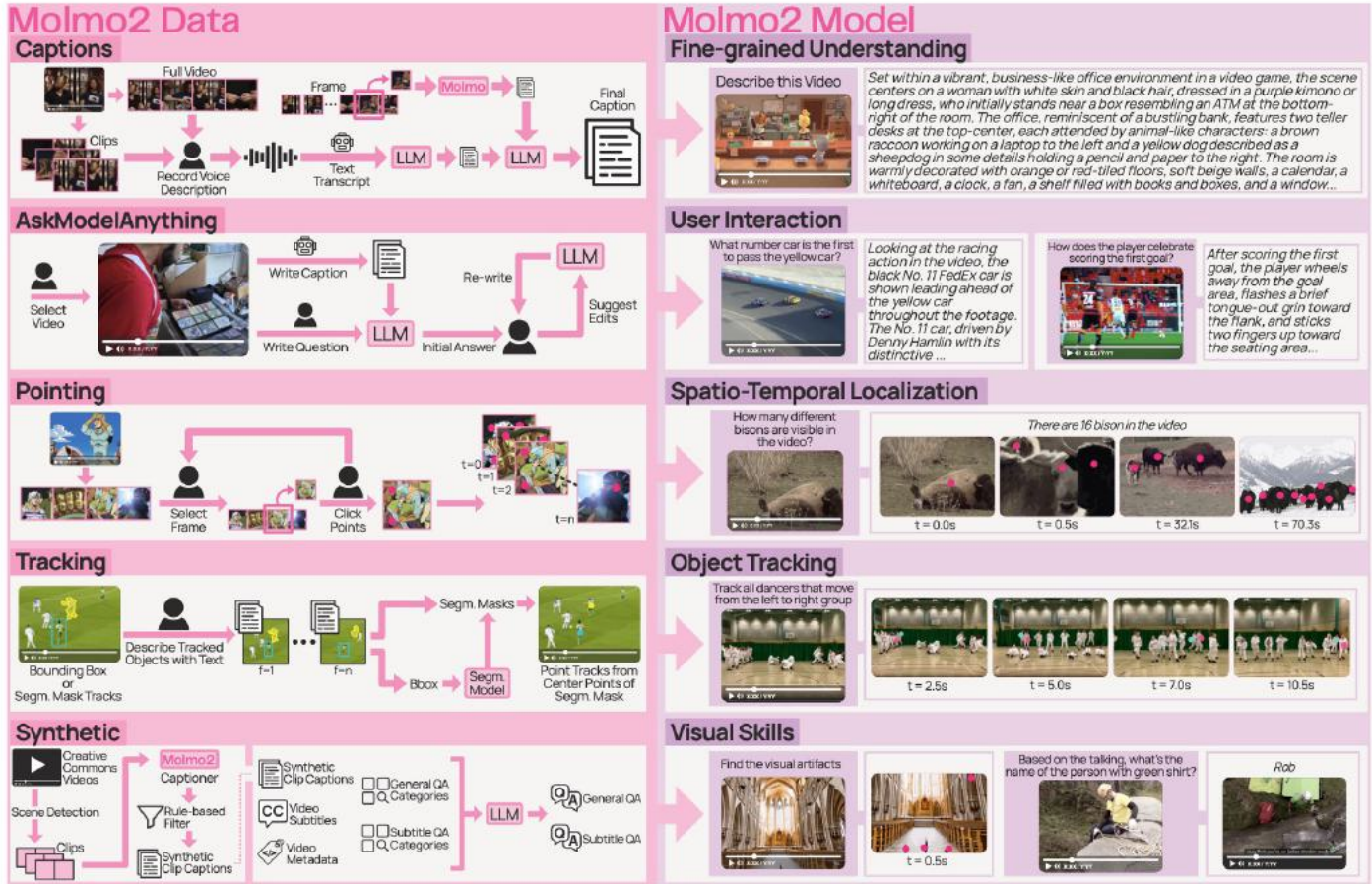


Figure 1 Molmo2 is trained on one of the largest fully open video-centric multimodal corpus to date, including nine new datasets for dense video captioning, long-form and long-video QA, and open-vocabulary pointing and tracking over images, multi-images, and videos. Molmo2 accepts single images, image sets, and videos as input and can produce both free-form language and grounded outputs such as spatio-temporal points, object tracks, and grounded chain-of-thoughts that localize objects and events over time. Across diverse video-language and grounding benchmarks, Molmo2 matches or surpasses prior open models, approaches proprietary systems, and remains fully open.

Molmo2 training

Dataset Group	Description	Rate(%)	Datasets	Examples
Captions/Long QA	Captioning and long-form question answering data on images and videos, including Molmo2-Cap, -AskModelAnything, -MultiImageQA and PixMo-Cap, -AskModelAnything and -CapQA.	13.6	6	1.2m
Image QA	Multiple-choice and short answer image QA data, including Molmo2-SynMultiImageQA, open-source image datasets [48, 130, 101, 102, 103, 105, 65, 125, 94, 95, 14, 2, 61, 62, 107] following Molmo with CoSyn [172] instead of PixMo-Docs, and open-source multi-image datasets [132, 89, 58].	22.7	32	2.4m
Video QA	Multiple-choice and short answer video QA, including Molmo2-CapQA, -SubtitleQA, and various open video datasets [80, 74, 154, 184, 115, 160, 57, 167, 163, 174, 155, 156, 138, 38, 86, 54, 46, 109, 64, 42, 134, 188, 15, 49, 26, 141, 75, 77, 161, 123, 159]. Downsampled since video-benchmarks converge quickly.	18.2	32	2.4m
Image Pointing	PixMo-Points and PixMo-Count, CoSyn-Point [172], and Molmo2-MultiImagePoint. PixMo-Points is weighted to emphasize high counts. Downsampled since it was seen during pre-training.	9.1	4	1.1m
Video Pointing	Molmo2-VideoPoint and AcademicVideoPoint. Upsampled since this task is slow to converge.	13.6	7	0.37m
Video Tracking	Molmo2-VideoTrack and AcademicVideoTrack. Re-weighted to emphasize tail concepts.	13.6	22	0.80m
NLP	Text-only SFT data from Tulu [71] to preserve performance on natural language understanding.	9.1	1	0.99m

Table 1 We create nine new datasets (in pink) to train Molmo2. We also include a suite of image and language data from academic datasets into our training mix. We categorize all datasets into categories and show each categories' sampling rate, dataset count, and total training examples after filtering and formatting the data into message trees. See Section 2 and the appendix for details.

Molmo2 Architecture

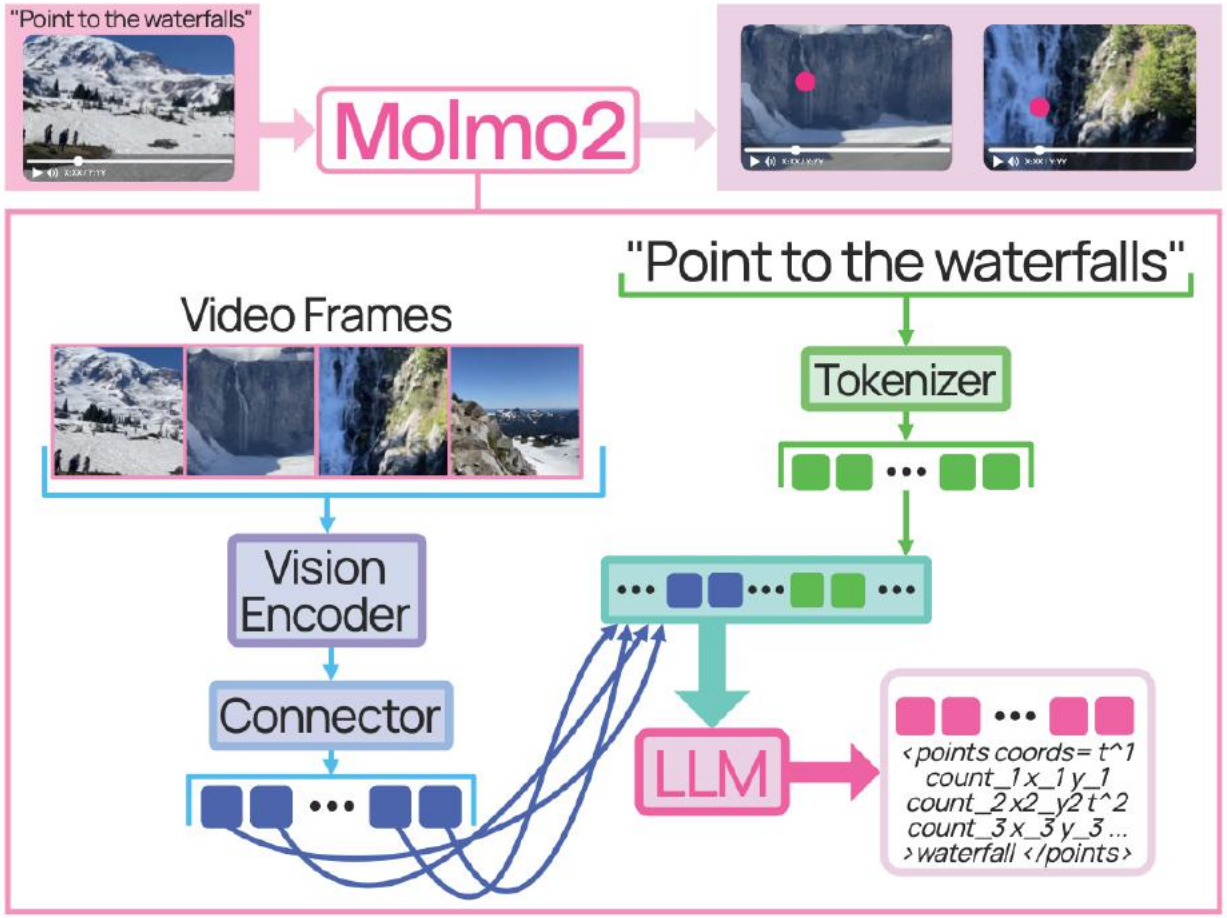


Figure 2 Molmo2 follows the standard design of connecting a vision encoder and a language model to process video inputs.

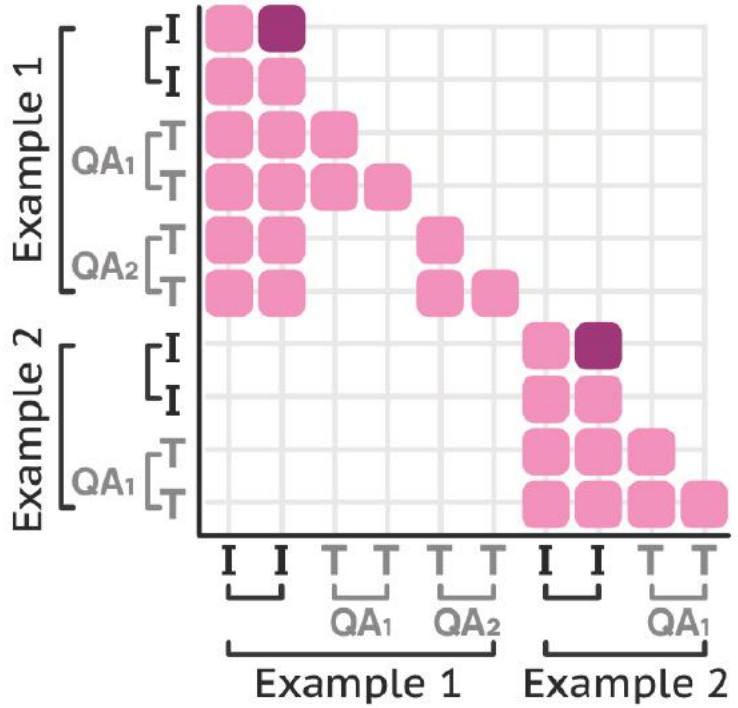


Figure 3 Attention mask for a *packed* sequence with two examples. The first contains two QA pairs for one image. Frame tokens (dark pink) have forward attention, while masking blocks cross-attention between different examples (lower-left empty block) and between distinct QA pairs within the same example (upper empty block).

Molmo2 results

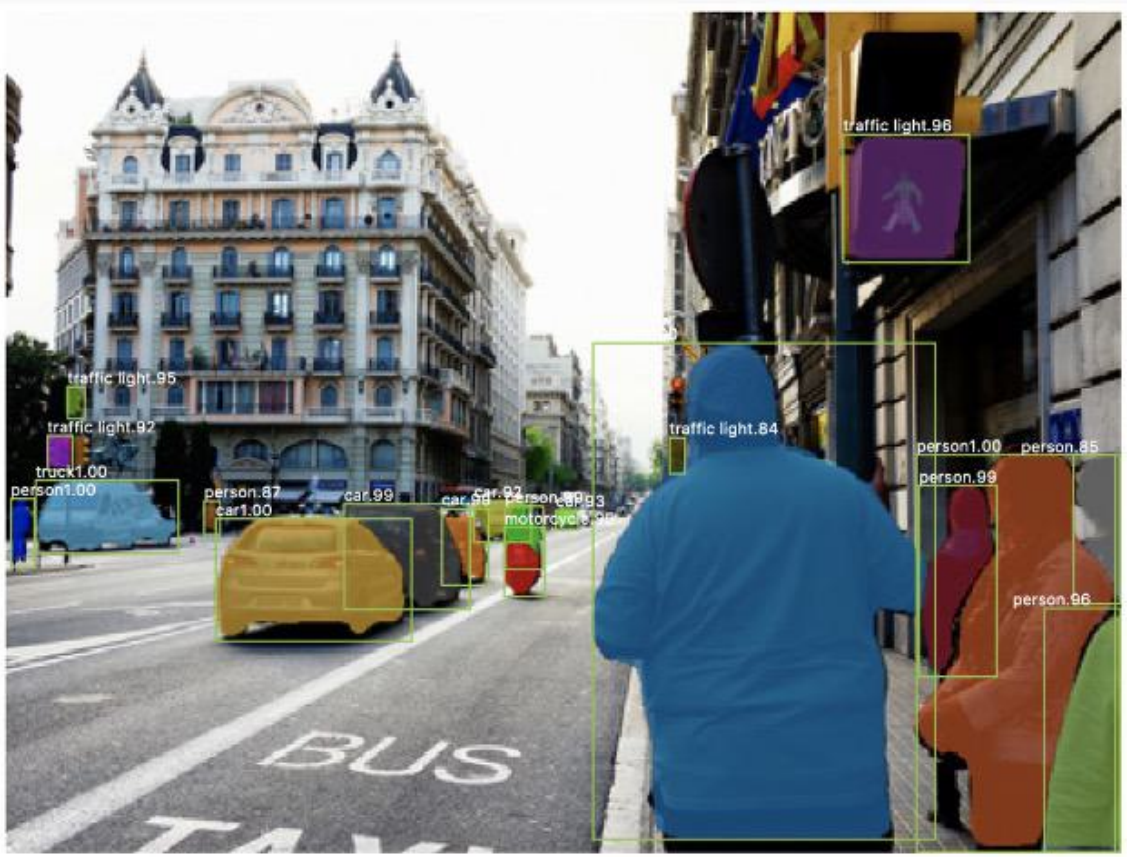
Model	NextQA test [160]	PerceptionTest test [115]	MVBench test [78]	Tomato test [128]	MotionBench val [54]	TempCompass test MCQ [91]	Video-MME test [40]	Video-MME-Sub test [40]	LongVideoBench val [157]	MLVU test MCQ [187]	LVBench test [150]	VideoEvalPro test [98]	Ego Schema test [100]	Molmo2 Caption test F1 Score	Molmo2 Count val accuracy	Short QA avg.	Long QA avg.	Average	Elo Score	Elo Rank
API call only																				
GPT-5 [114]	86.3	79.4	74.1	53.0	65.4	80.4	83.3	86.9	72.6	77.7	65.2	68.8	75.6	50.1	35.8	73.1	76.3	70.6	1031	10
GPT-5 mini [114]	83.2	72.0	66.5	44.1	59.9	74.9	77.3	82.3	69.7	69.1	54.7	60.1	70.9	56.6	29.8	66.8	69.8	65.0	1076	4
Gemini 3 Pro [45]	84.3	77.6	70.4	48.3	62.6	82.8	88.6	87.5	75.9	75.7	77.0	78.0	68.9	36.0	37.1	71.0	78.8	70.0	1082	3
Gemini 2.5 Pro [25]	85.3	78.4	70.6	48.6	62.0	81.9	87.8	87.8	76.8	81.5	75.7	78.4	72.2	42.1	35.8	71.1	80.4	71.2	1096	1
Gemini 2.5 Flash [25]	81.8	74.7	67.0	39.1	59.3	80.2	84.2	84.2	73.1	75.1	64.9	69.6	70.2	46.0	31.9	67.0	74.5	66.7	1084	2
Claude Sonnet 4.5 [5]	79.2	64.3	62.1	39.6	58.5	72.8	74.2	80.5	65.1	64.0	50.5	50.5	73.1	26.0	27.2	62.8	66.4	59.6	1008	12
Open weights only																				
InternVL3.5-4B [149]	80.3	68.1	71.2	26.8	56.5	68.8	65.4	68.6	60.8	52.0	43.2	46.5	58.9	7.7	26.3	62.0	56.5	53.4	935	18
InternVL3.5-8B [149]	81.7	72.7	72.1	24.6	56.6	70.3	66.0	68.6	62.1	53.2	43.4	48.1	58.6	7.8	26.1	63.0	57.1	54.1	941	19
Qwen3-VL-4B [169]	81.4	70.7	68.9	31.8	58.6	70.8	69.3	74.0	62.8	58.4	56.2	49.8	68.4	25.2	25.3	63.7	62.7	58.1	1048	7
Qwen3-VL-8B [169]	83.4	72.7	68.7	35.7	56.9	74.3	71.4	75.2	62.4	57.6	**58.0**	50.3	69.8	26.7	29.6	65.3	63.5	59.5	1054	6
Keye-VL-1.5-8B [170]	75.8	64.2	56.9	33.0	55.1	**75.5**	**73.0**	**76.2**	66.0	53.8	42.8	54.9	56.3	25.4	27.2	60.1	60.4	55.7	952	17
GLM-4.1V-9B [137]	81.3	74.2	68.4	30.0	59.0	72.3	68.2	75.6	65.7	56.6	44.0	51.1	62.6	18.4	26.6	64.2	60.5	56.9	962	14
MiniCPM-V-4.5-8B [173]	78.8	70.9	60.5	29.8	59.7	72.7	67.9	73.5	63.9	60.6	50.4	54.9	49.6	29.3	26.3	62.1	60.1	56.6	975	13
Eagle2.5-8B [17]	85.0	81.0	74.8	31.0	55.7	74.4	72.4	75.7	66.4	60.4	50.9	58.6	**72.2**	22.8	28.9	67.0	**65.2**	60.7	1019	11
Open models																				
PLM-3B [22]	83.4	79.3	74.7	30.9	60.4	69.3	54.9	59.4	57.9	48.4	40.4	46.2	66.9	12.3	24.4	66.3	53.5	53.9	841	20
PLM-8B [22]	84.1	**82.7**	**77.1**	33.2	61.4	72.7	58.3	65.4	56.9	52.6	44.5	47.2	68.8	10.9	26.6	68.5	56.2	56.2	853	21
LLaVA-Video-7B [184]	83.2	68.8	58.6	24.9	54.2	66.6	63.3	69.7	58.2	52.8	44.2	47.8	57.3	19.9	21.4	59.4	56.2	52.7	959	15
VideoChat-Flash-7B [79]	85.5	76.5	74.0	32.5	60.6	69.4	65.3	69.7	64.7	56.0	48.2	51.2	51.3	14.8	21.6	66.4	58.1	56.1	956	16
Molmo2 family: Open weights, Open data (no distillation), Open code																				
Molmo2-4B	85.5	81.3	75.1	**39.8**	61.6	72.8	69.6	75.7	**68.0**	**63.0**	53.9	59.9	61.2	39.9	34.3	69.3	64.5	62.8	1041	8
Molmo2-8B	**86.2**	82.1	75.9	39.6	**62.2**	73.4	69.9	75.8	67.5	60.2	52.8	**60.4**	62.0	**43.2**	**35.5**	**69.9**	64.1	**63.1**	**1057**	**5**
Molmo2-O-7B	84.3	79.6	74.8	36.2	60.6	73.0	64.9	69.2	63.7	55.2	49.6	55.1	56.8	40.1	33.2	68.1	59.2	59.7	1033	9

Table 2 Video benchmark results for a range of proprietary APIs, open-weight baselines, video-specialized models, and our Molmo2 family across video understanding, captioning, and counting benchmarks. The result of the best-performing open-weight model is in **bold**, and the second best is underlined.

Segment Anything

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

Segment Anything Model (SAM)

- What does it mean to have a segmentation foundation model?

Masking model trained on dataset of a huge number of categories

Model outputs masks of any objects in that the user cares about

Segment Anything Model (SAM)

- What does it mean to have a segmentation foundation model?

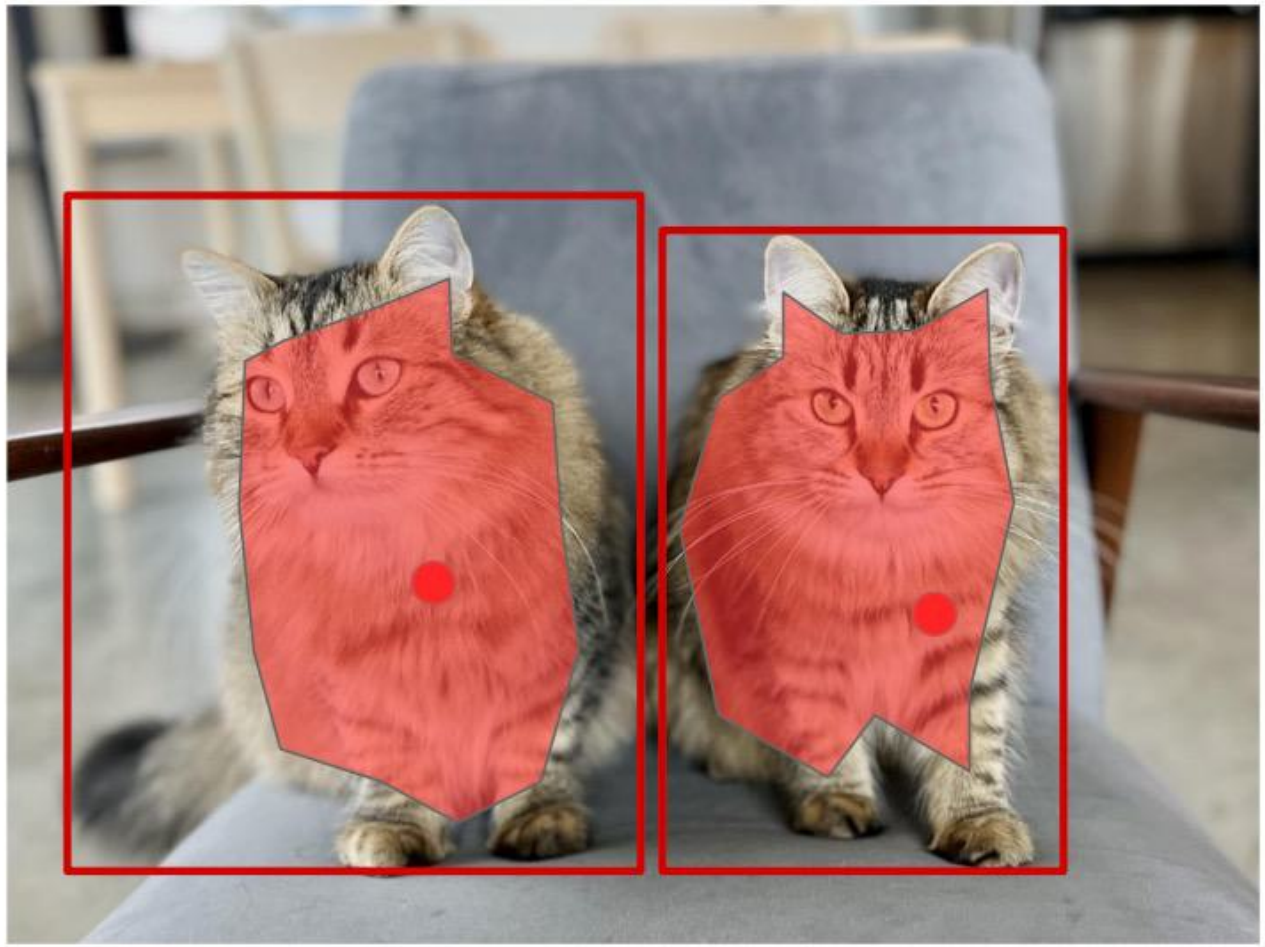
Masking model trained on dataset of a huge number of categories

How to get this?

Model outputs masks of any objects in that the user cares about

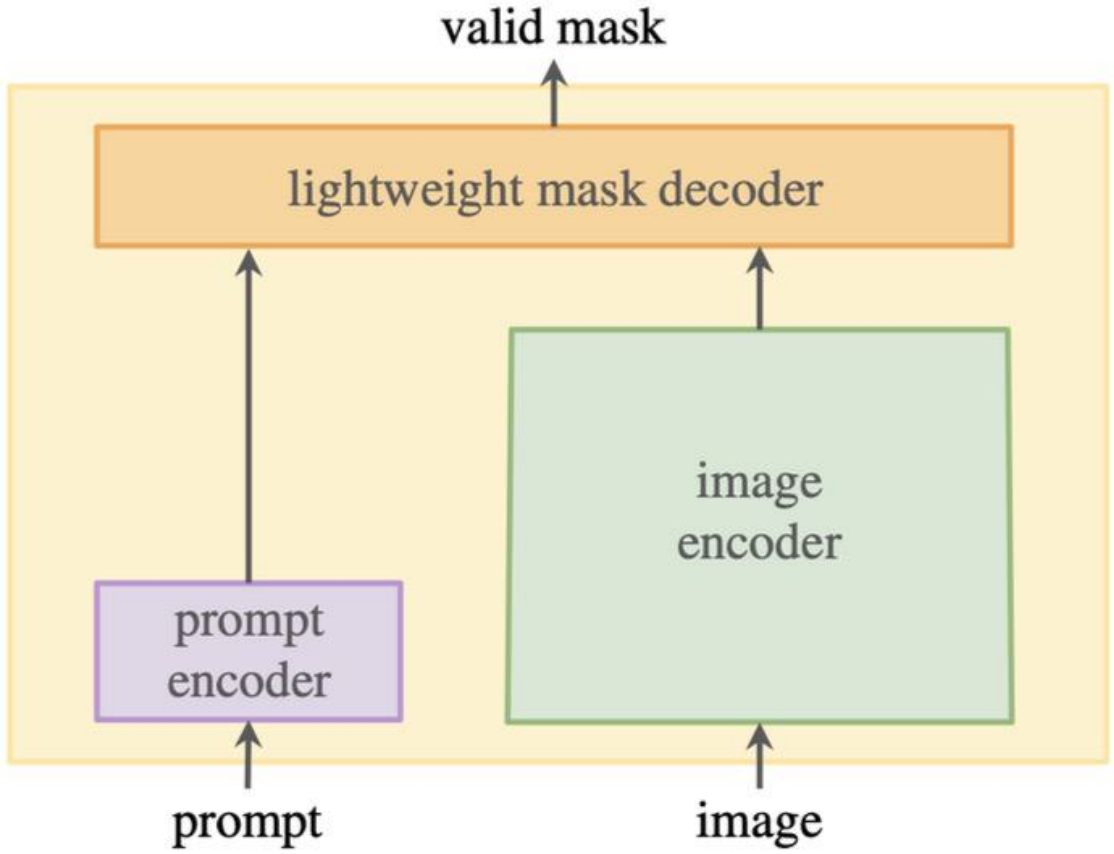
How to know this?

How to know what to mask?



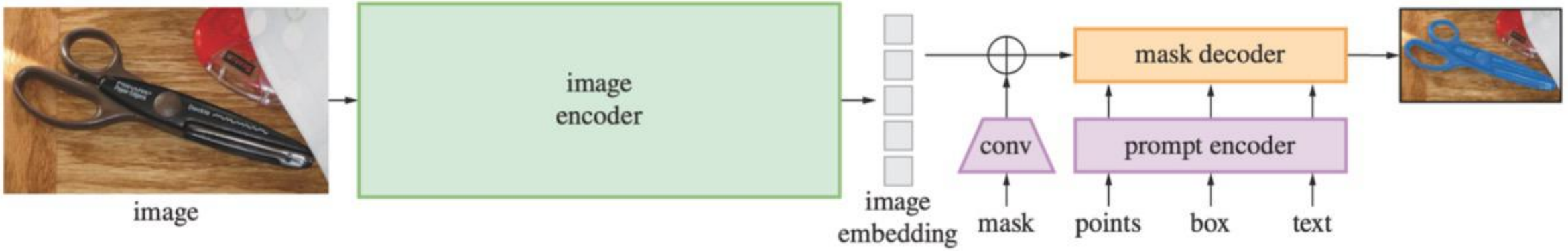
"Cats"

Basic SAM Architecture



Images: Kirillov et al. Segment Anything. 2023

SAM Architecture

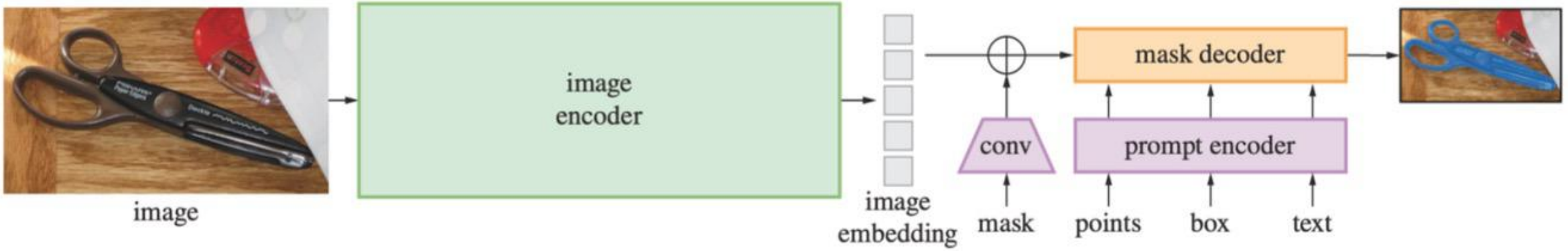


Images: Kirillov et al. Segment Anything. 2023

Ambiguity in correct prompt

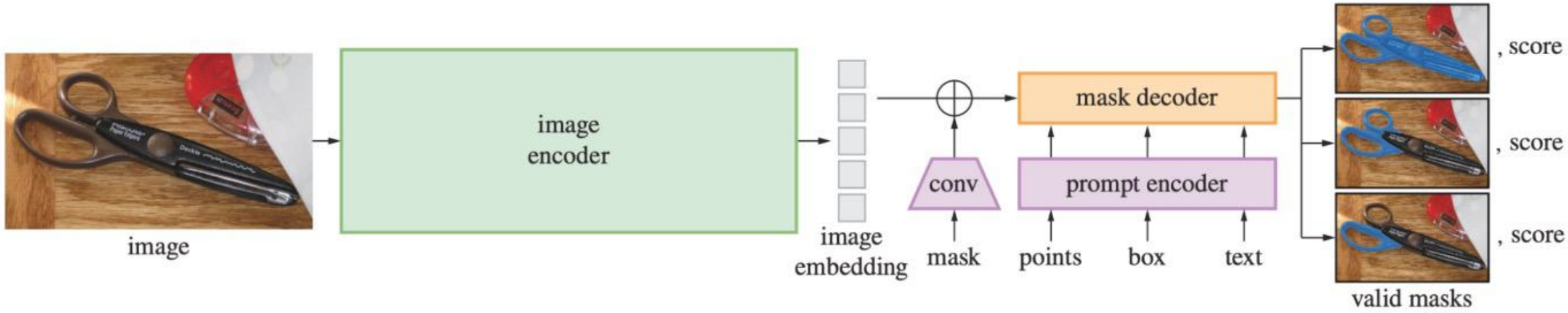
Ambiguity in correct prompt

SAM Architecture



Images: Kirillov et al. Segment Anything. 2023

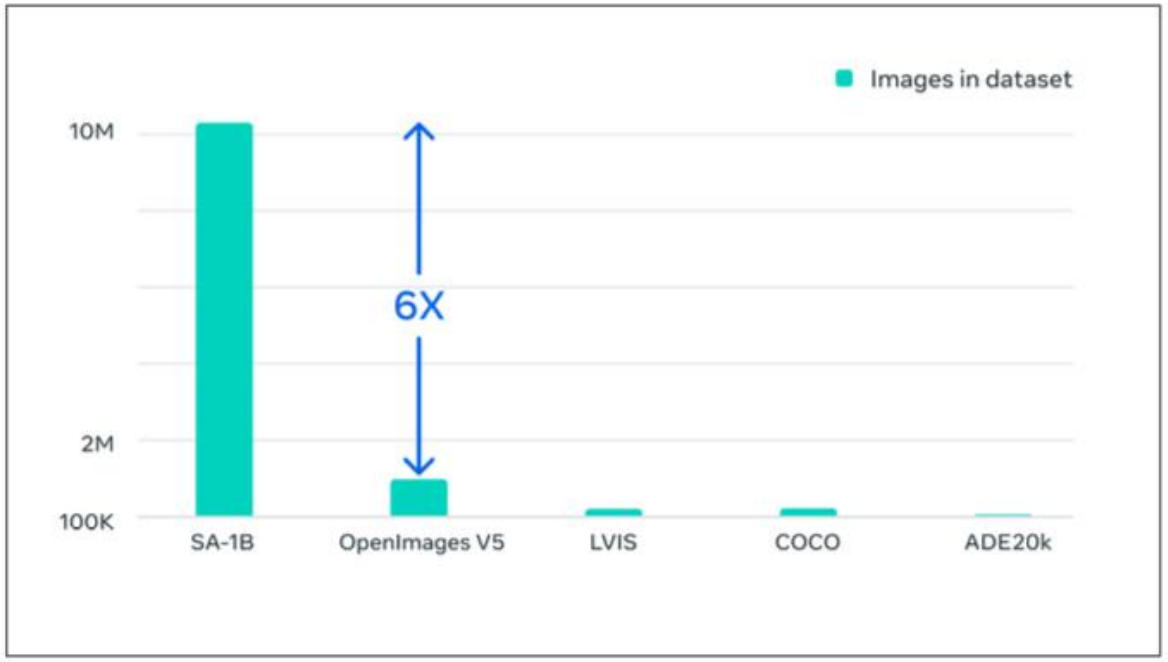
Basic SAM Architecture

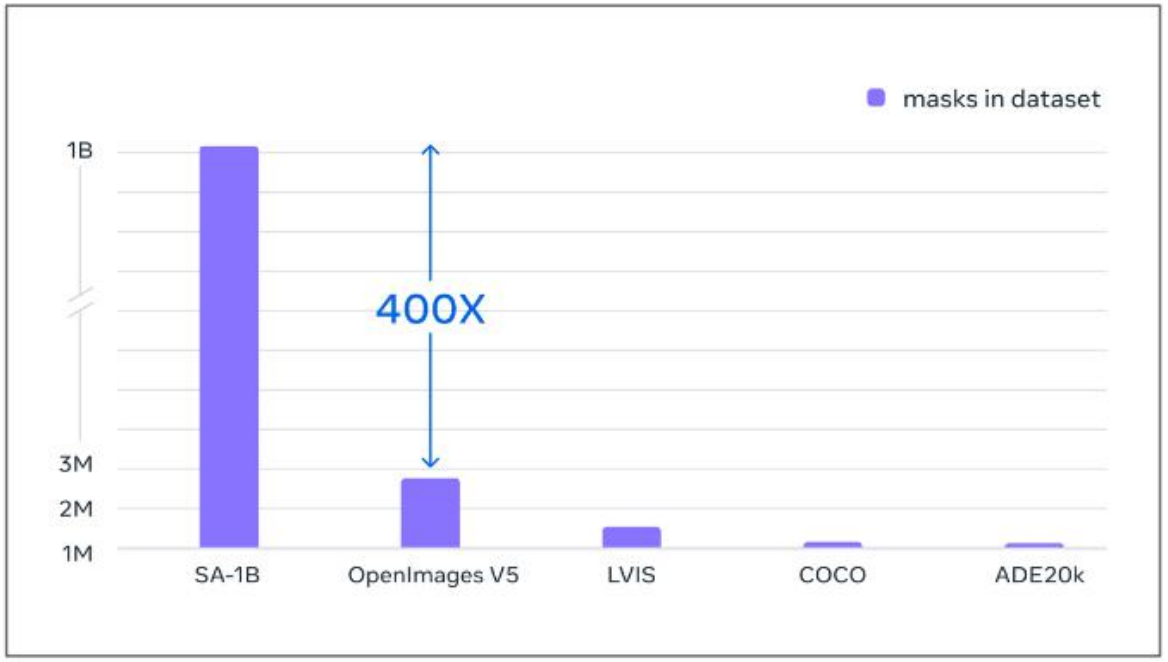


1. Loss only calculated with respect to best mask
2. Model also trained to output confidence score for each mask

Images: Kirillov et al. Segment Anything. 2023

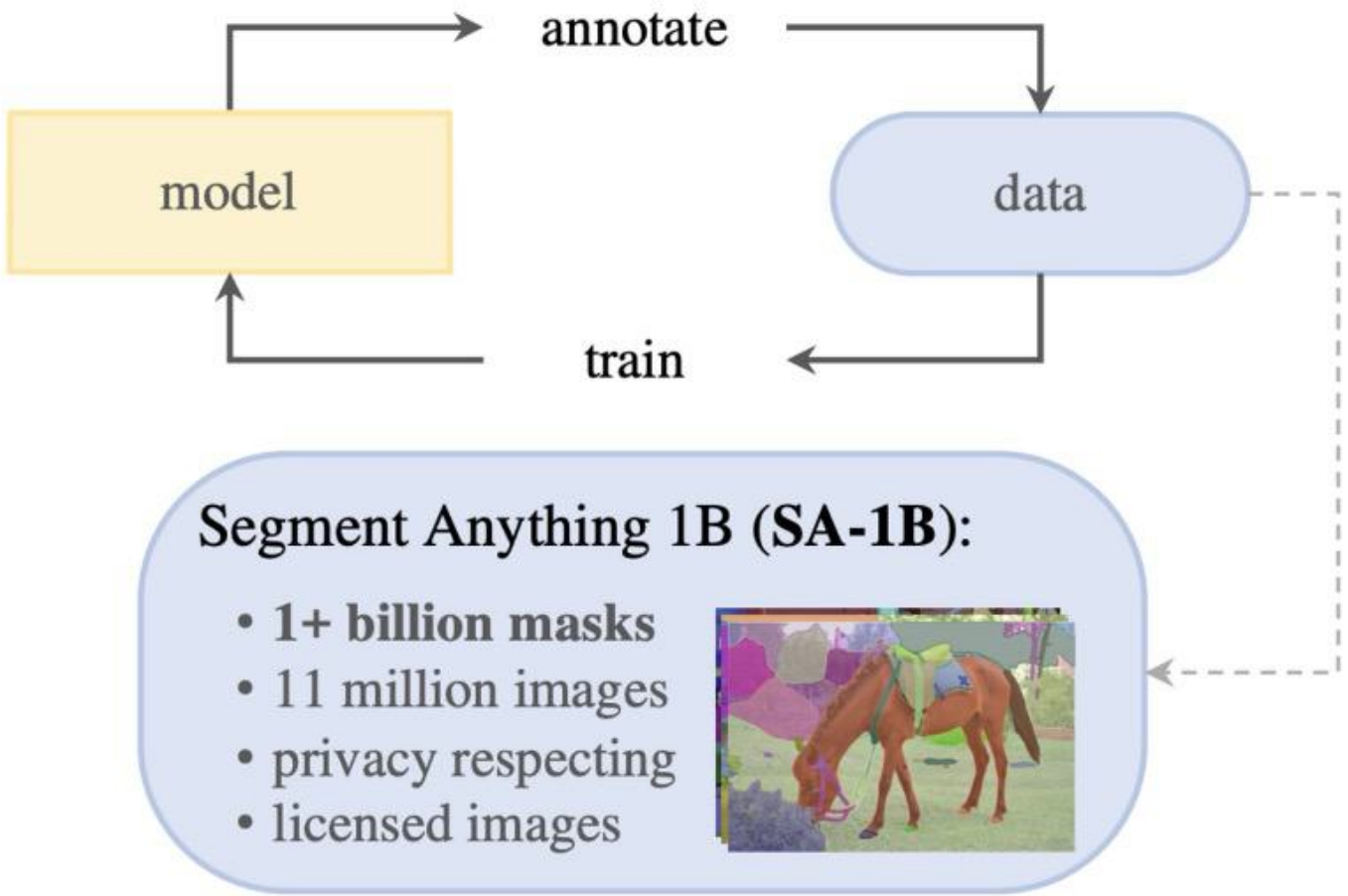
Segment Anything Model (SAM)





Images: Kirillov et al. Segment Anything. 2023

Segment Anything Model (SAM)



Images: Kirillov et al. Segment Anything. 2023

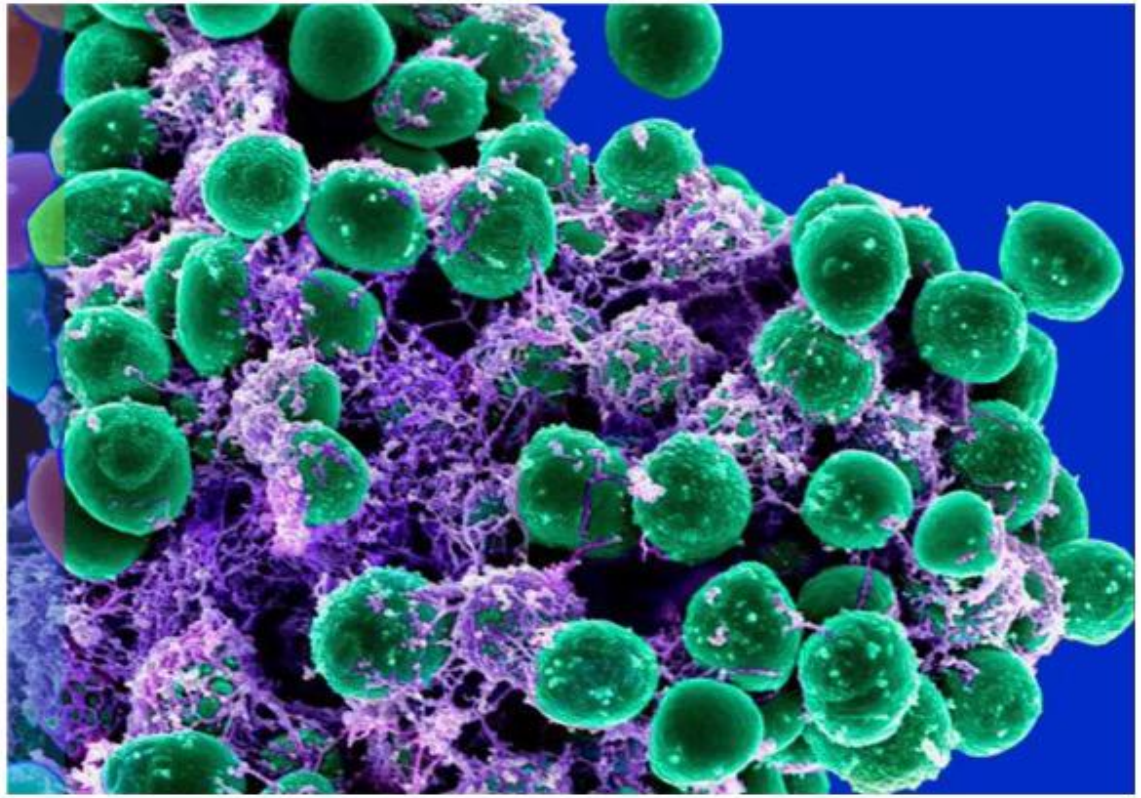
SAM Results

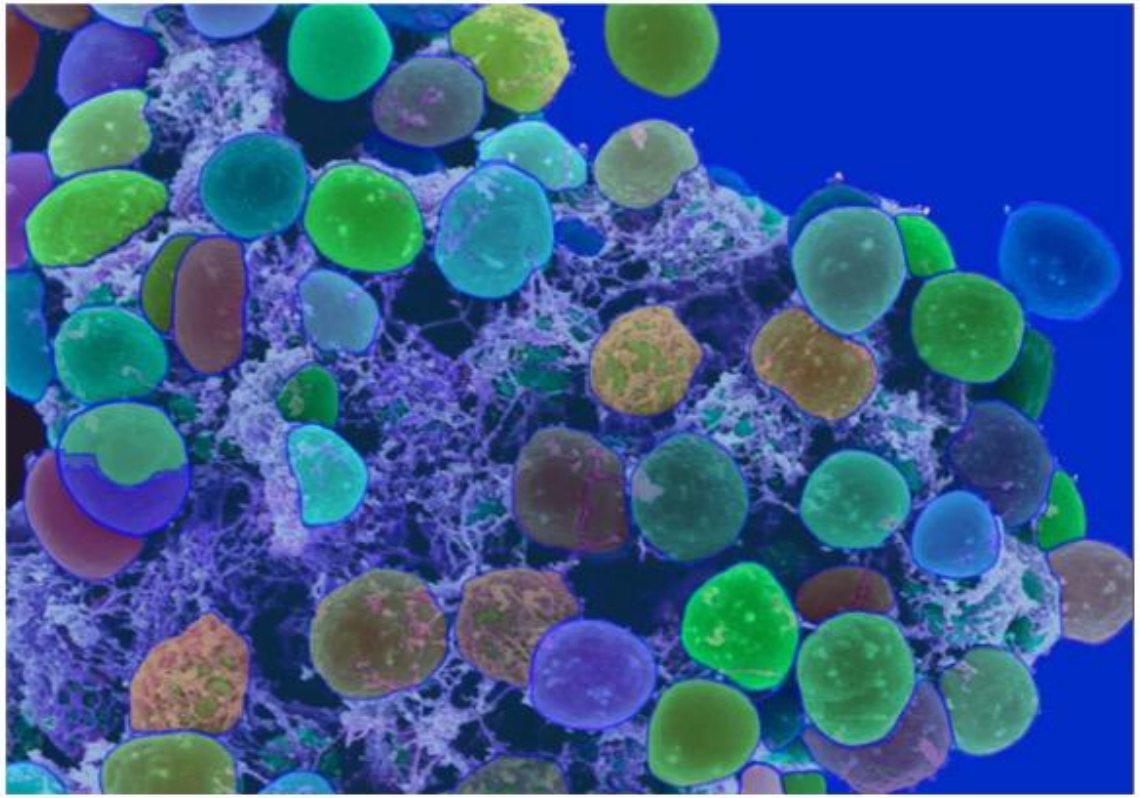
Images: Kirillov et al. Segment Anything. 2023

SAM Results

Images: Kirillov et al. Segment Anything. 2023

Zero-Shot with SAM





Images: Kirillov et al. Segment Anything. 2023

Zero-Shot with SAM

Images: Kirillov et al. Segment Anything. 2023

Chaining Foundation Models

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)

LEFT:

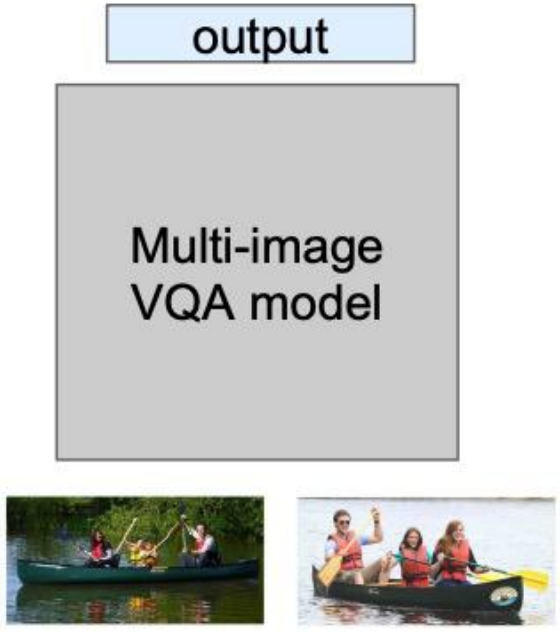
RIGHT:

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



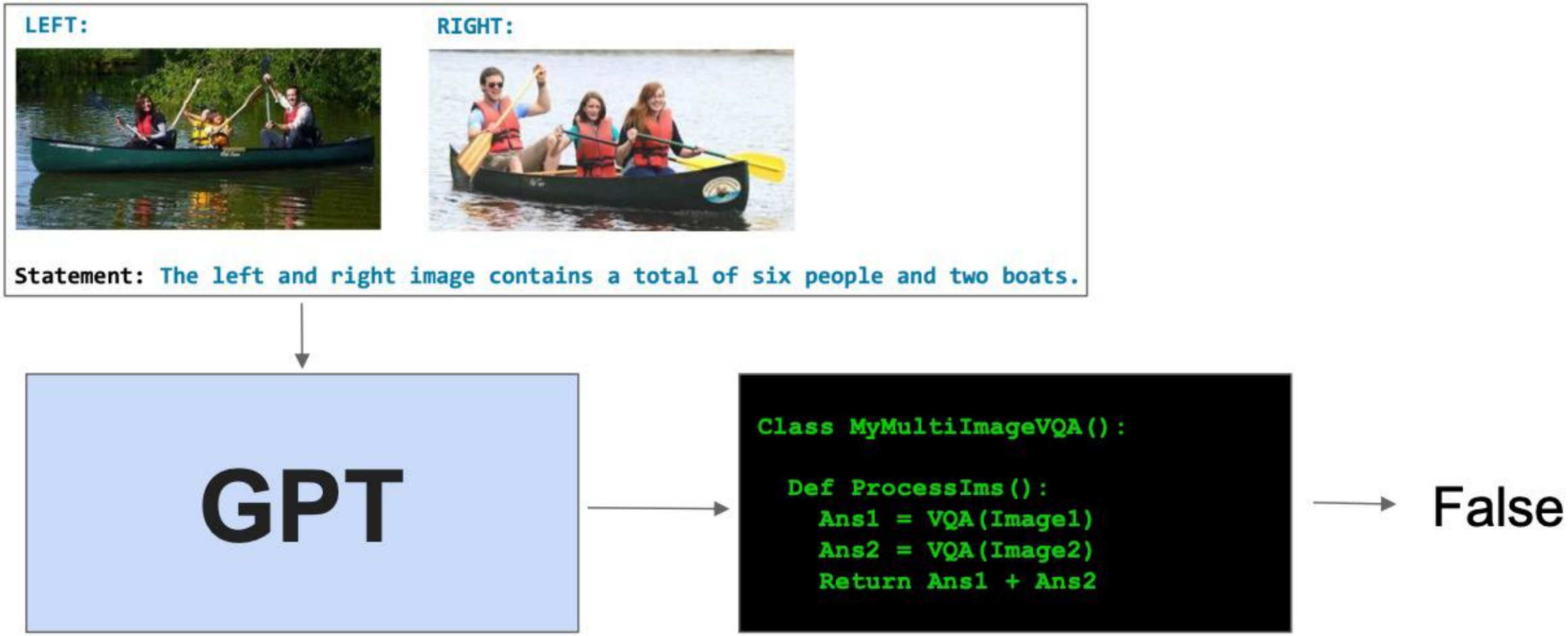
Write a python script with the models you have

```
Class MyMultiImageVQA():

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

General to 2 images now, but not beyond that

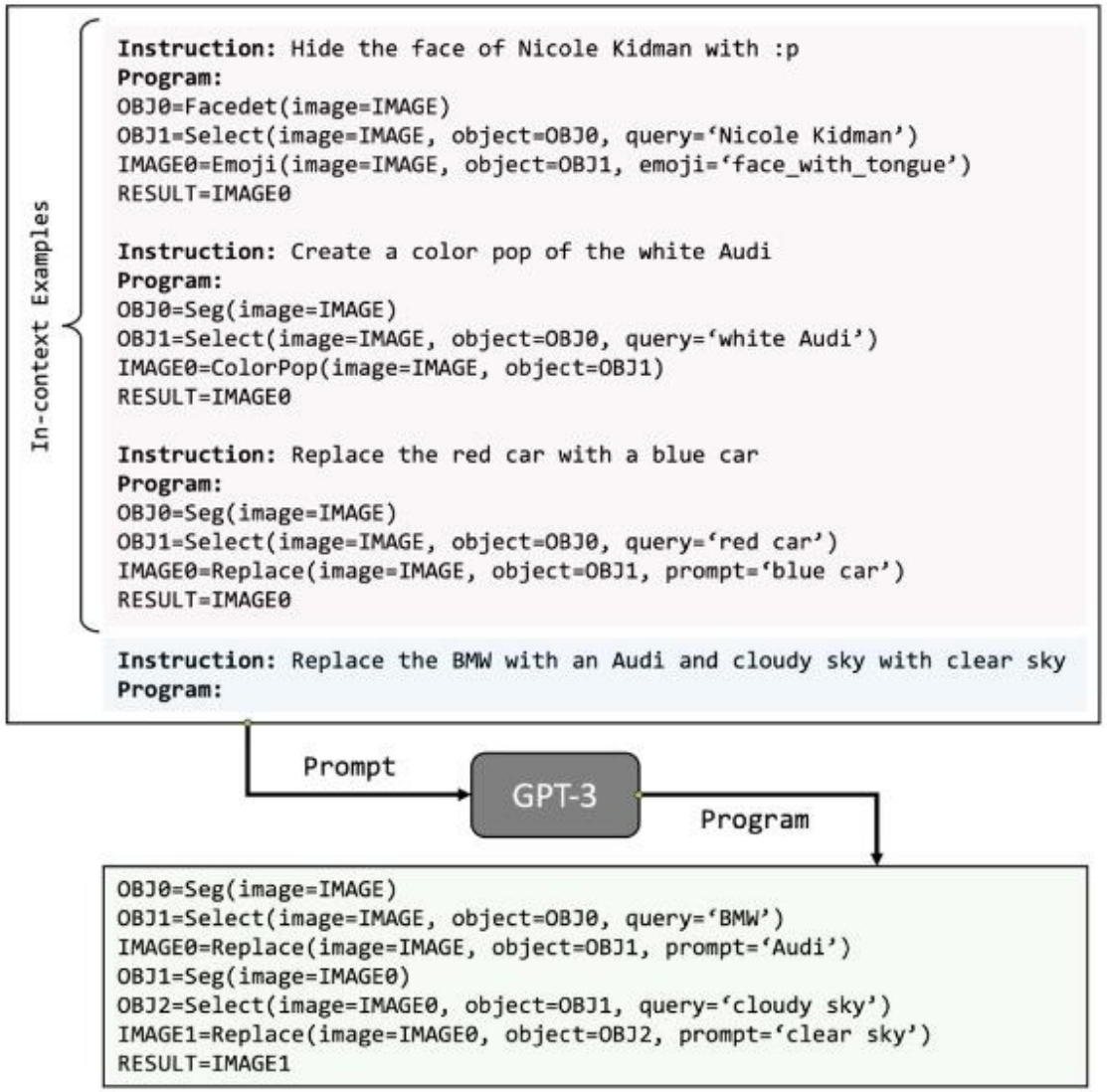
Gupta et al "Visual Programming: Compositional visual reasoning without training". 2023.

VisProg (visual programming)



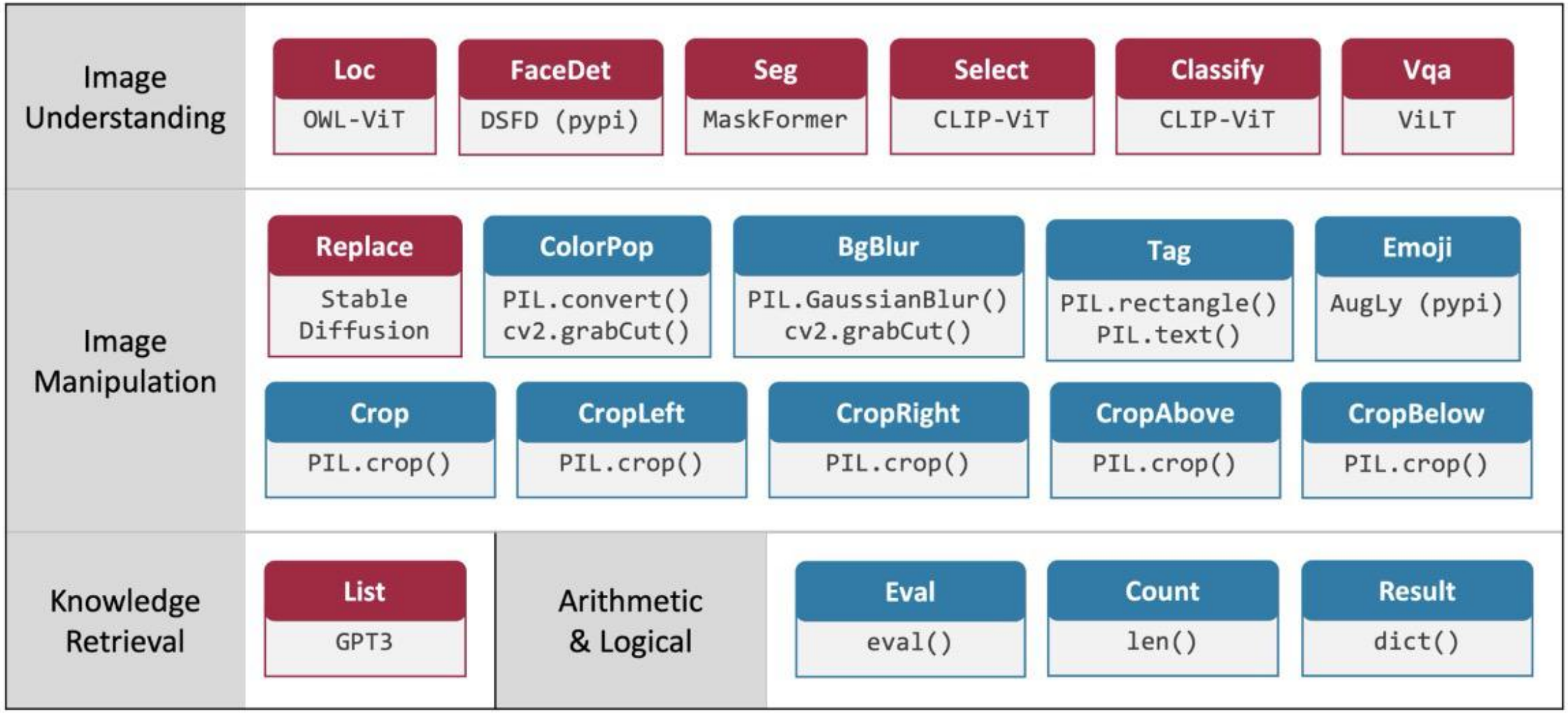
Gupta et al "Visual Programming: Compositional visual reasoning without training". 2023.

VisProg (visual programming)



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

VisProg (visual programming)



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: Foundation Models

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

The End