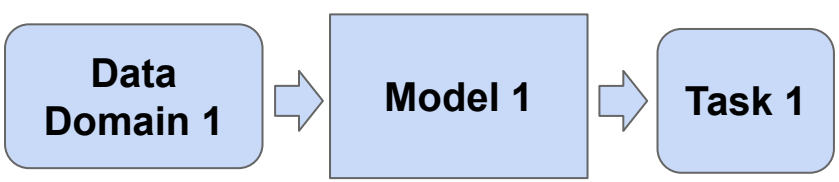
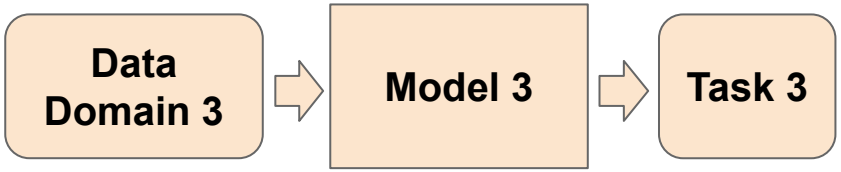
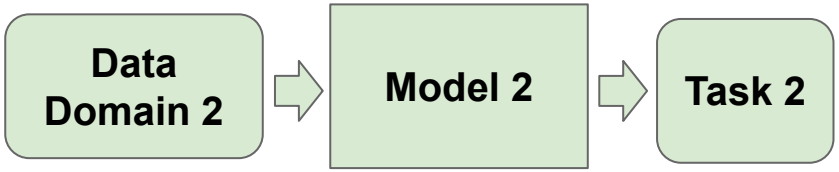
Lecture 17:
Multimodal Foundation Models

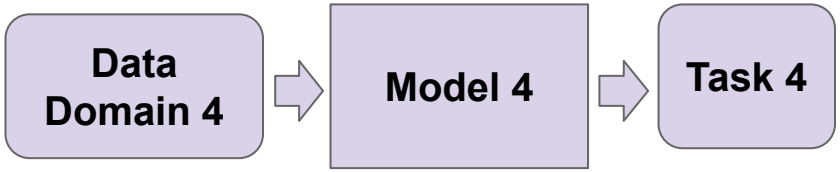
How have we been thinking about models in this class so far?

Train a *specialized* model for *each* task



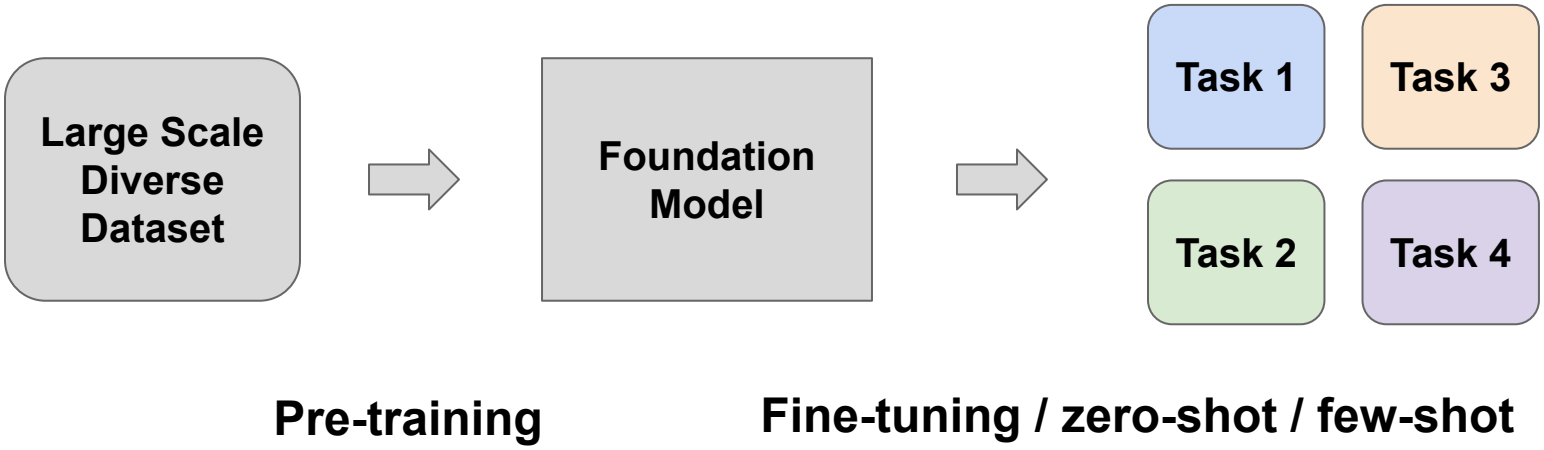






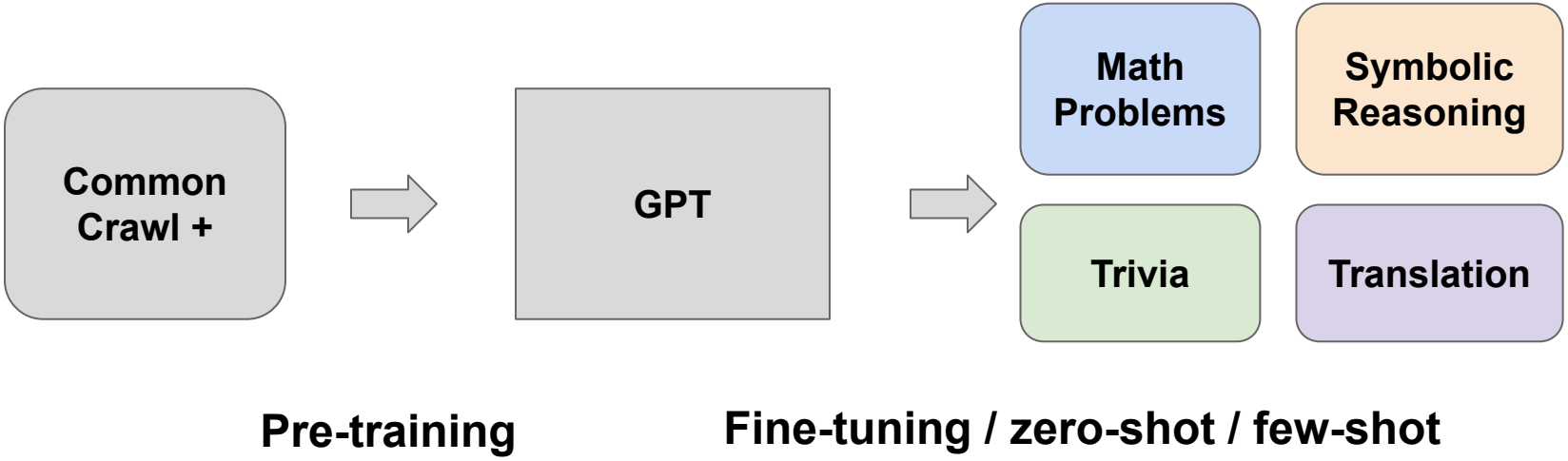
Now, we build Foundation Models

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



Foundation Models

Language



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	Dalle	Tool use
T5		Gemini	Stable Diffusion	
		Molmo	Imagen	

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	Dalle	Tool use
T5		Gemini	Stable Diffusion	
		Molmo	Imagen	

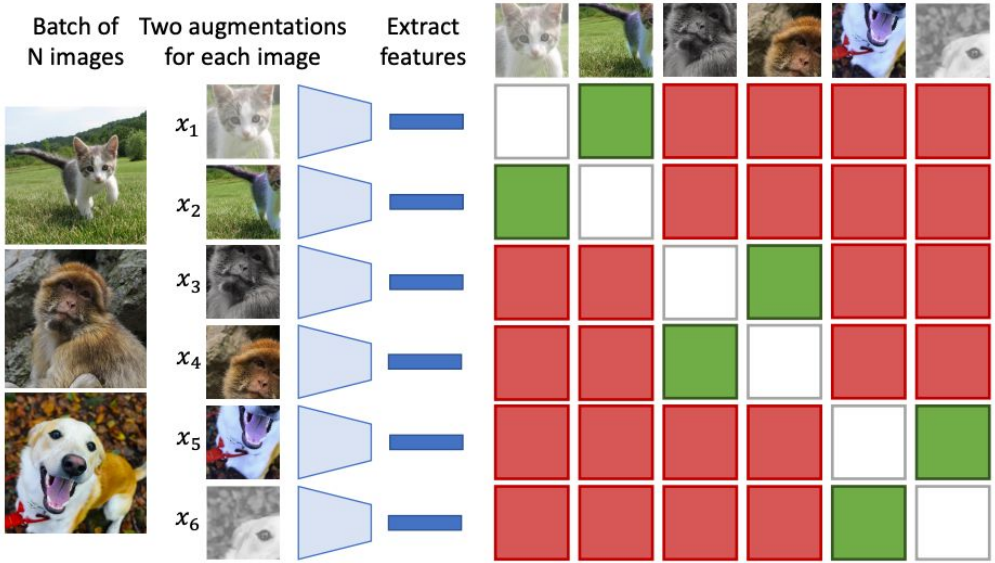
We will focus on multimodal (vision) foundation models

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

Let's start with the foundation models for classification

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

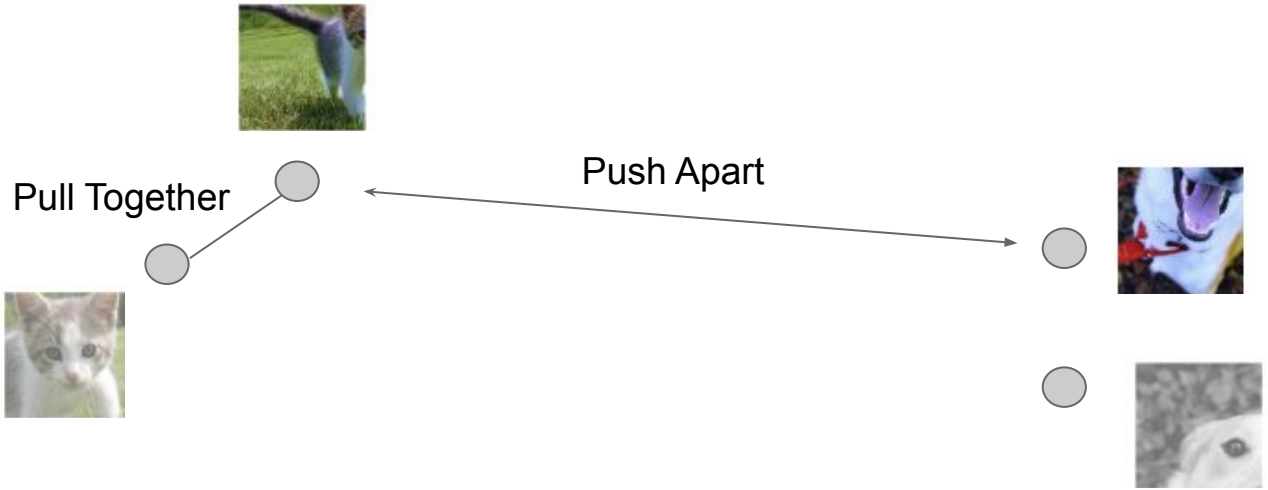
Recall this self-supervised objective from SimCLR



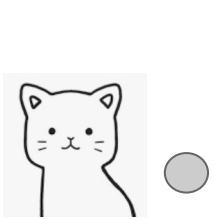
Use Self Supervised learning to learn good image features

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



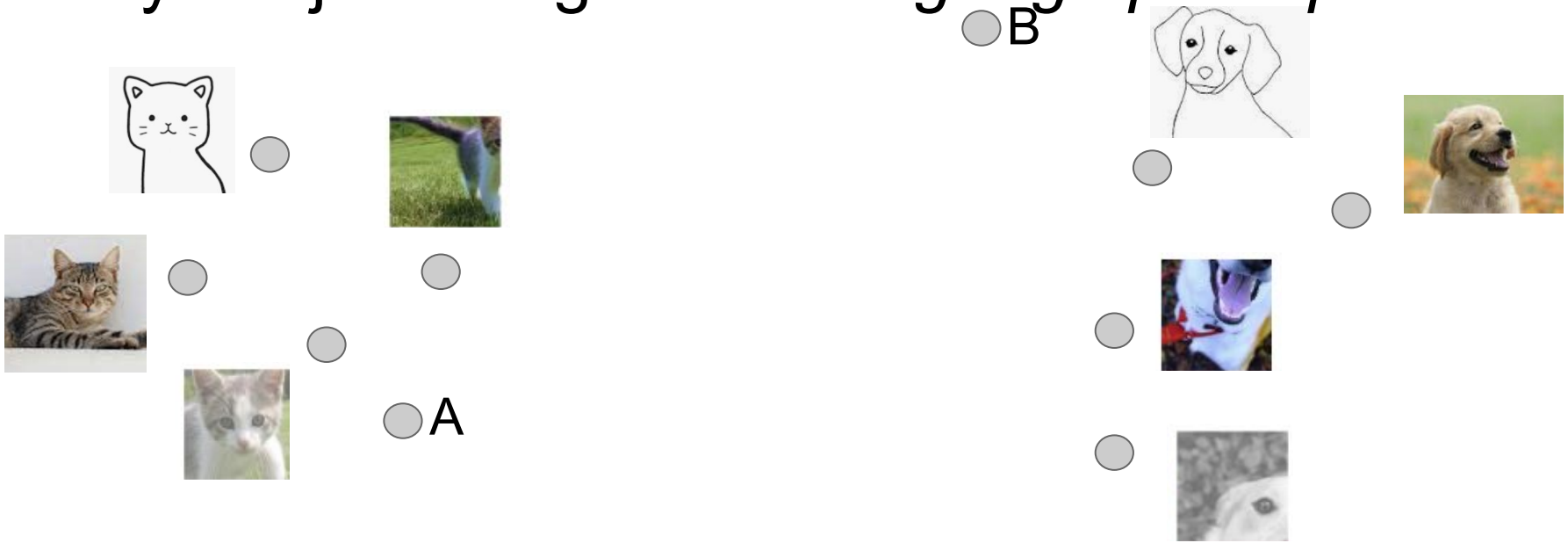
The hope was that the learned representations generalize to new instances








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



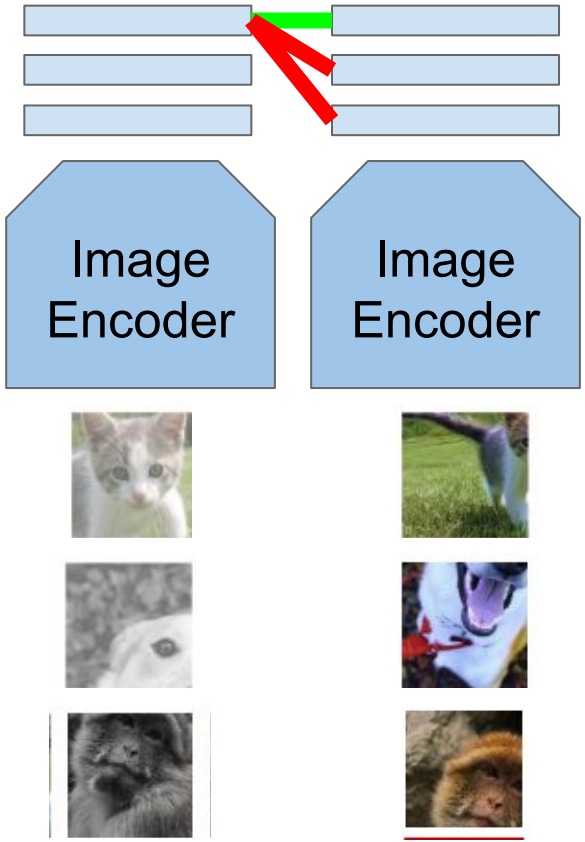
1. “A cute fluffy cat”
2. “My favorite dog is a golden retriever”

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

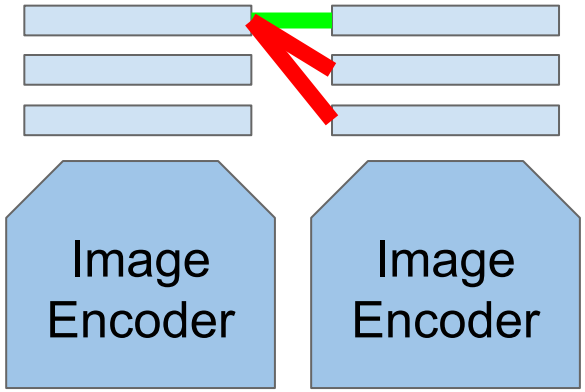
"A cute fluffy cat"

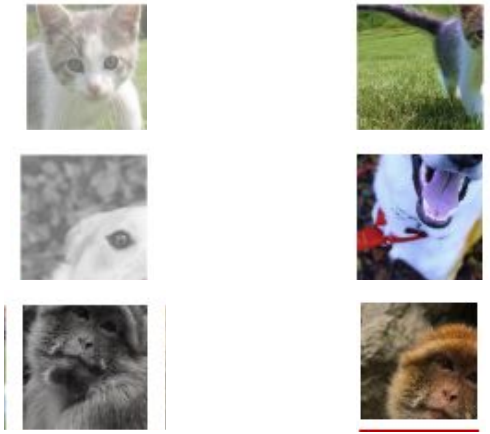
"My favorite dog is a golden retriever"

SimClr

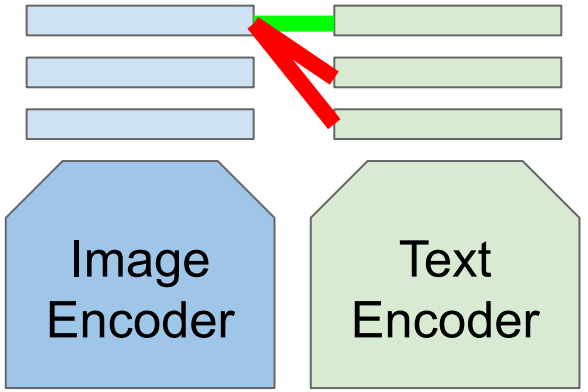


SimClr





CLIP



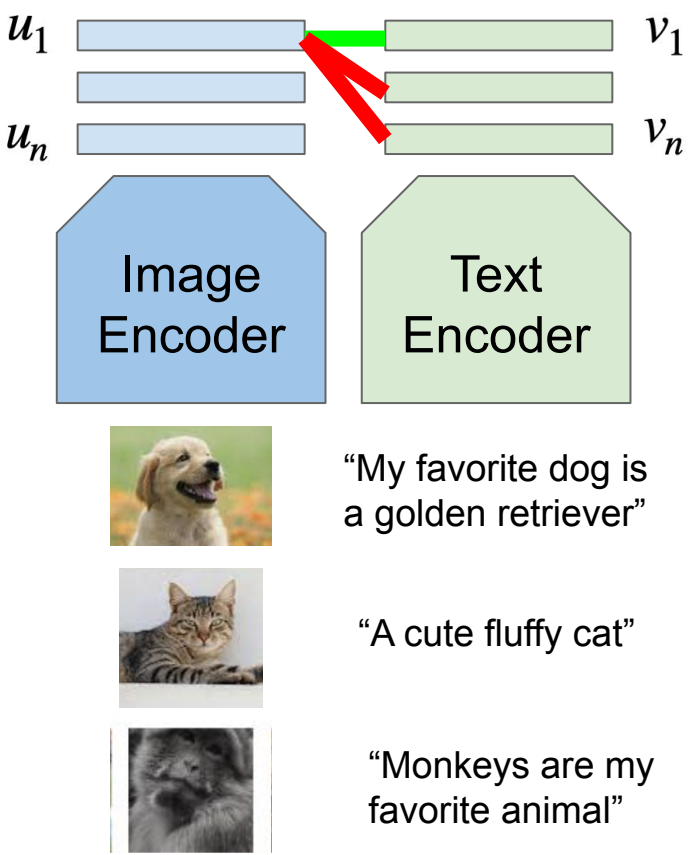
"My favorite dog is a golden retriever"

"A cute fluffy cat"

"Monkeys are my favorite animal"

CLIP is trained with the same contrastive objective

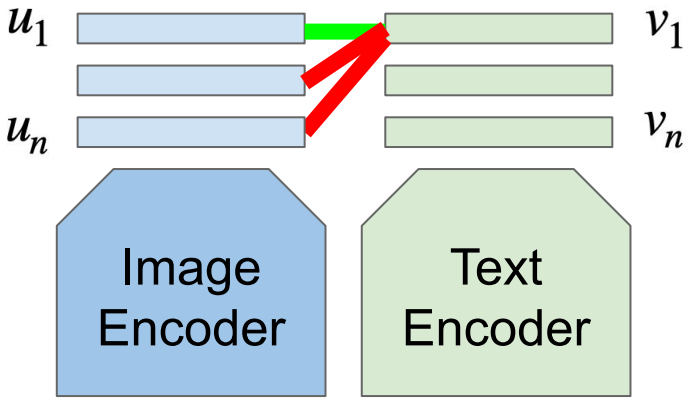
$$\sum_{i=1}^{n} -\log\left(\frac{e^{\langle u_i, v_i\rangle}}{\sum_{j=1}^{n} e^{\langle u_i, v_j\rangle}}\right)$$



CLIP 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"

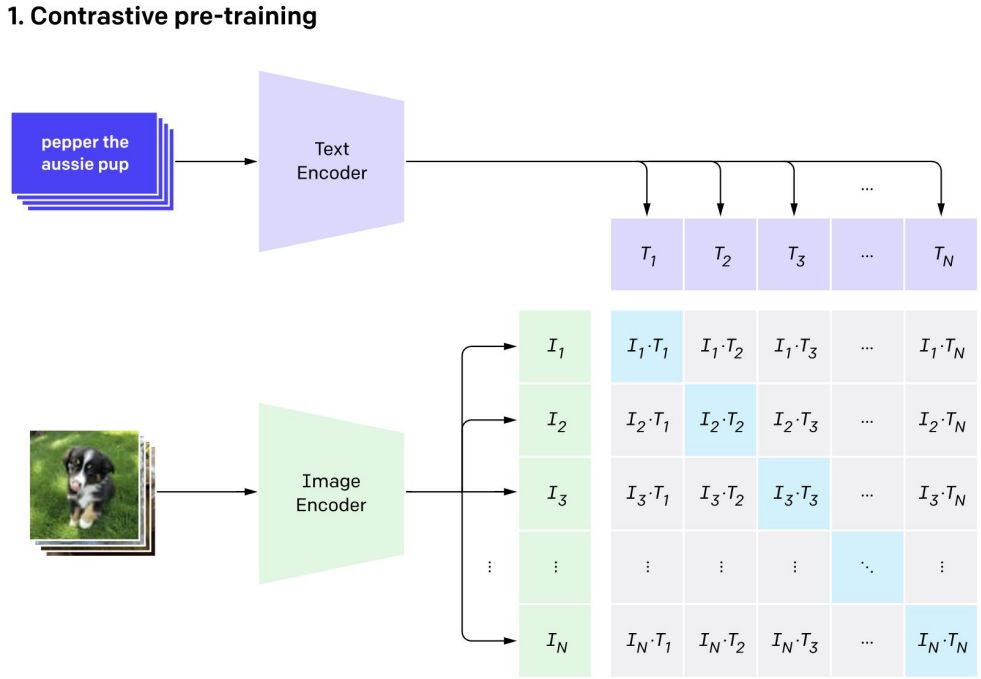
Lots of image-text data can be found online

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

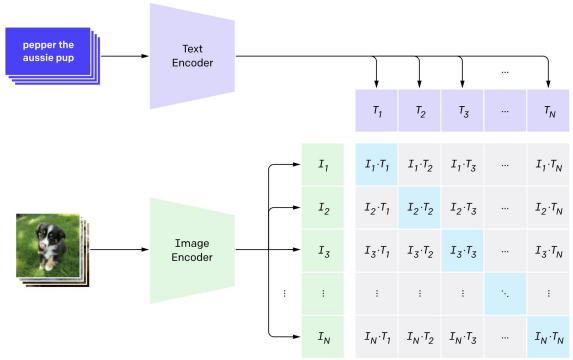
CLIP Training Objective



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

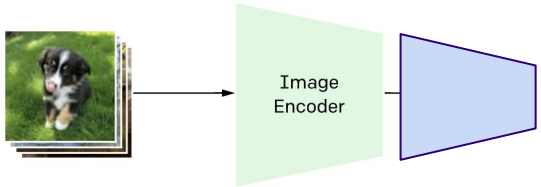
Using pre-trained models out of the box

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



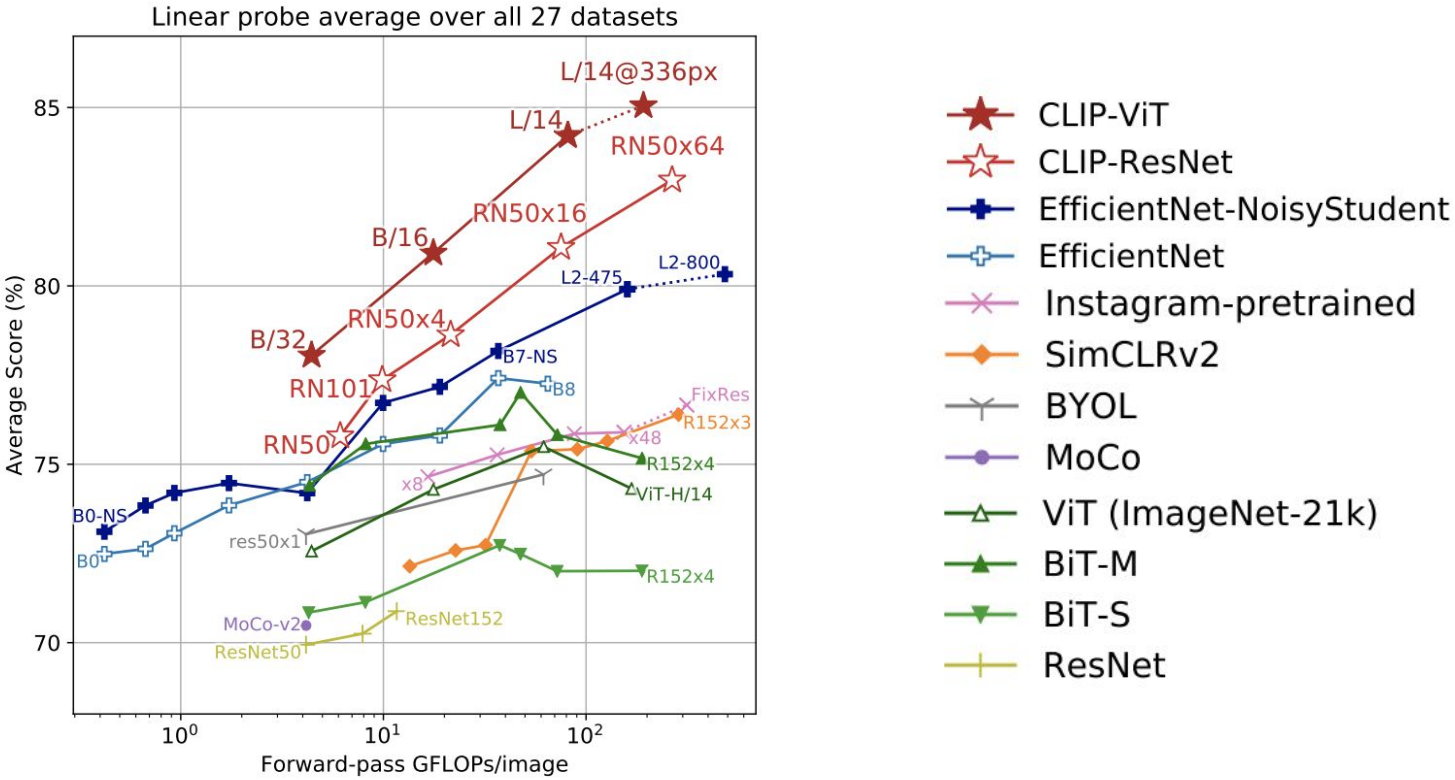
Pre-training tasks:
Contrastive Objective

Step 2: Transfer encoder to downstream tasks via linear classifiers



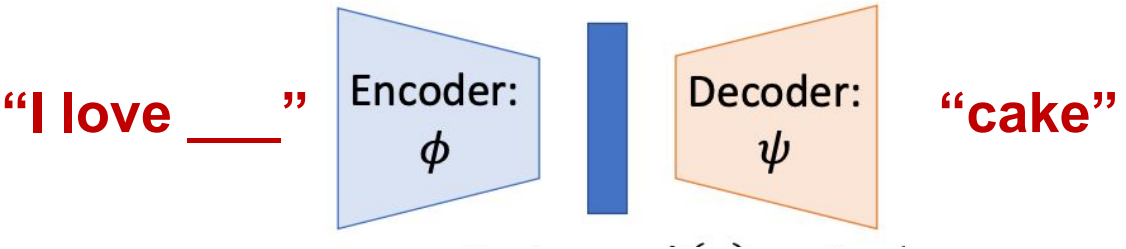
Downstream tasks:
Image classification, object detection, semantic segmentation

CLIP features w/ linear probe across multiple datasets

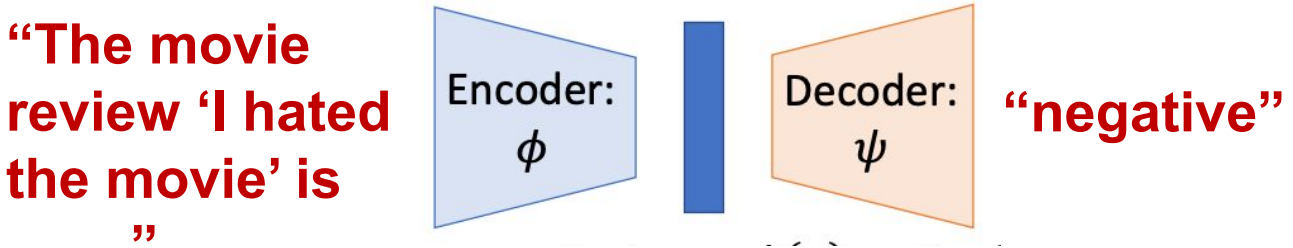


Big difference with language models: We can use LLMs zero-shot for new downstream tasks

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

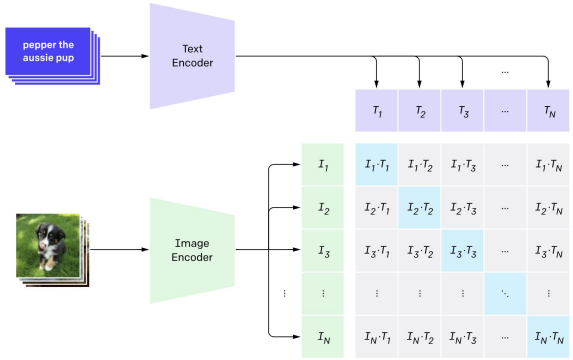


Step 2: Use the model out of the box in a creative way!



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

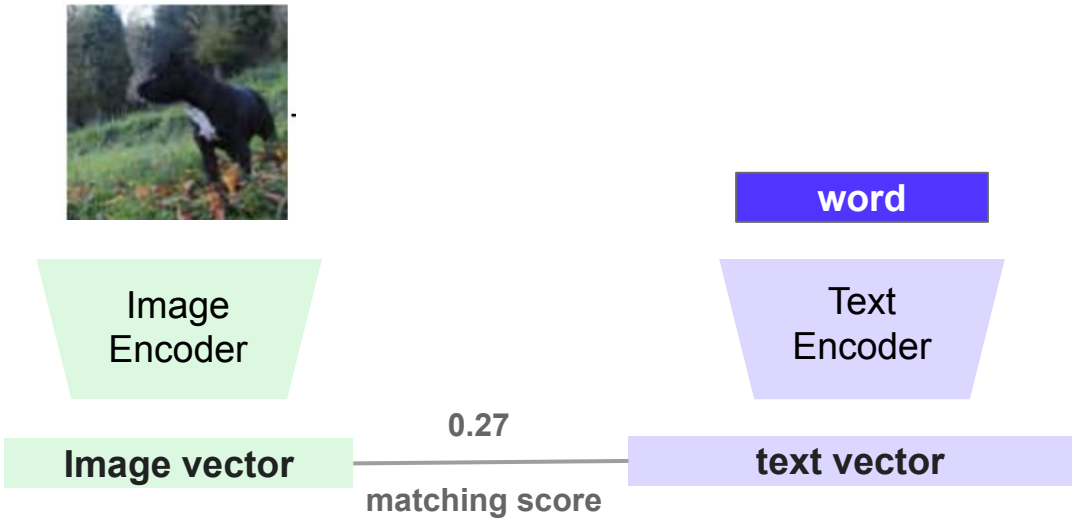


Pre-training tasks:
Contrastive Objective

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!

Image Encoder

Image vector

plane

dog

bird

Text Encoder

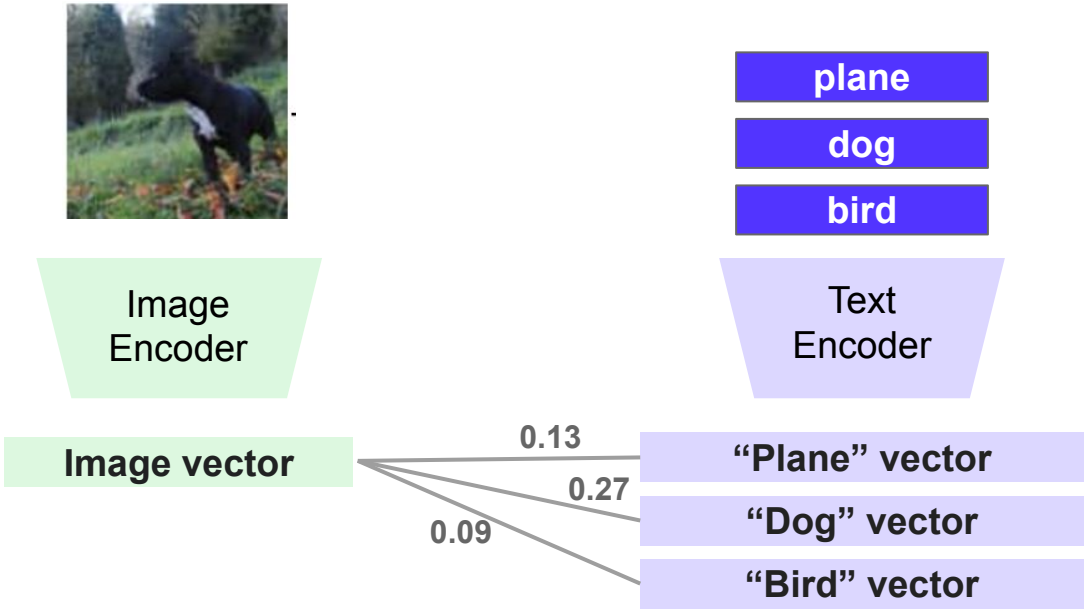
“Plane” vector

“Dog” vector

“Bird” vector

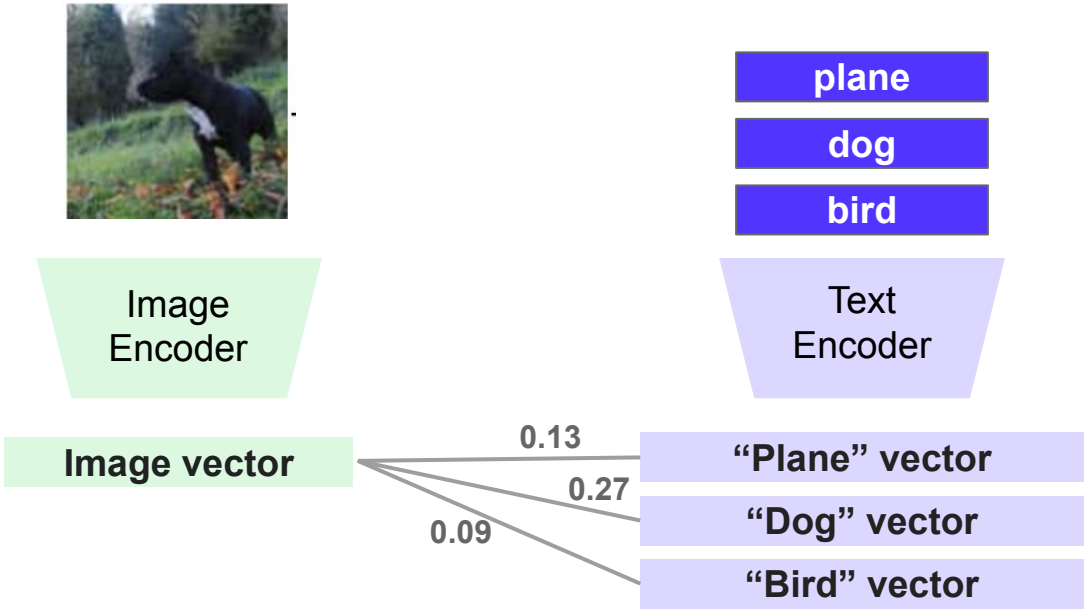
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Match a new image to the most similar vector



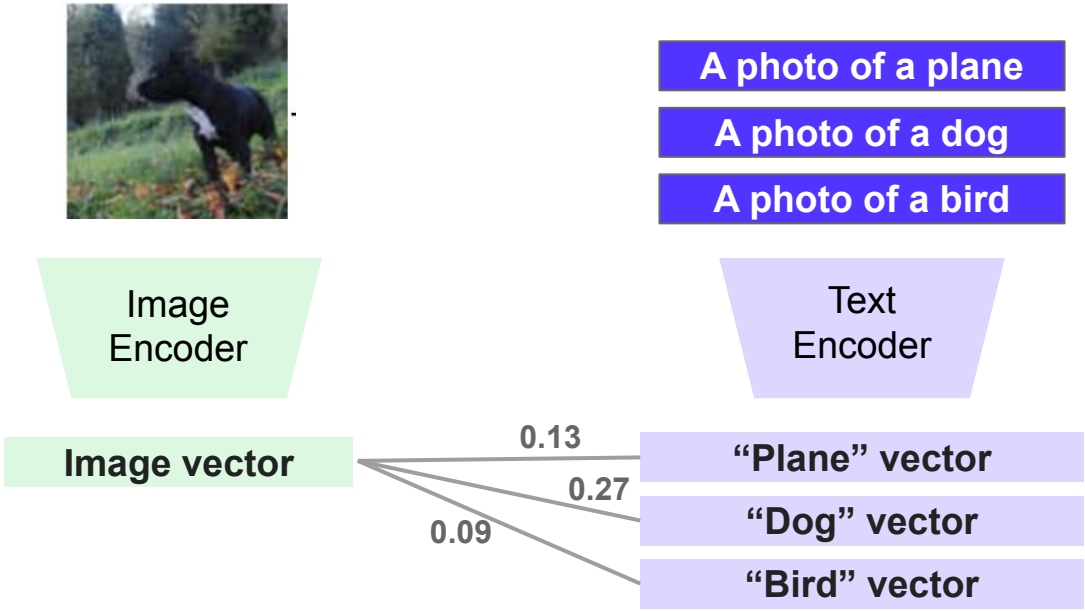
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

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

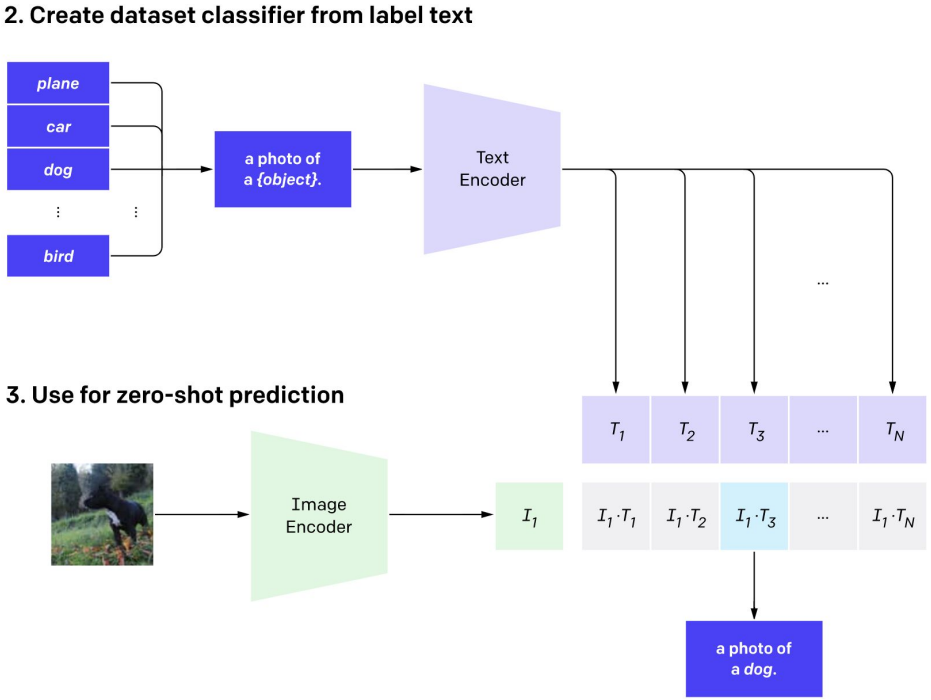
Mean “Plane” vector

Mean “Dog” vector

Mean “Bird” vector

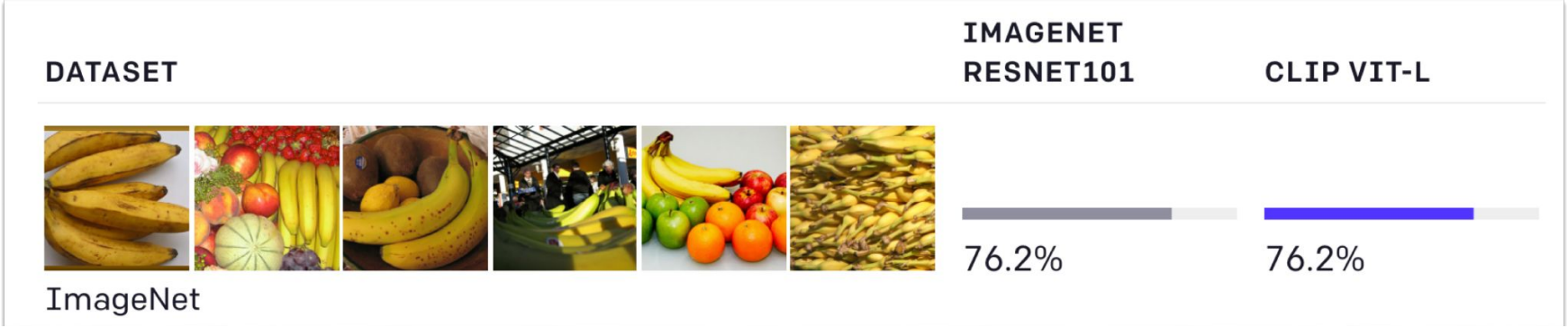
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

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!

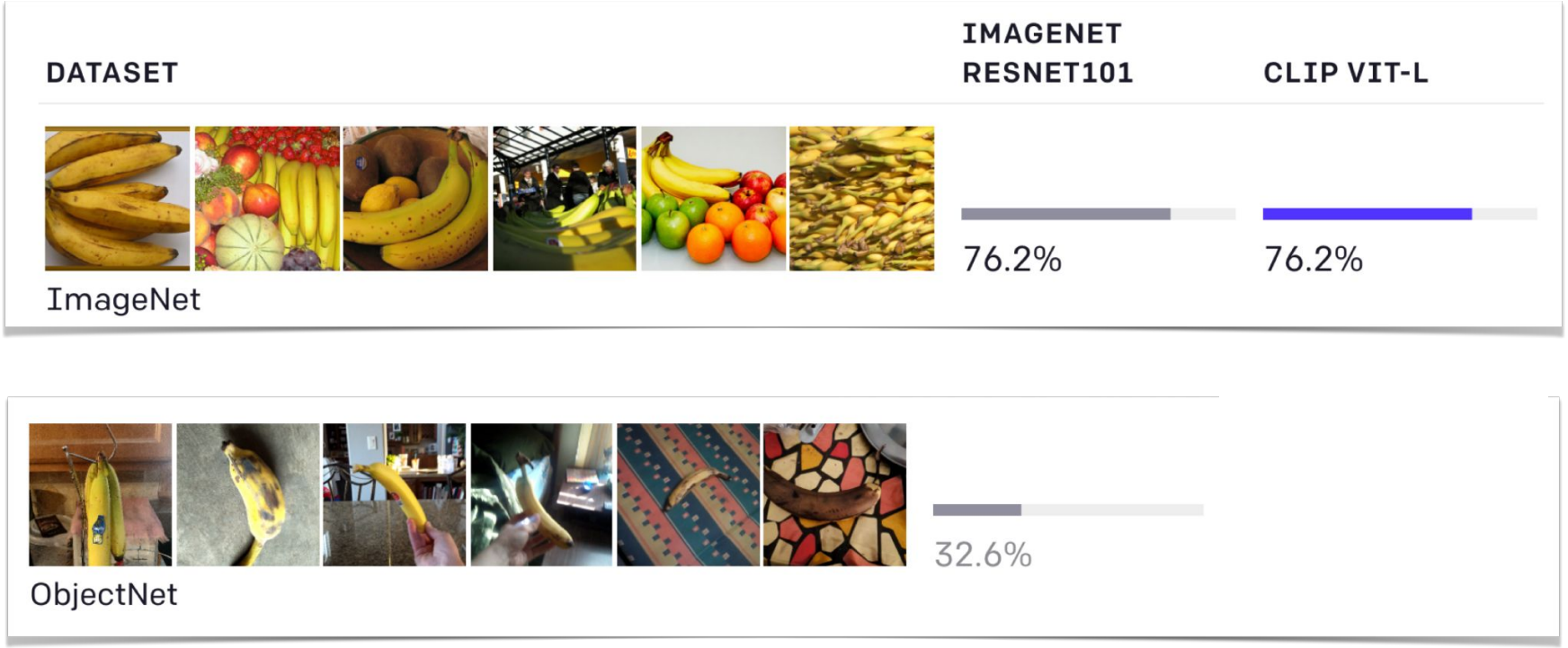
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

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”

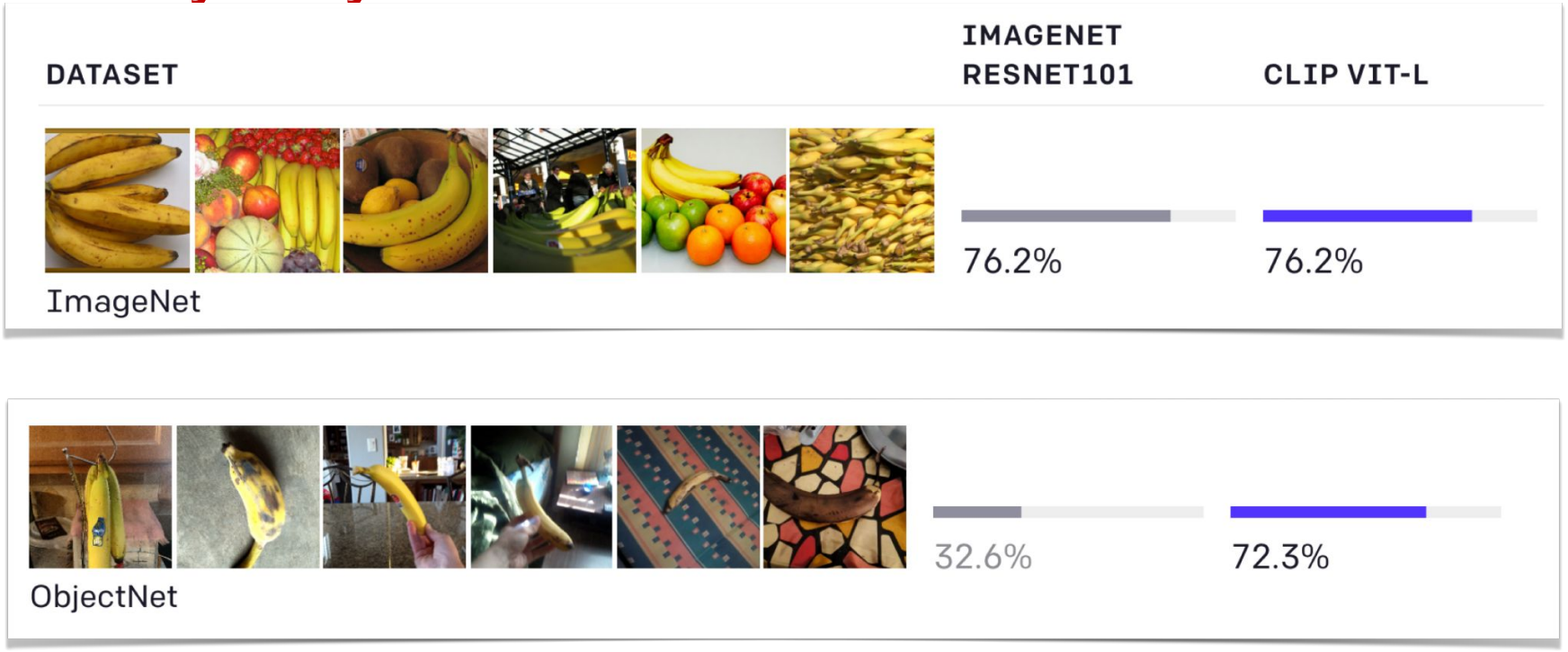
Training on ImageNet doesn't generalize to other datasets. ObjectNet contains the same categories but in weird viewpoints



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

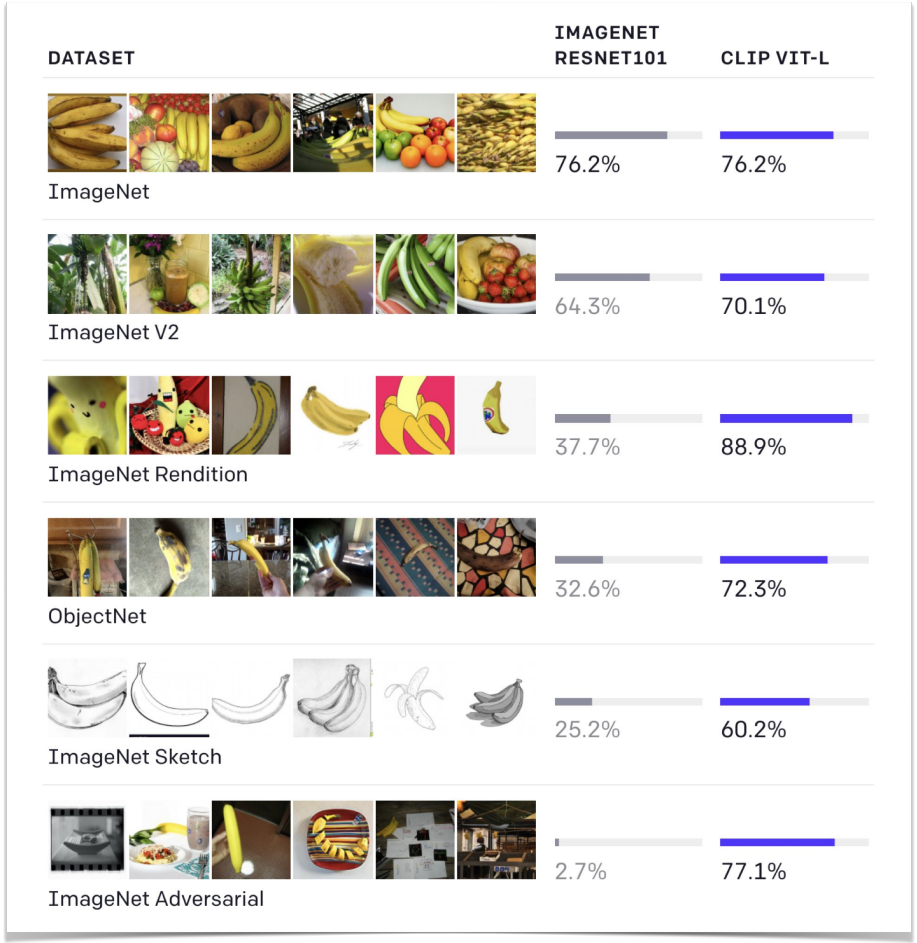
But CLIP zero-shot does so well!

Q. Why do you think that is?



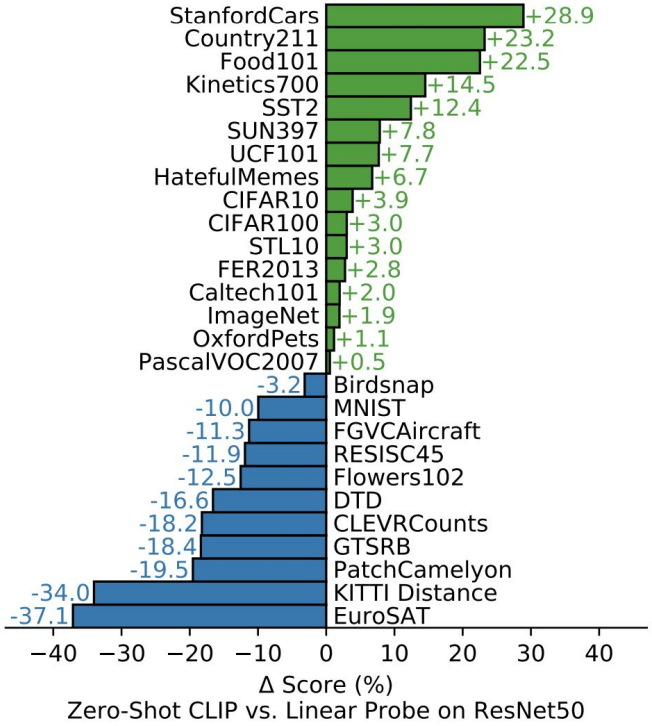
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

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?

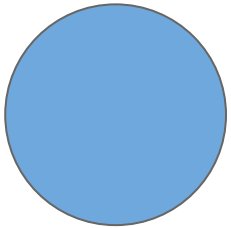
How can no labels beat labels??

Why does CLIP perform so well?

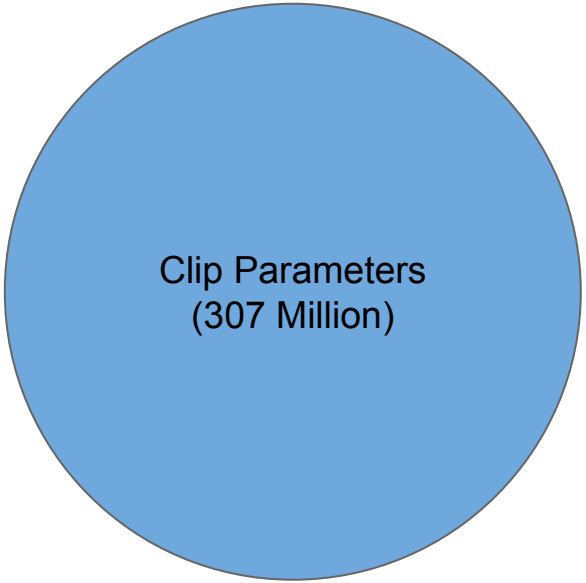
How can no labels beat labels??

Scale!

CLIP scaled up the model parameters with the transformer architecture

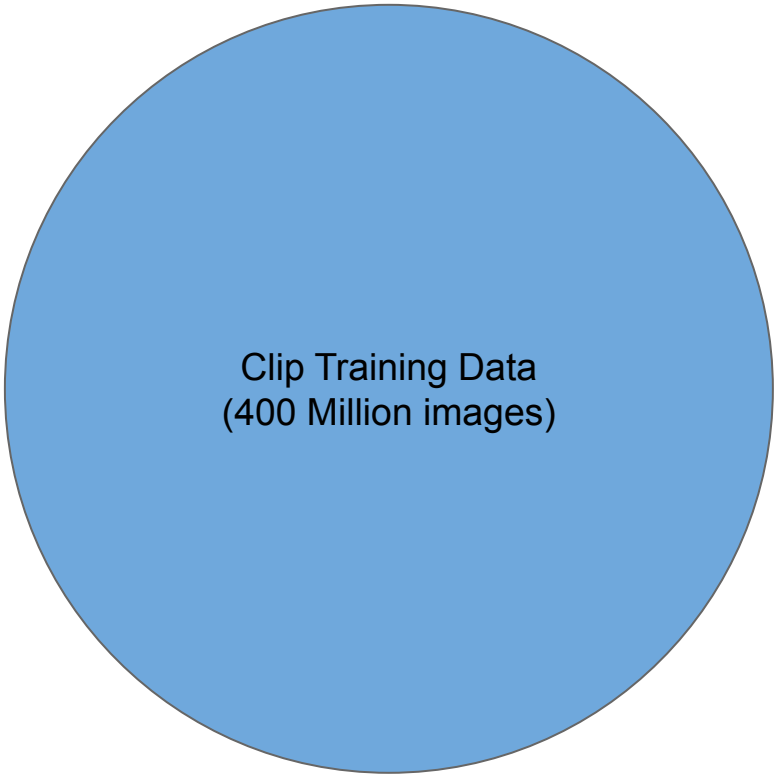


ImageNet ResNet Parameters
(44.5 Million)

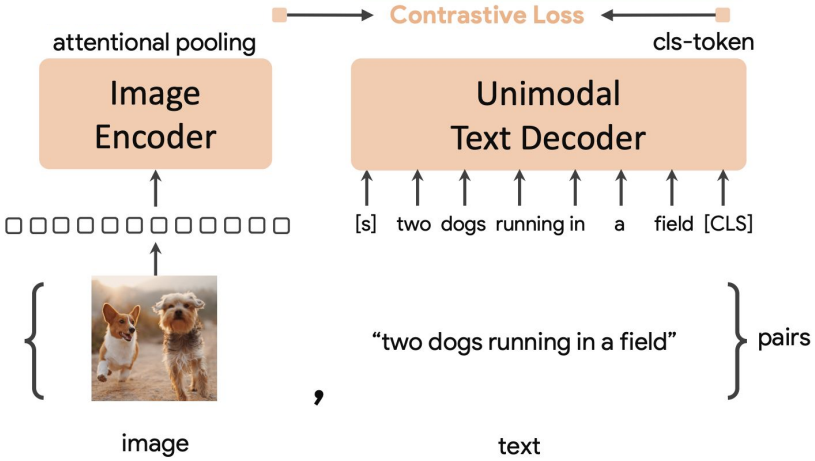


CLIP Scaled up the training data by scraping image-text pairs from the internet

ImageNet ResNet Training Data
(1.28 Million)

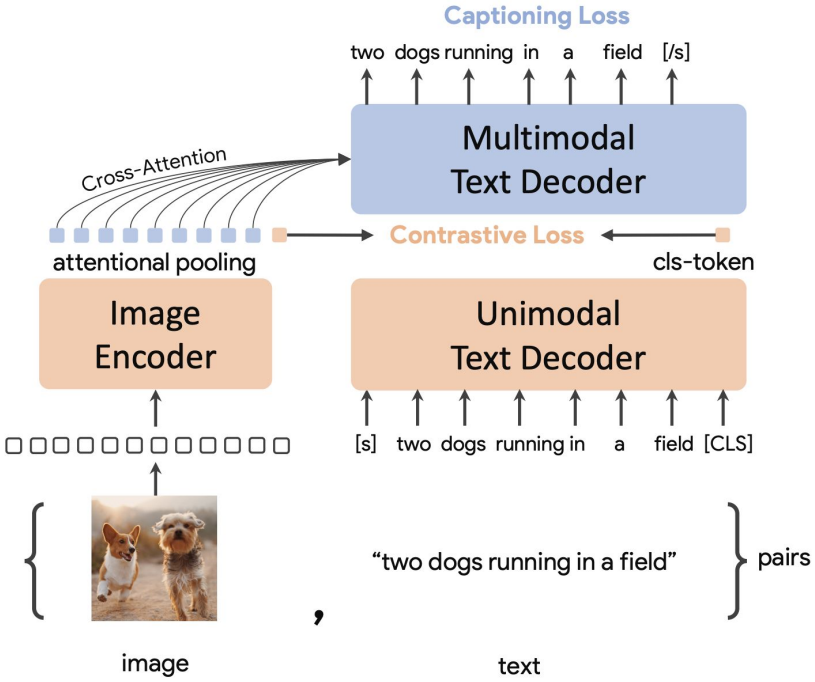


CoCa improved upon CLIP by adding a generation objective



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

CoCa added a decoder with a captioning loss



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

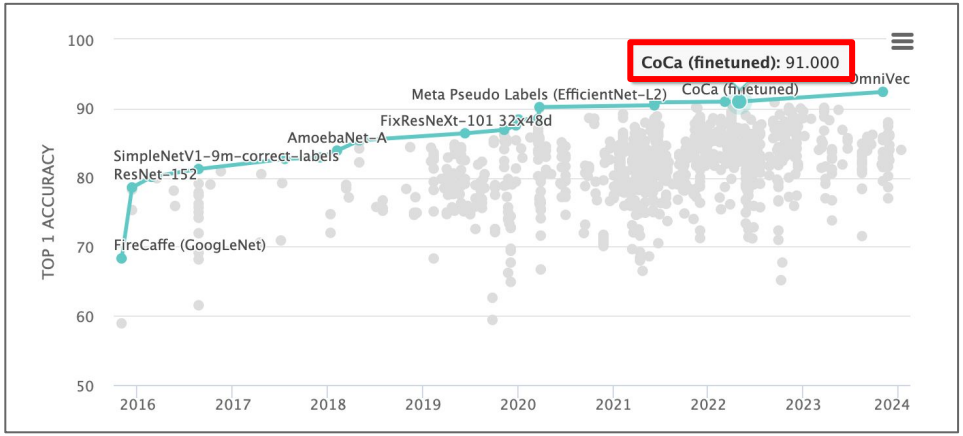
CoCa: Contrastive Captioners are Image-Text Foundation Models

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].

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**



Flamingo

Motivation: CLIP is extremely general in its learned representation, but limited in its out-of-the box applications.

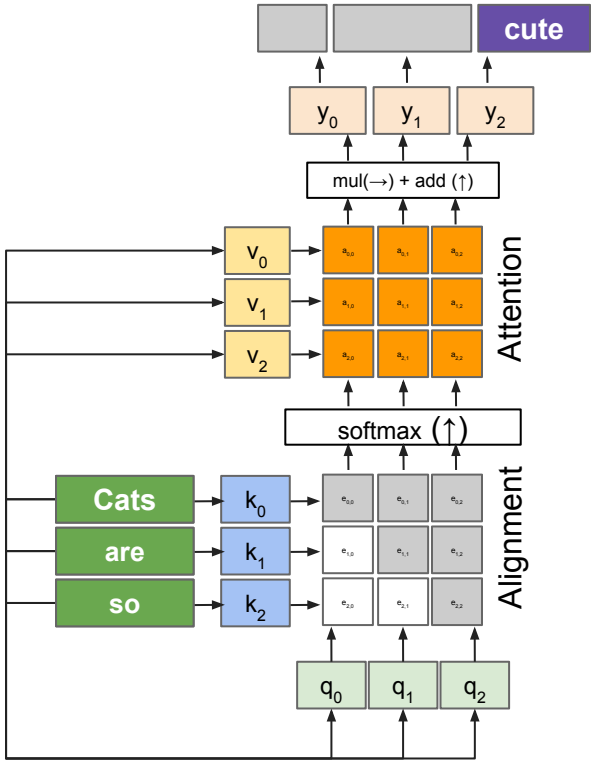
(only can output similarity scores between image and text)

Flamingo

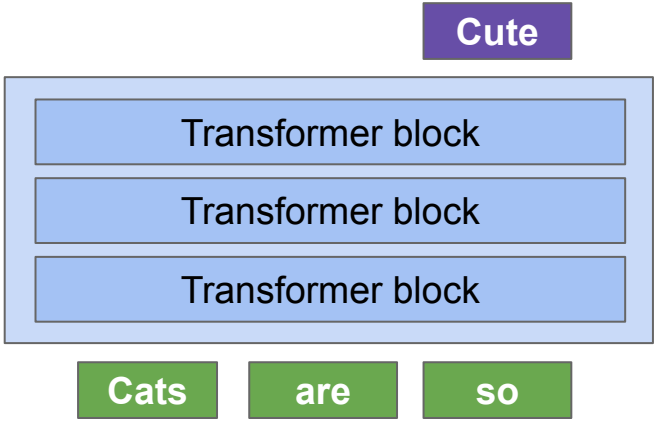
Motivation: Language models which do next token prediction can be applied to a wide variety of tasks at inference (Math, sentiment analysis, symbolic reasoning)

Can we build something like GPT but can accept images and text as input, and then output text?

Flamingo

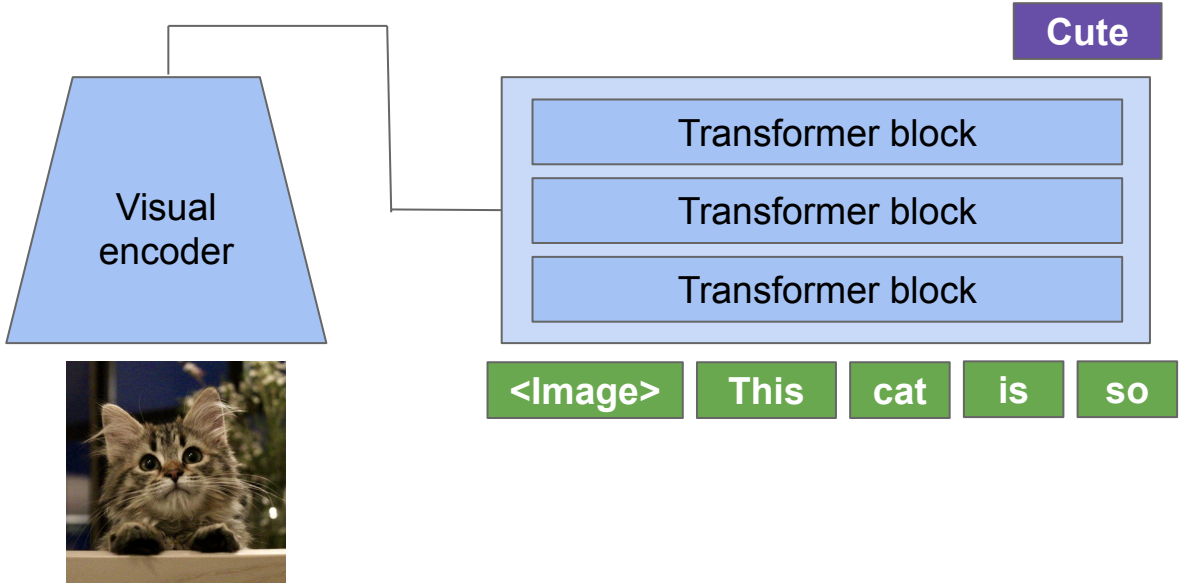

cute
y_0
y_1
y_2
mul(→) + add (↑)
v_0
v_1
v_2
Attention
softmax (↑)
Cats
are
so
k_0
k_1
k_2
Alignment
q_0
q_1
q_2

Flamingo

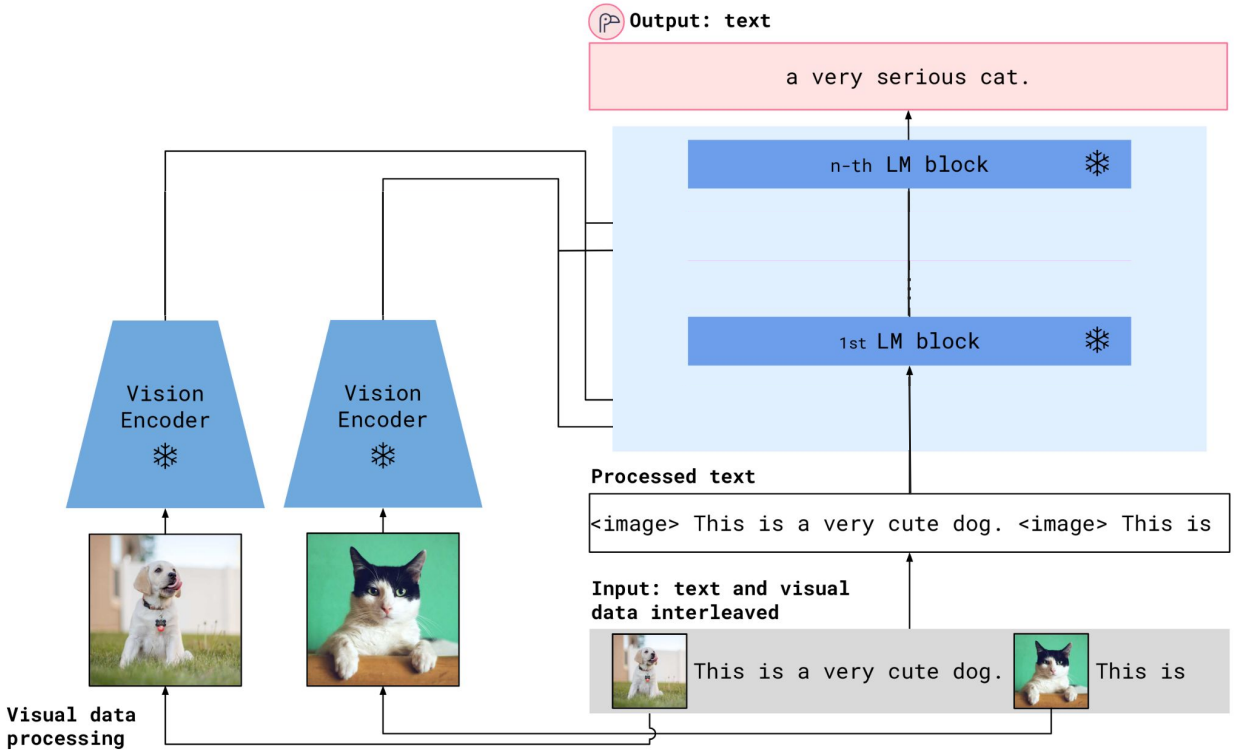

Cute
Transformer block
Transformer block
Transformer block
Cats
are
so

Flamingo

What kind of model is this? (think types of LLMs)

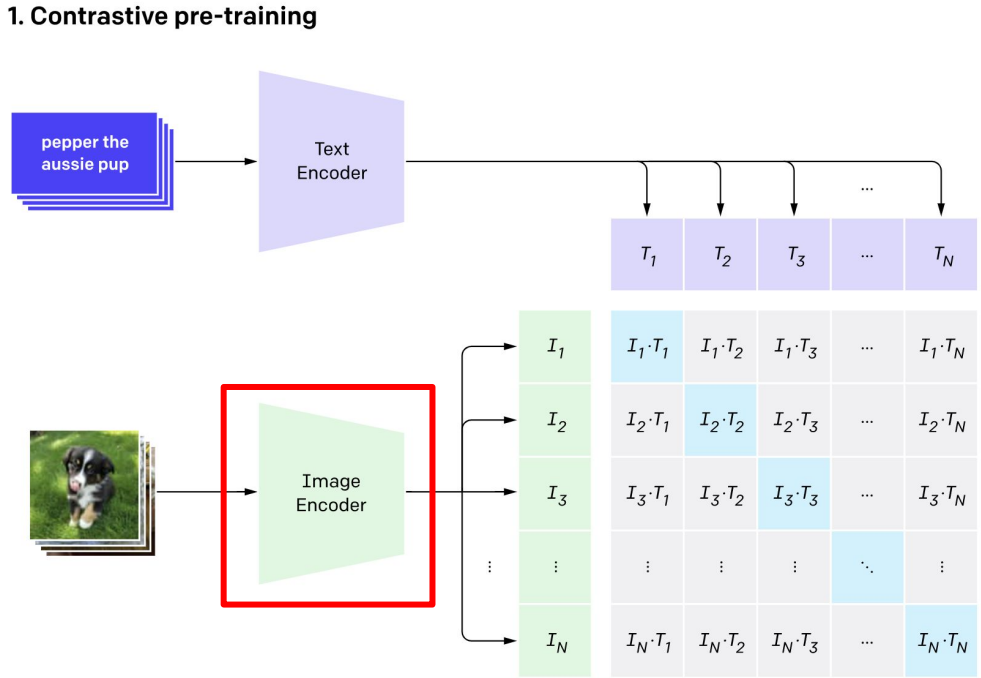


Pre-trained parts of Flamingo



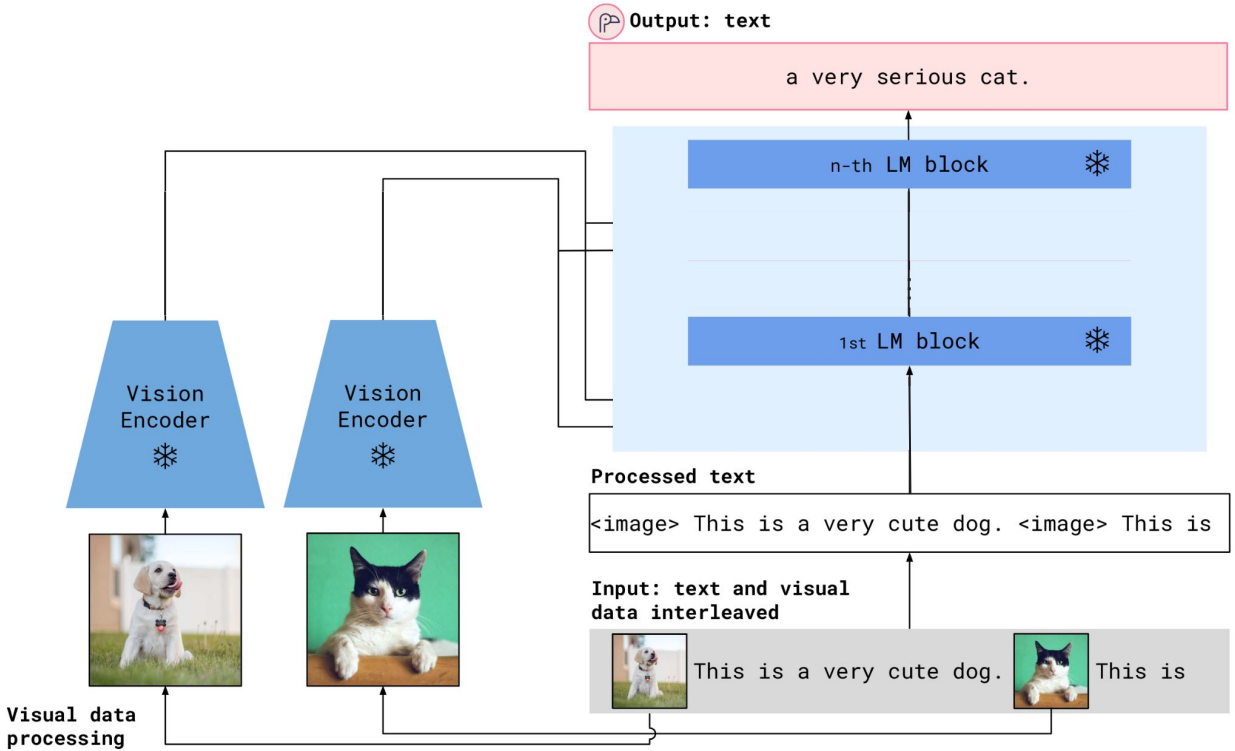
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

CLIP Training Objective



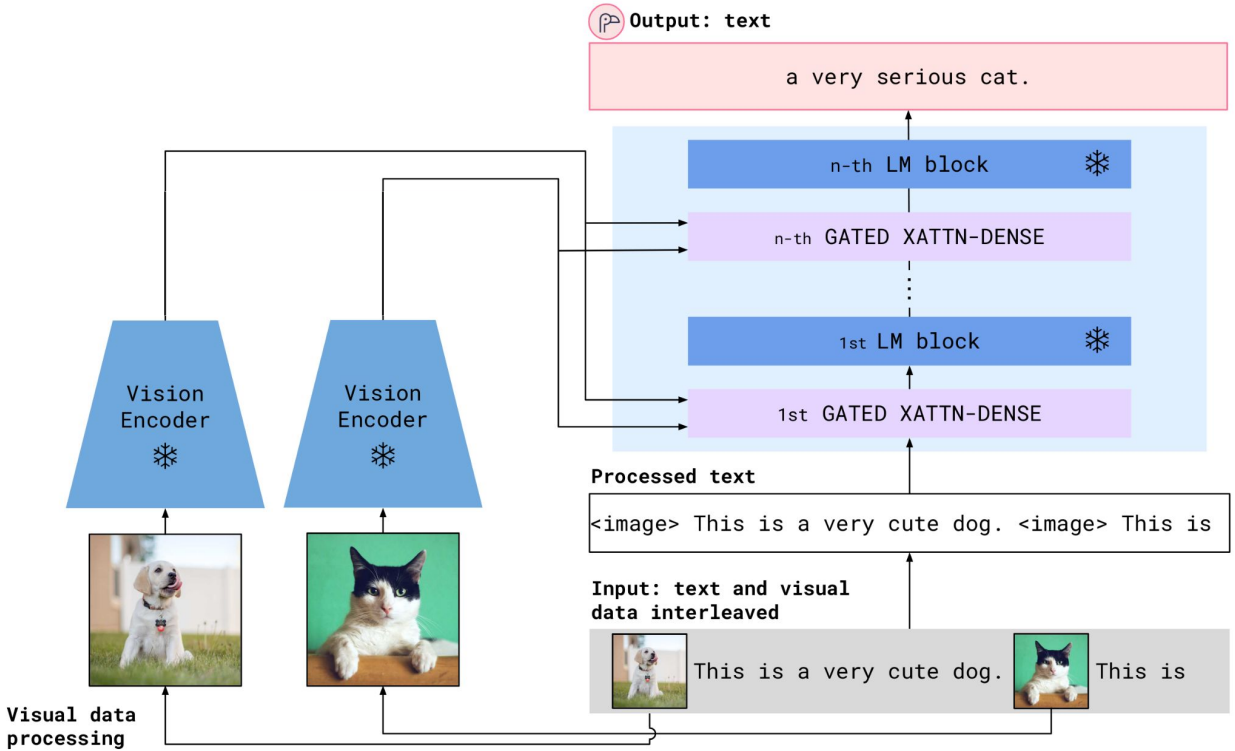
At the end of training, you have a model that will give you a similarity score between an image and a text

Pre-trained parts of Flamingo



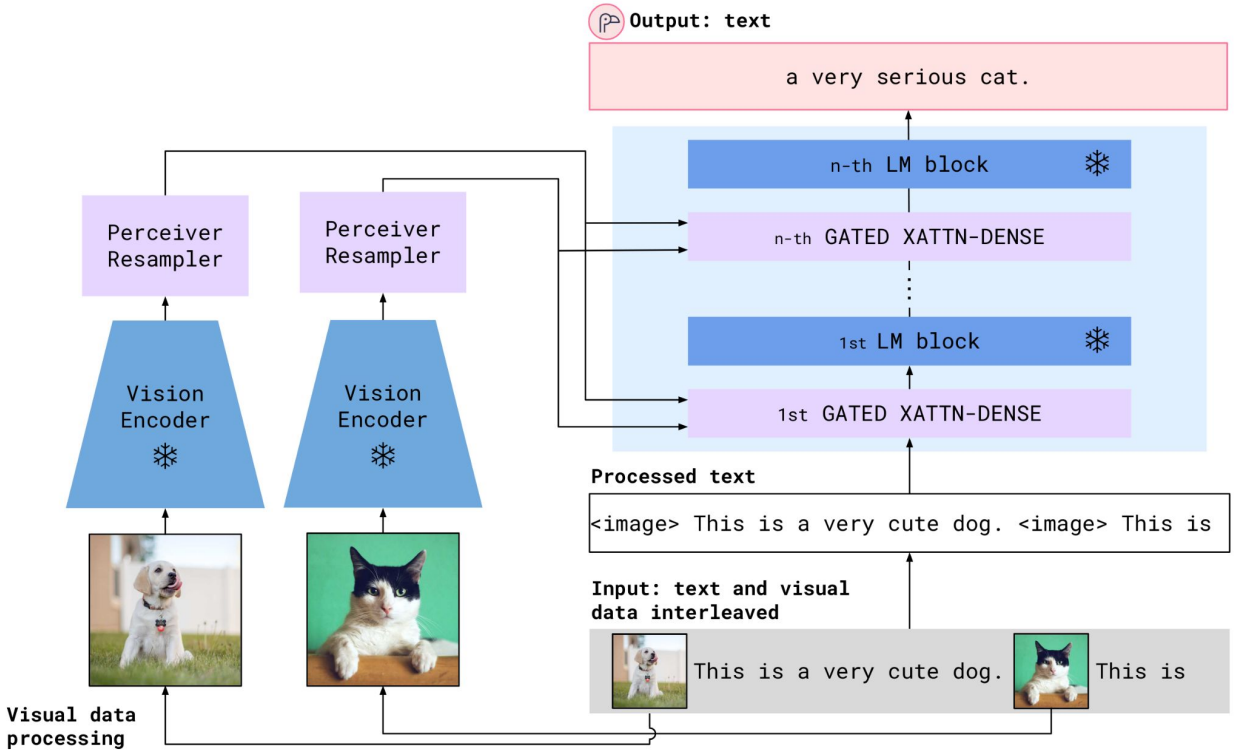
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Learned parts of Flamingo



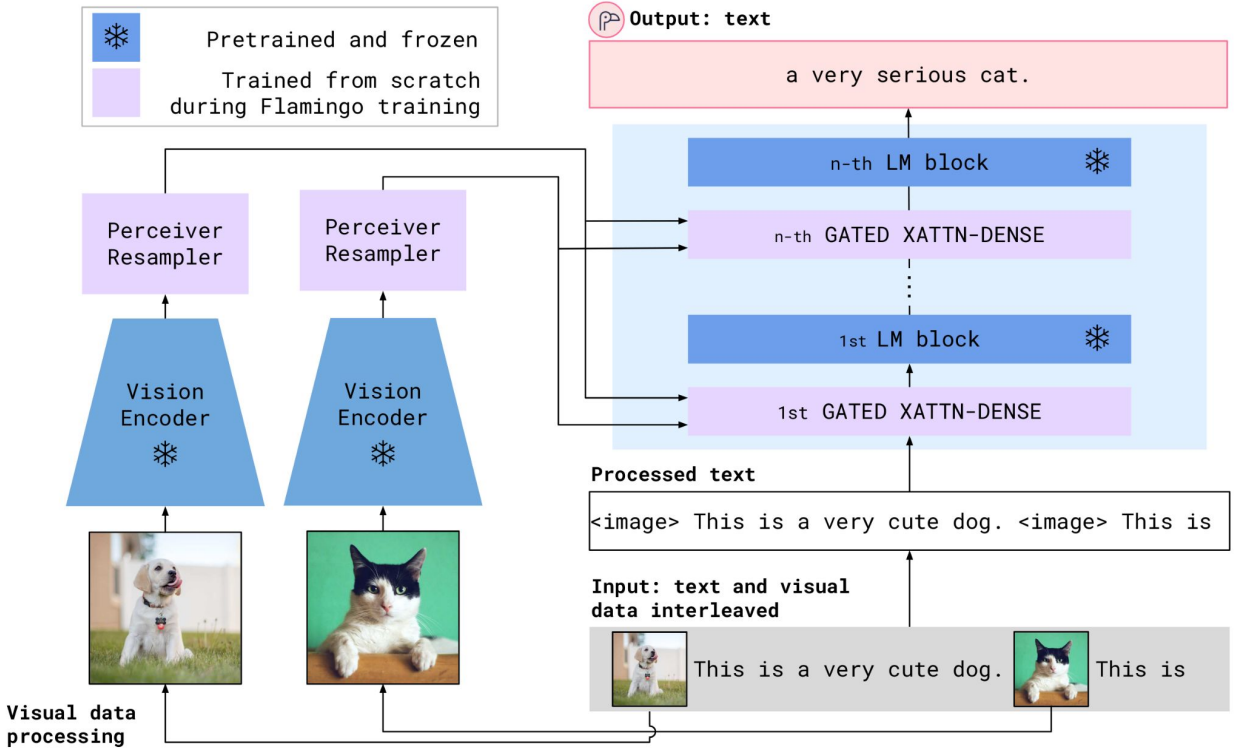
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Learned parts of Flamingo



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

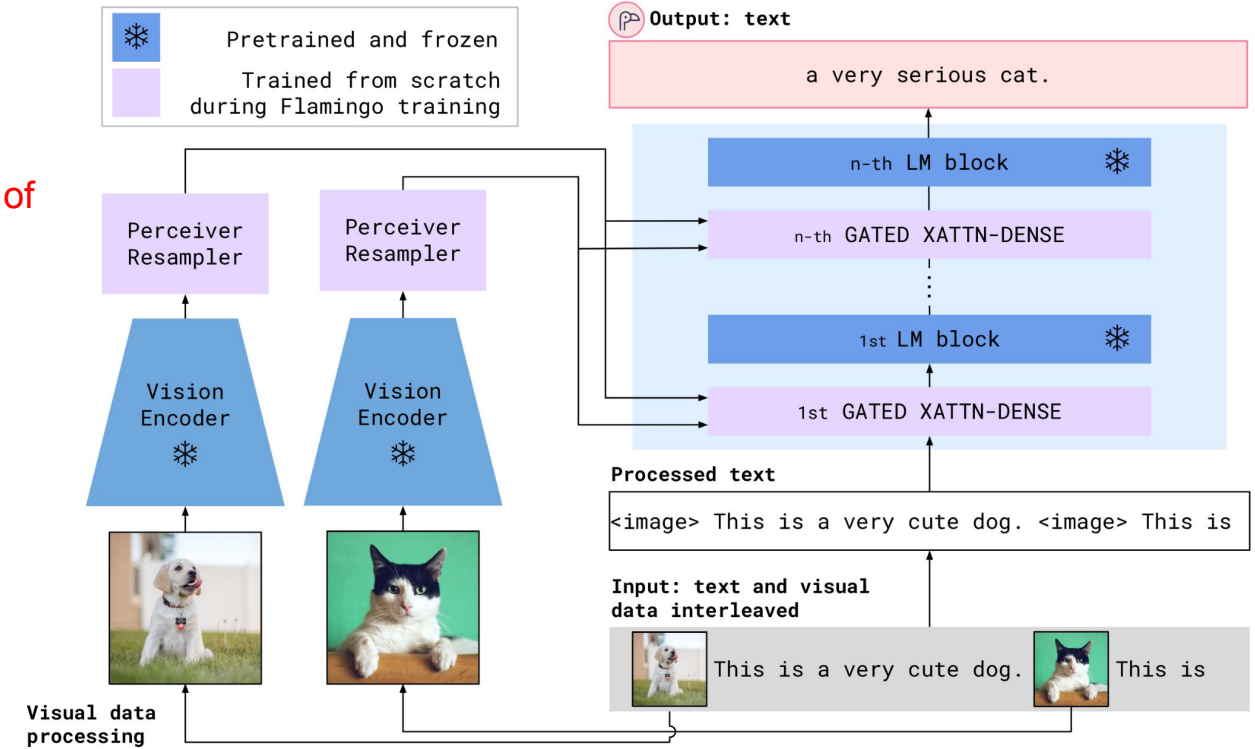
Flamingo full architecture



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

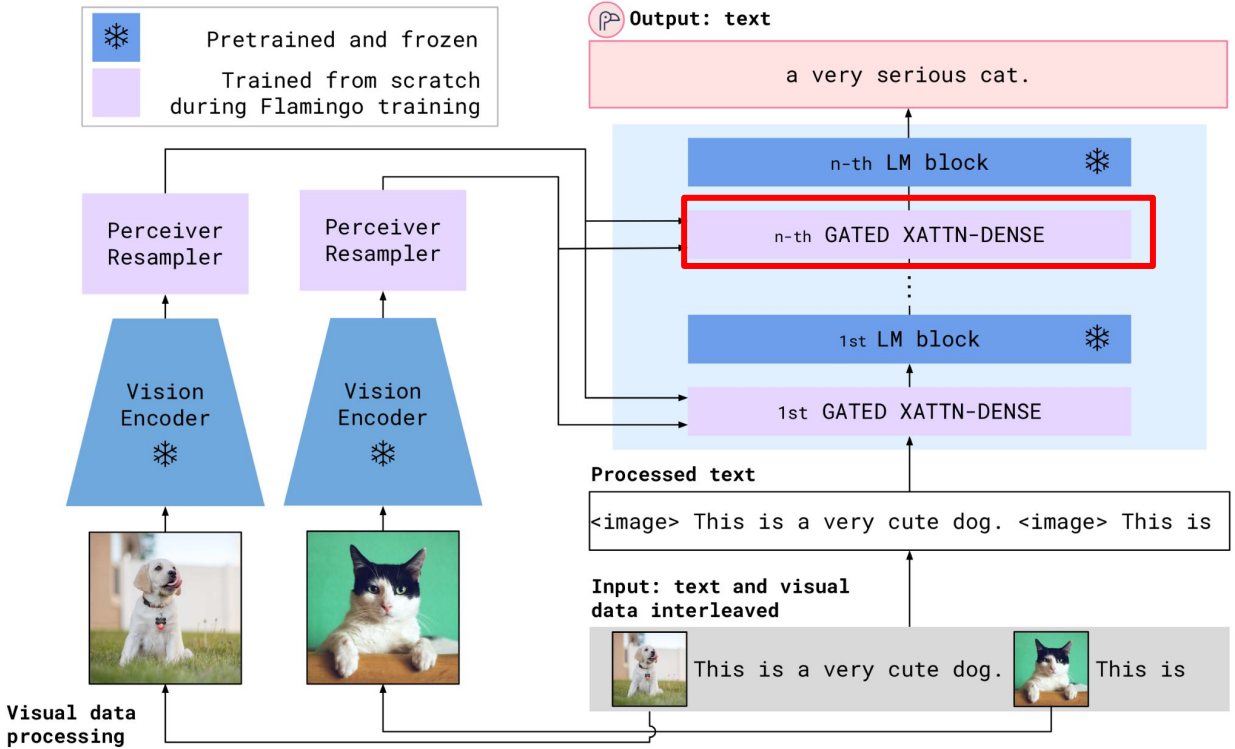
Flamingo full architecture

Learned method of down-sampling image/video representations



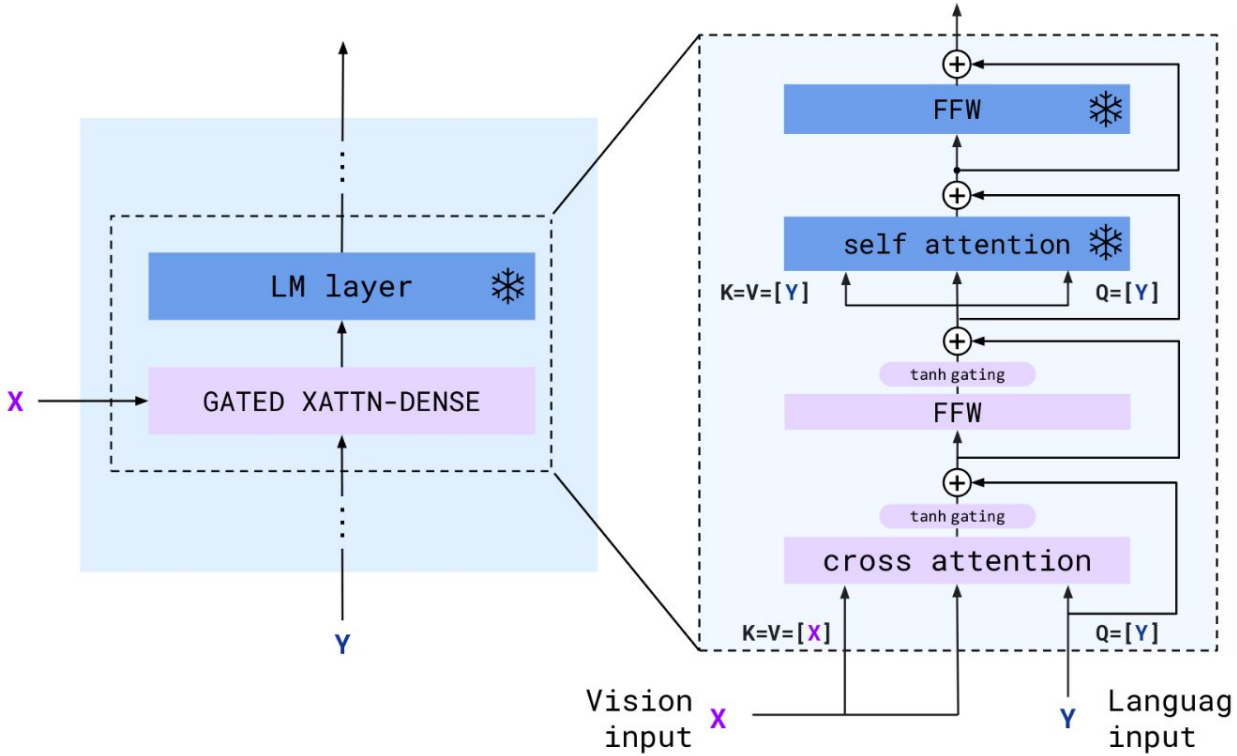
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo full architecture



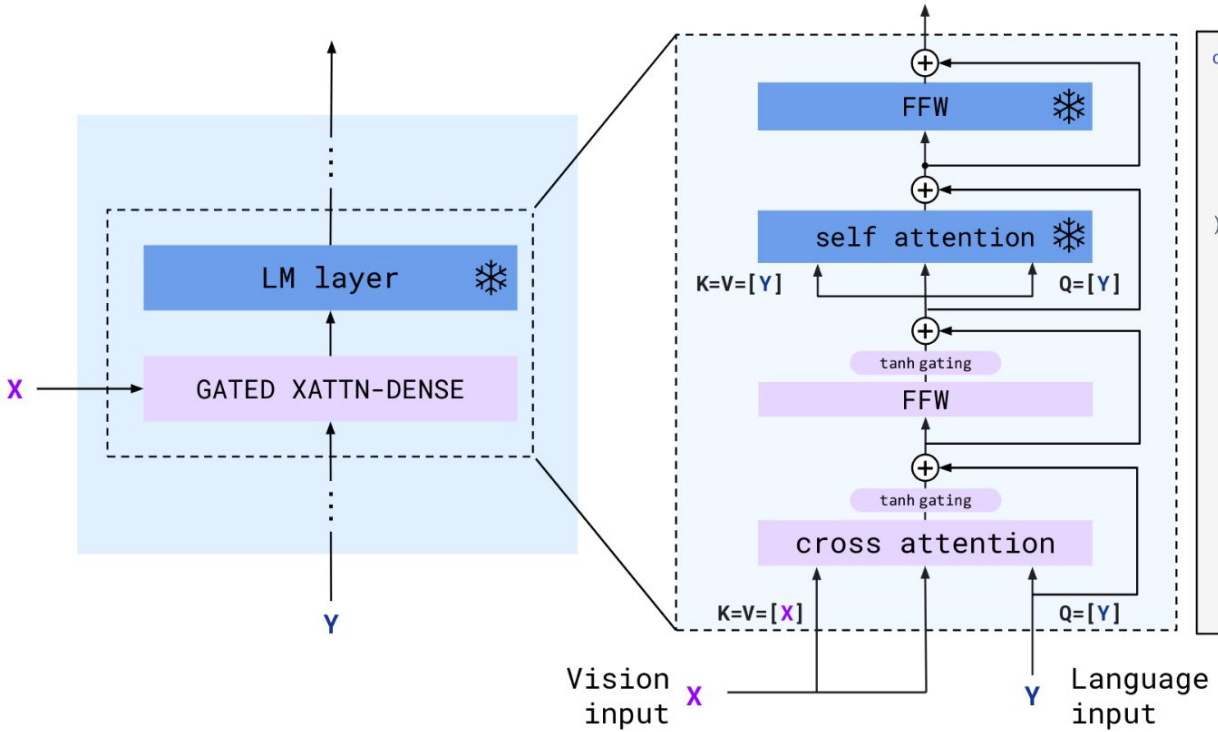
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

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 masked attention

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

Flamingo masked attention

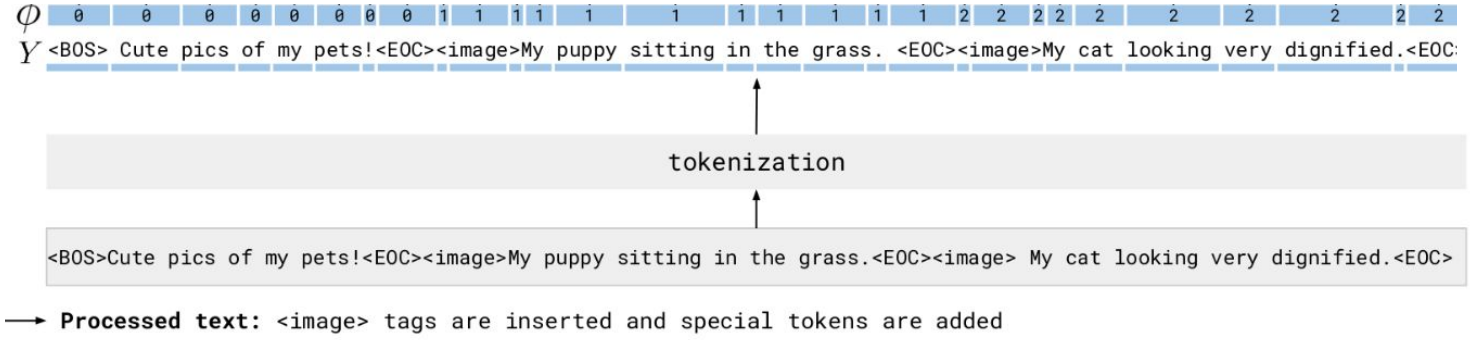
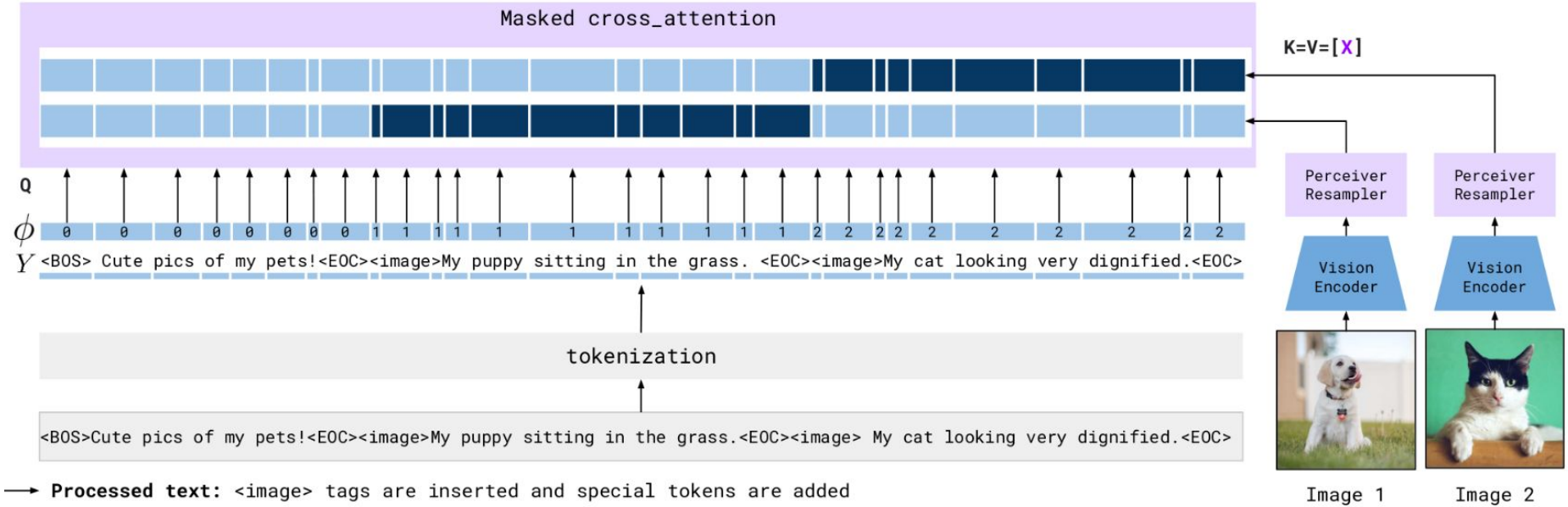


Image 1 Image 2

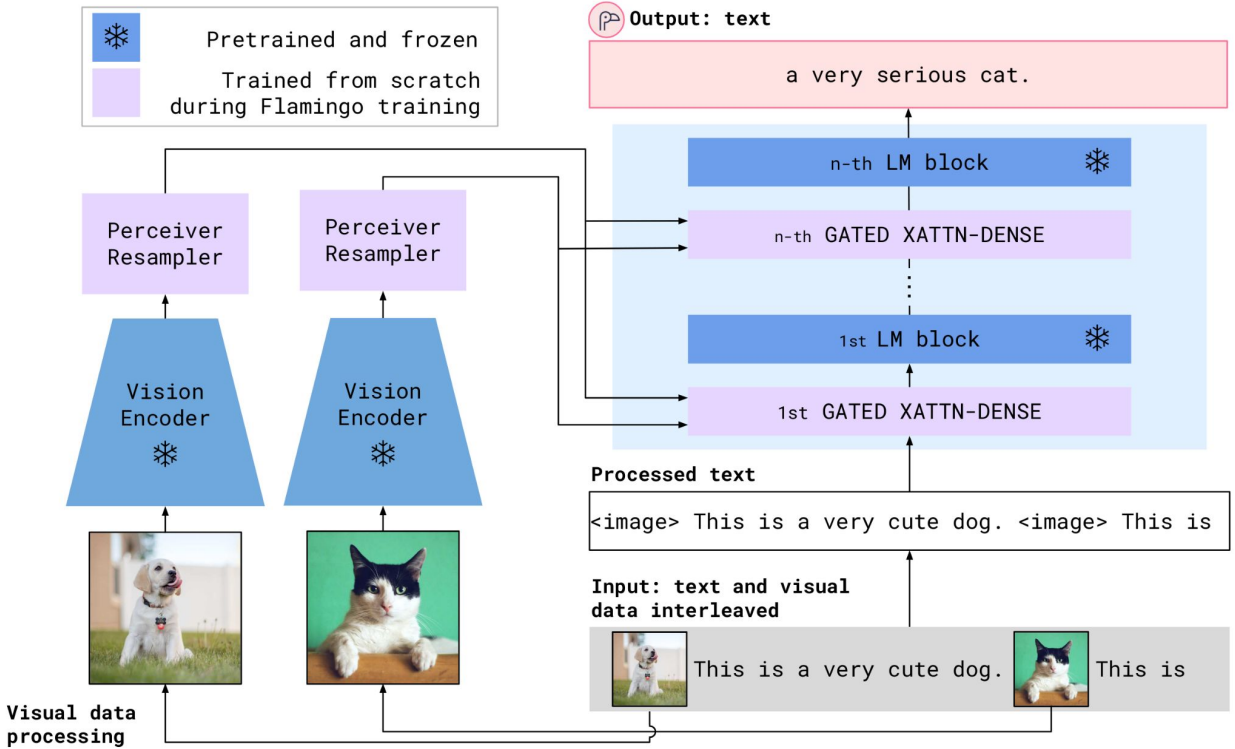
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo masked attention



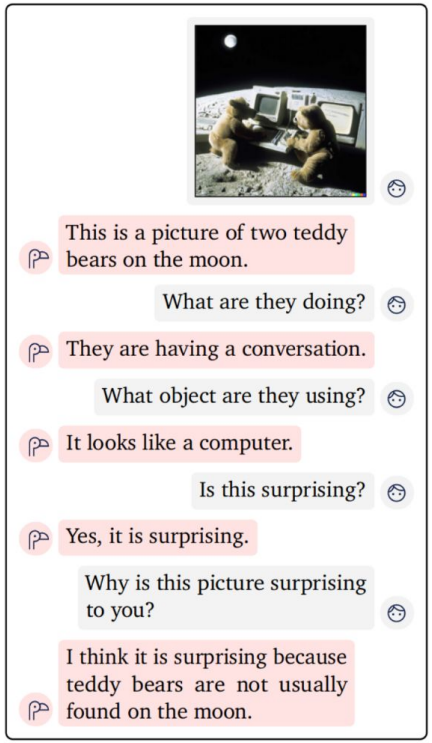
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

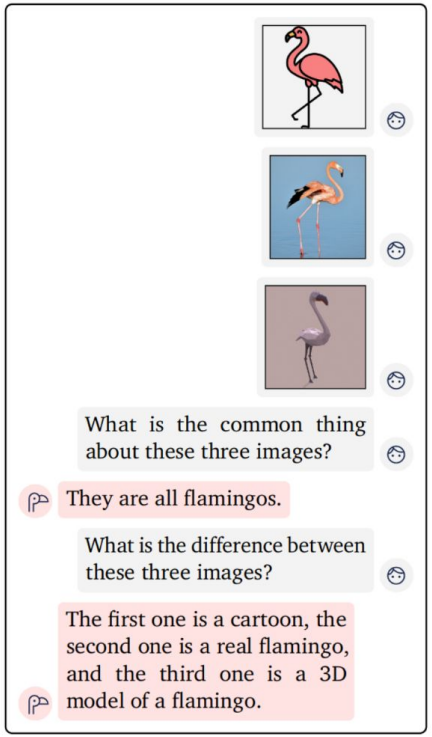
Flamingo full architecture

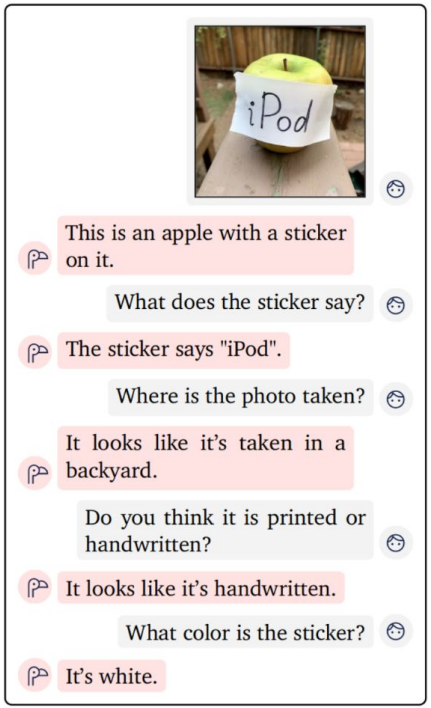


Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flamingo results



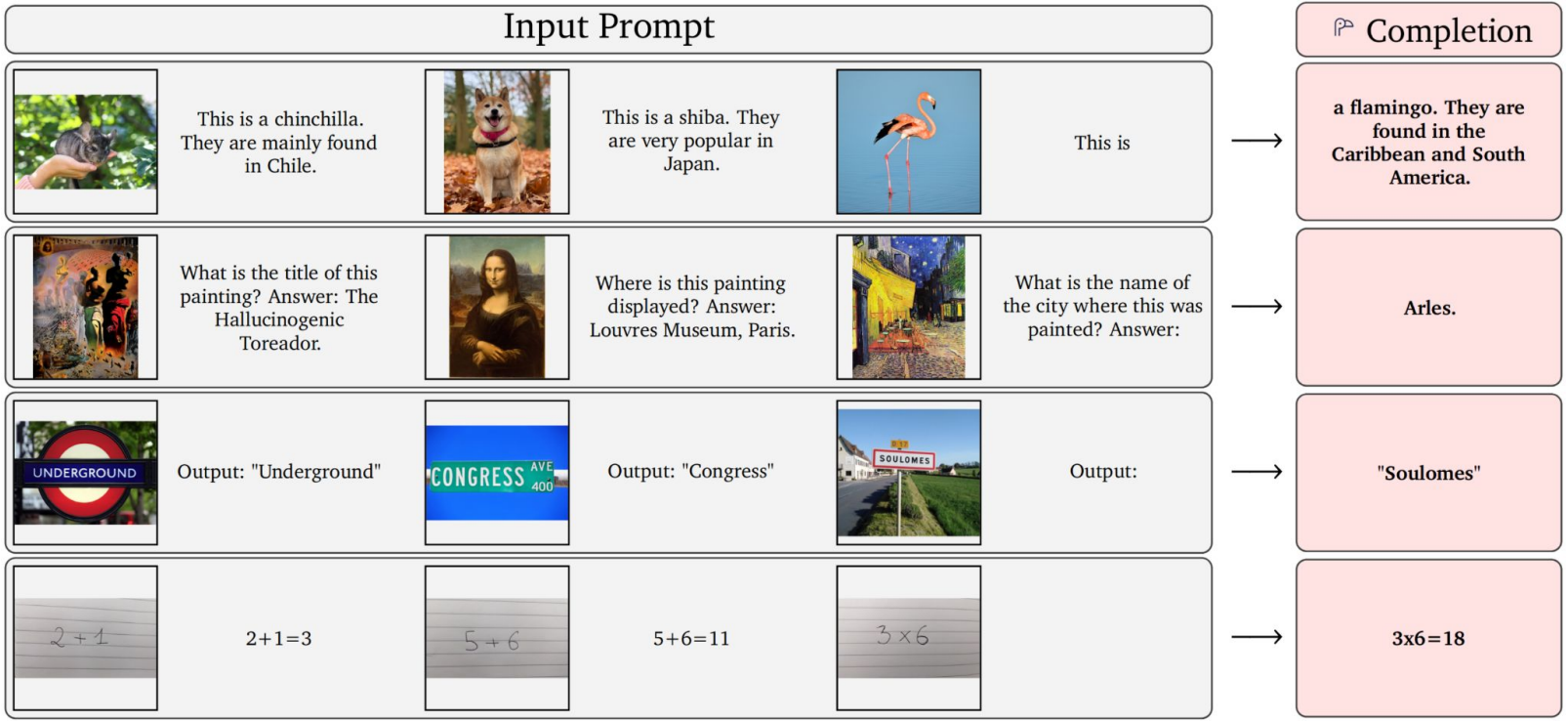




Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

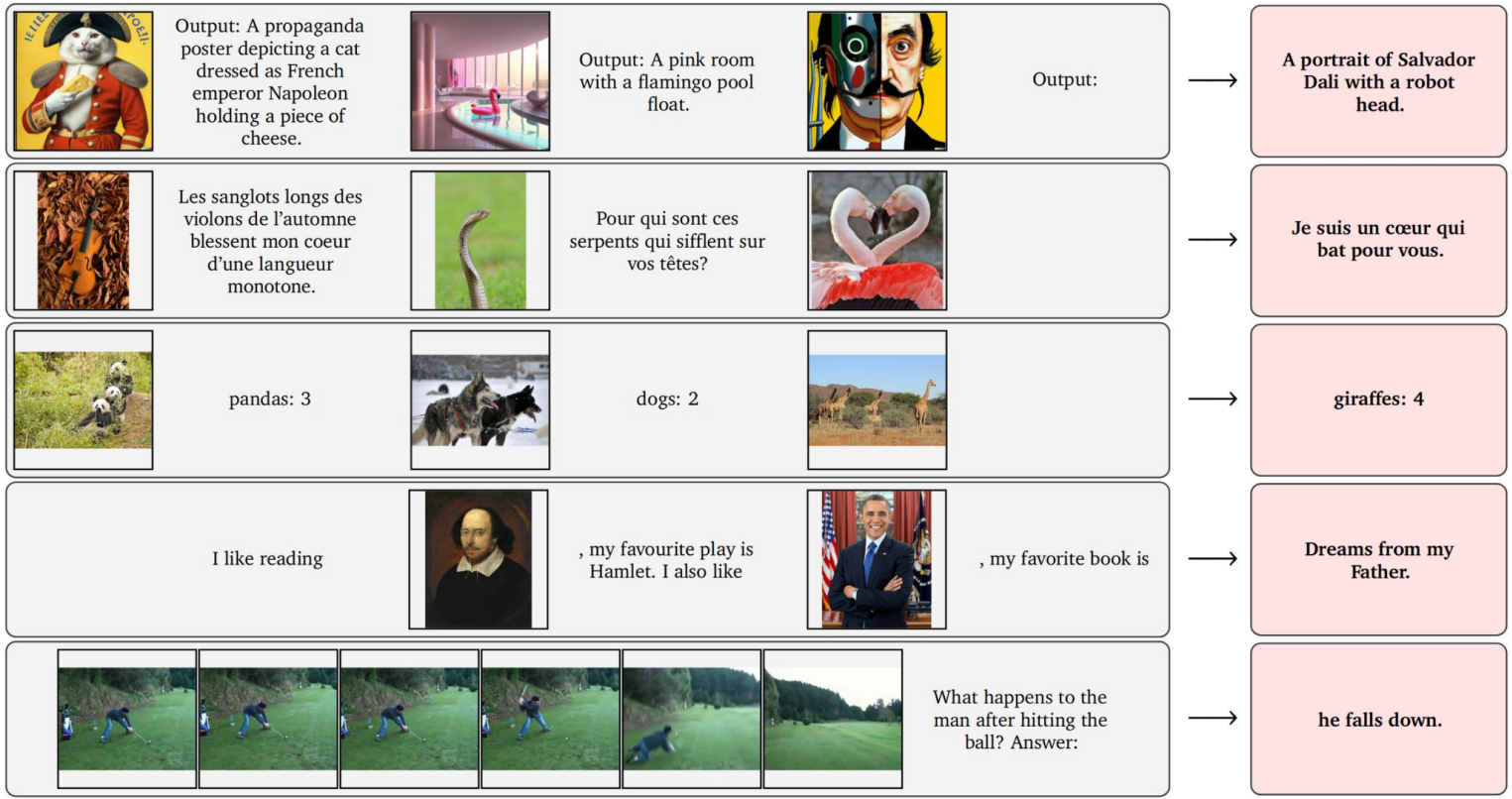
Flamingo results

What is this type of learning called?



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.

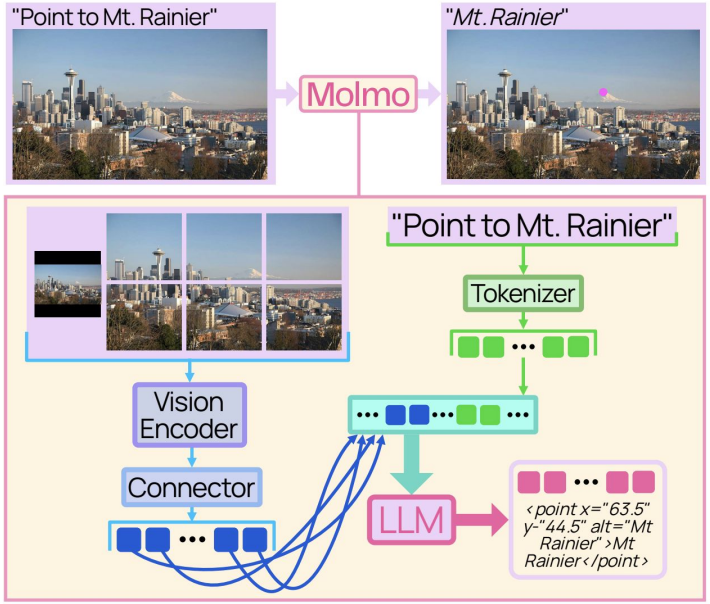
Results: zero & few shot

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

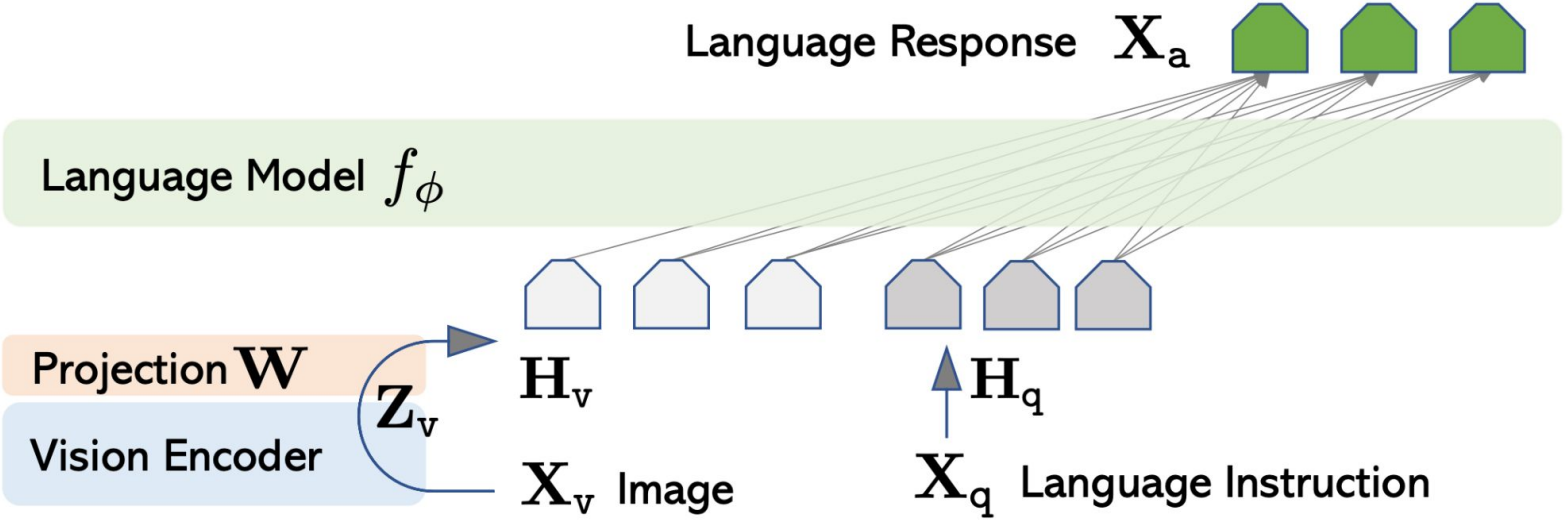
Method	FT	Shot	OKVQA	VQAv2	COCO	MSVDQA	VATEX	VizWiz	Flick30K	MSRVTTQA	iVQA	YouCook2	STAR	VisDial	TextVQA	NextQA	HatefulMemes	RareAct
Zero/Few shot SOTA	✗	(X)	[39] 43.3 (16)	[124] 38.2 (4)	[134] 32.2 (0)	[64] 35.2 (0)	-	-	-	[64] 19.2 (0)	[145] 12.2 (0)	-	[153] 39.4 (0)	[87] 11.6 (0)	-	-	[94] 66.1 (0)	[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)	-

Later multimodal LMs - Molmo



Llava: https://arxiv.org/pdf/2304.08485
Molmo: https://arxiv.org/pdf/2409.17146

Later multimodal LMs - Llava



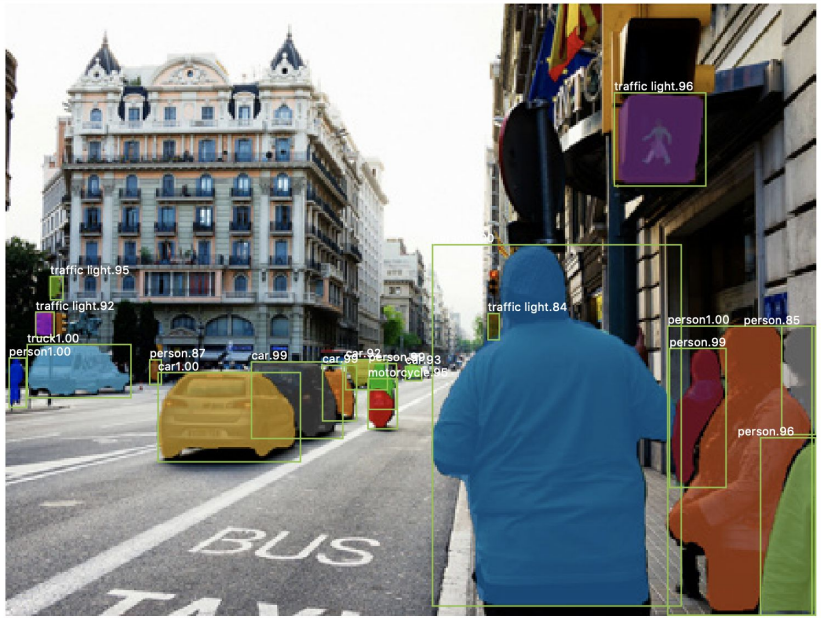
Llava: https://arxiv.org/pdf/2304.08485
Molmo: https://arxiv.org/pdf/2409.17146

Foundation Models

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

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

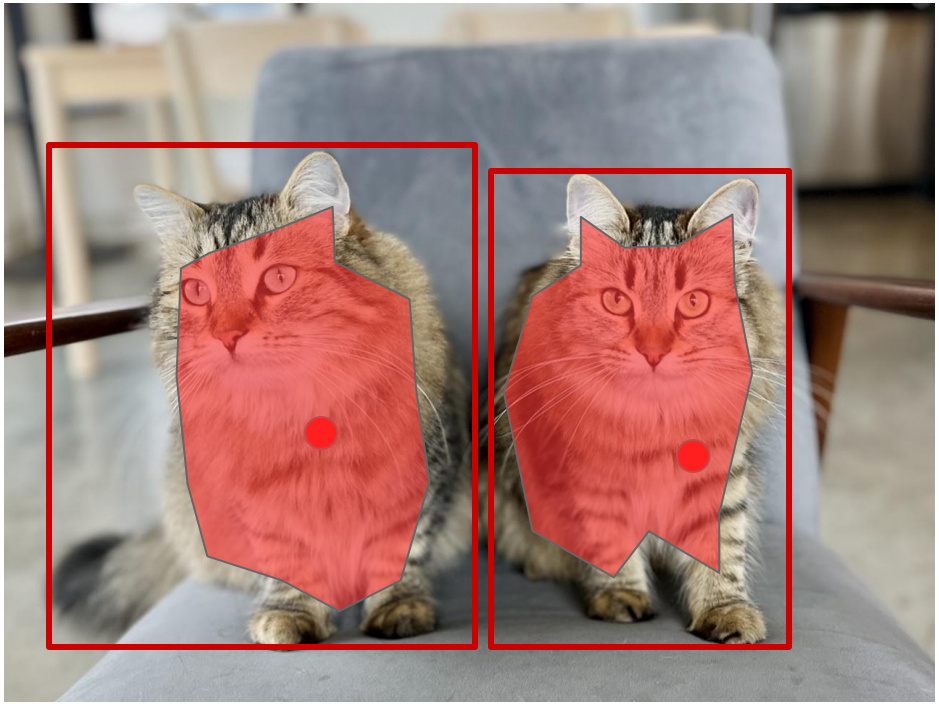
How to get this?

Model outputs mask of any objects that the user cares about

How to know this?

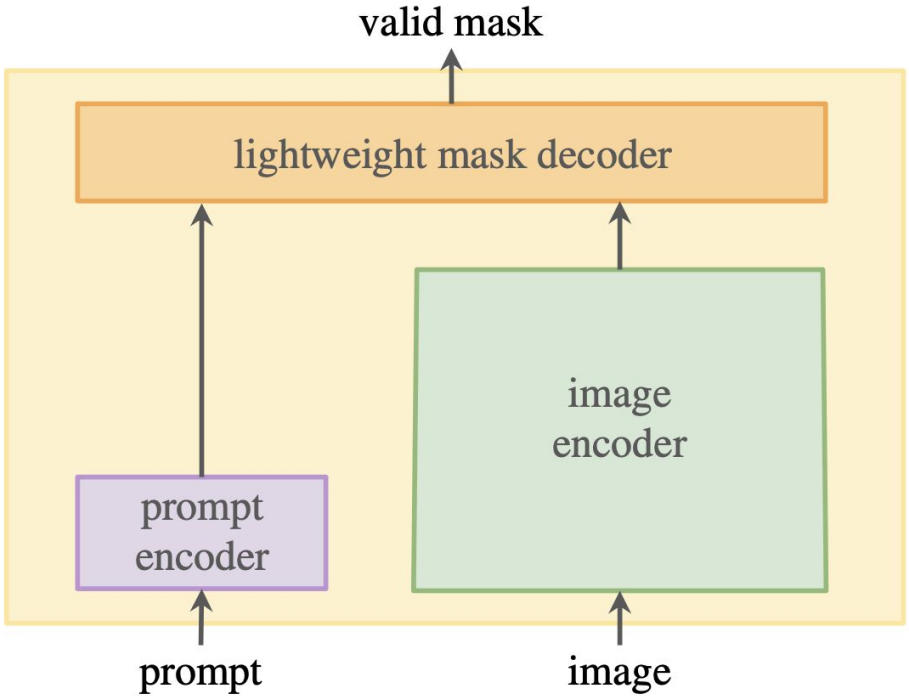
Images: Kirillov et al. Segment Anything. 2023.

How to know what to mask?



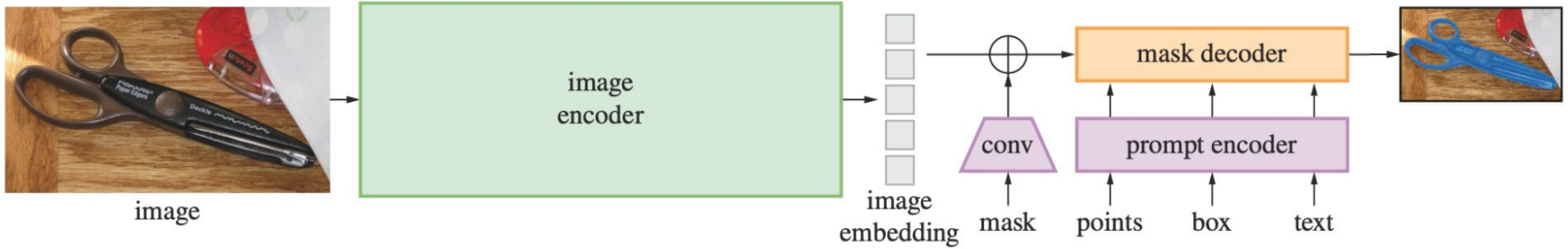
“Cats”

Basic SAM Architecture



Images: Kirillov et al. Segment Anything. 2023.

SAM Architecture



Images: Kirillov et al. Segment Anything. 2023.

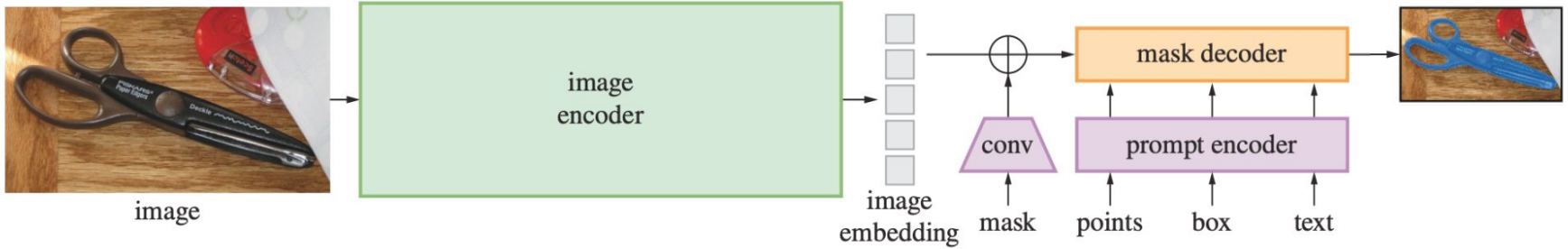
Ambiguity in correct prompt

Images: Kirillov et al. Segment Anything. 2023.

Ambiguity in correct prompt

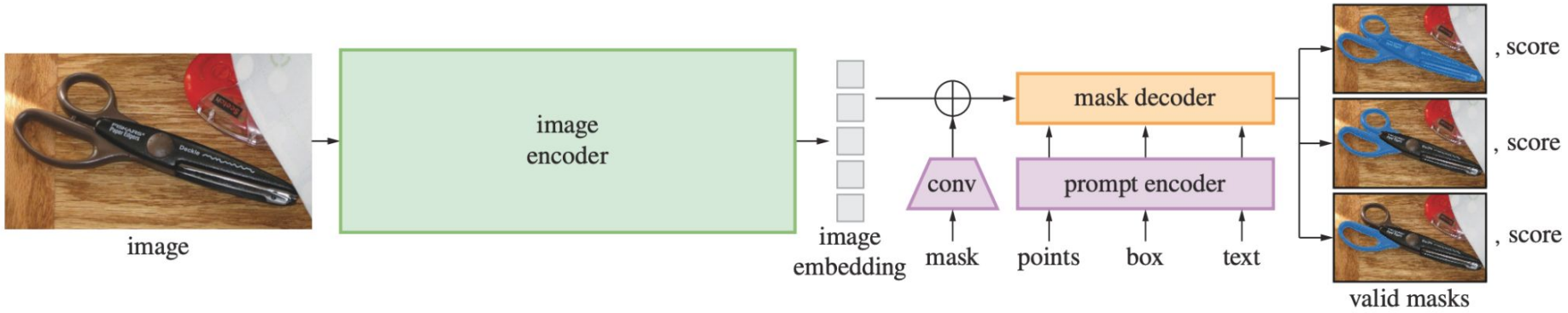
Images: Kirillov et al. Segment Anything. 2023.

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

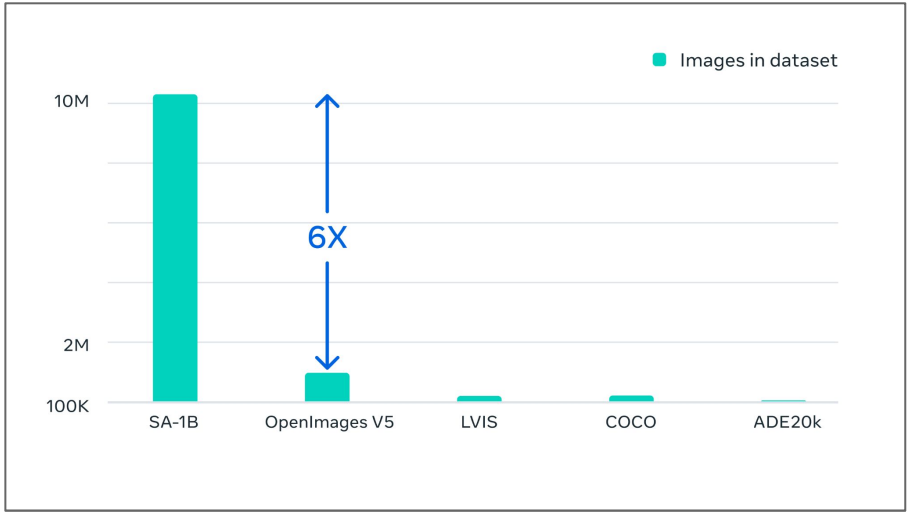
How to get this?

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)



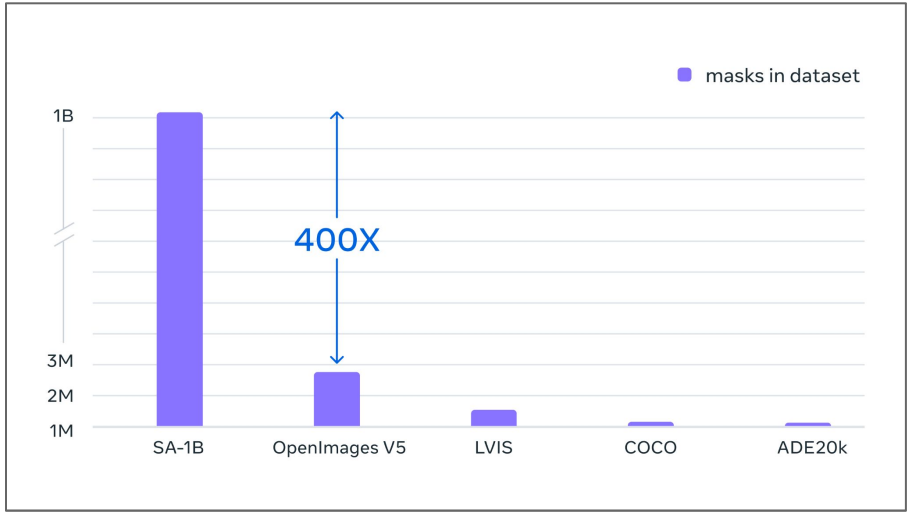
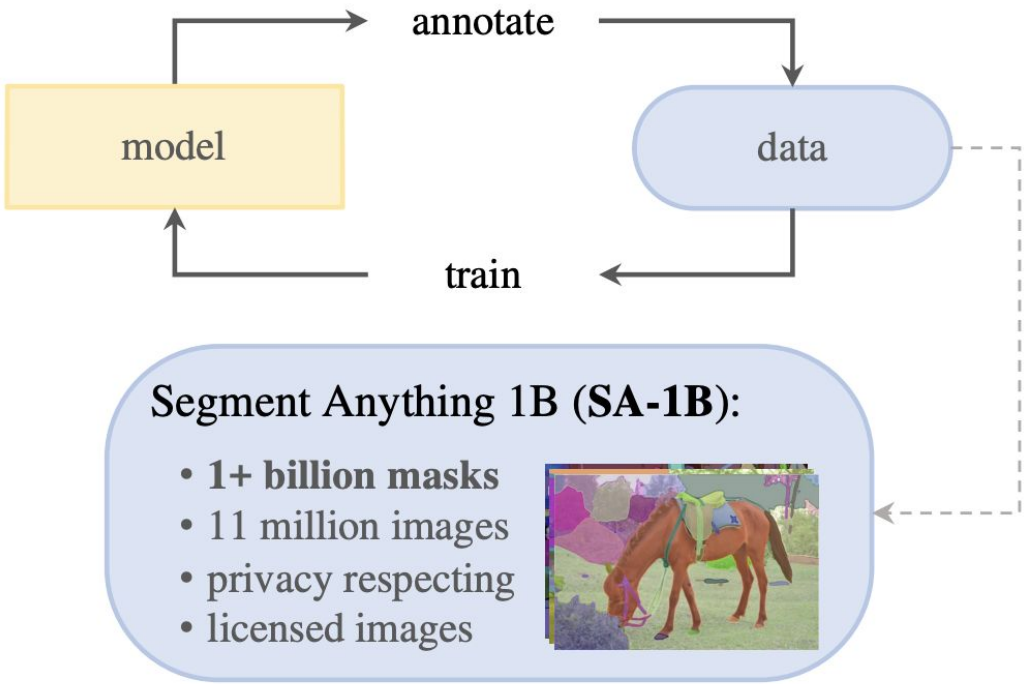


Image Source: https://segment-anything.com/

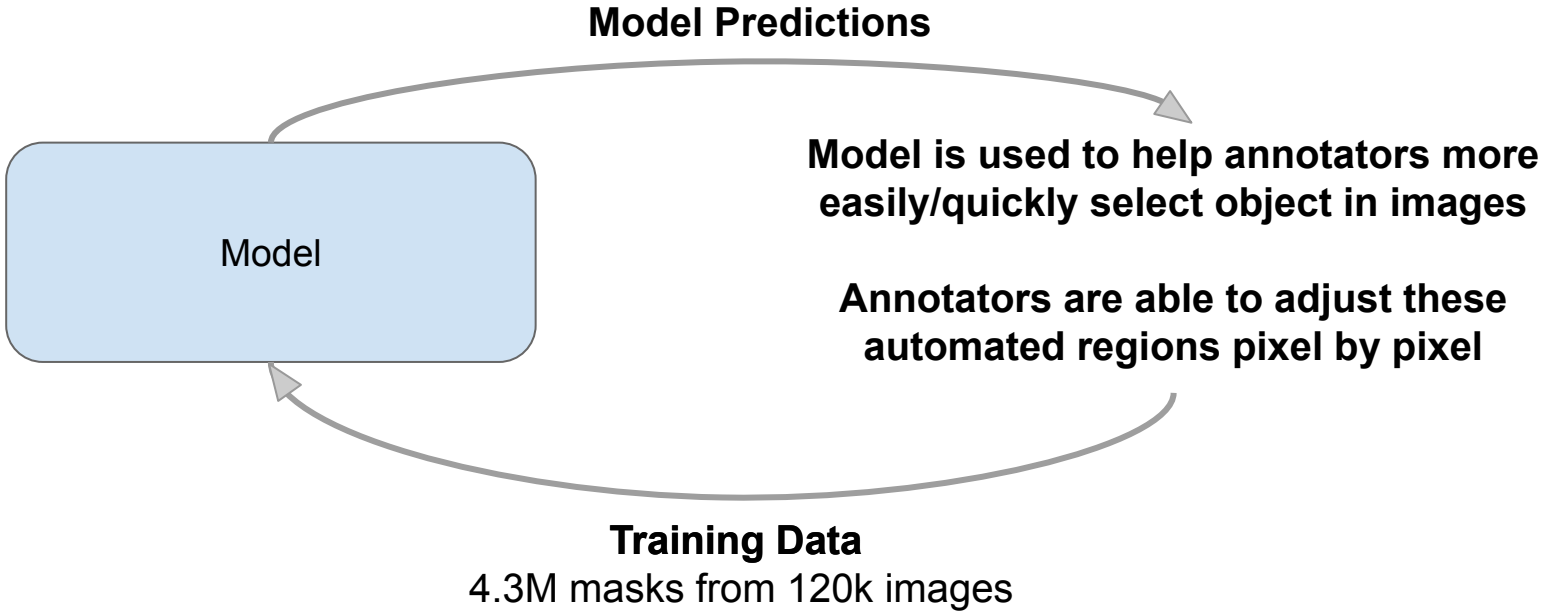
Segment Anything Model (SAM)



Images: Kirillov et al. Segment Anything. 2023.

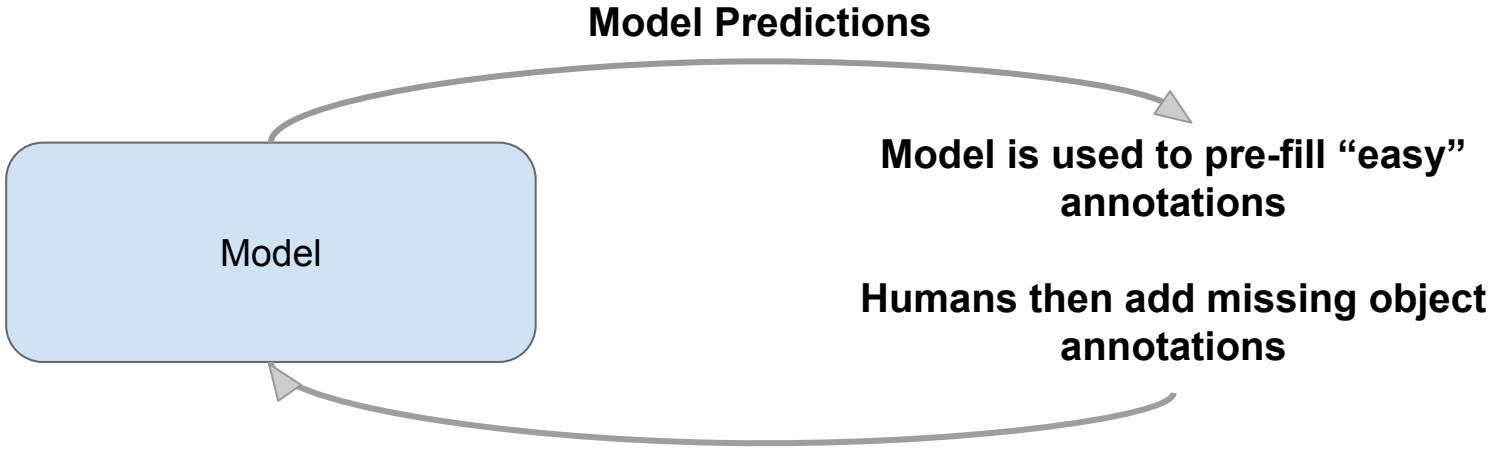
Segment Anything Model (SAM)

Assisted-manual stage



Segment Anything Model (SAM)

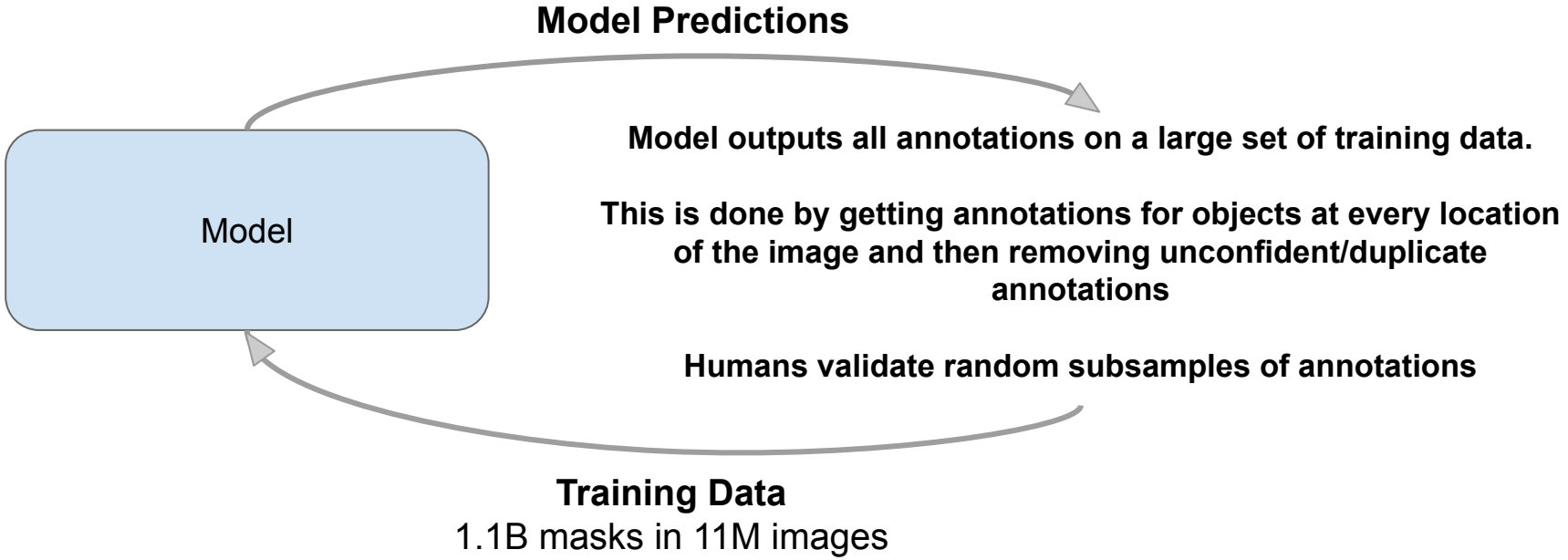
Semi-automatic stage



Training Data
5.9M masks in 180k images

Segment Anything Model (SAM)

Fully automatic stage



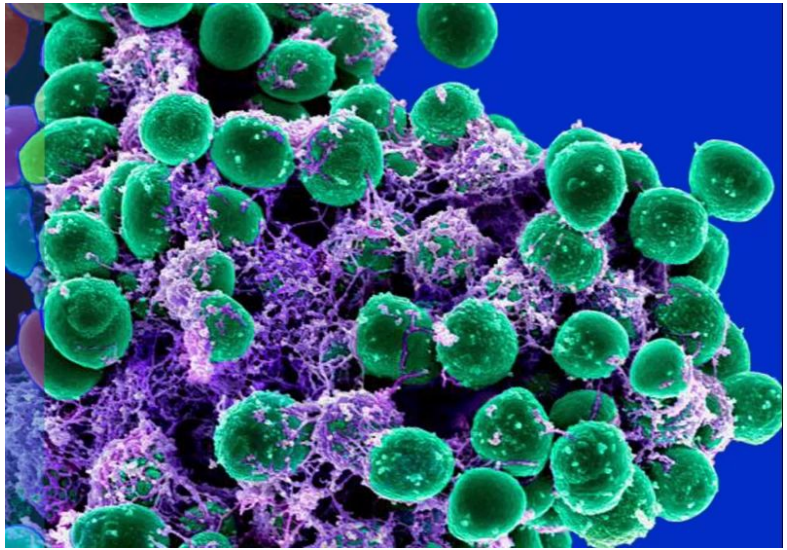
SAM Results

Image Source: Kirillov et al. Segment Anything. 2023

SAM Results

Image Source: Kirillov et al. Segment Anything. 2023

Zero-Shot with SAM



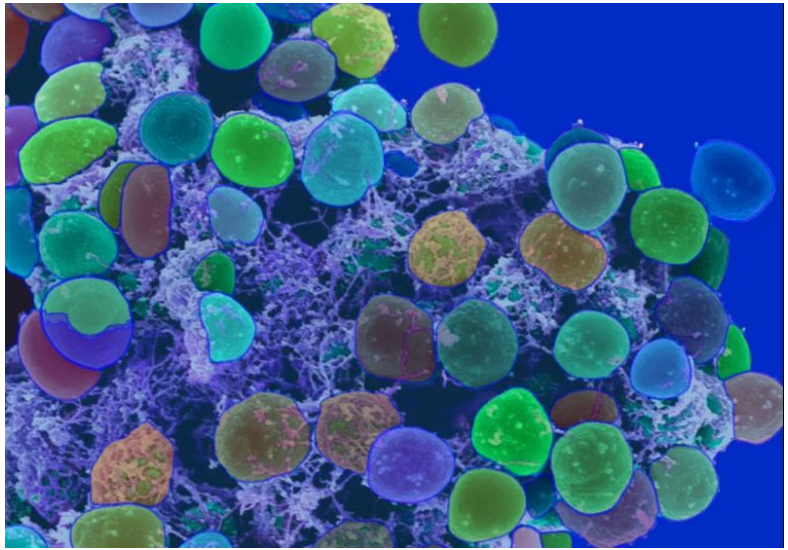


Image Source: https://segment-anything.com/

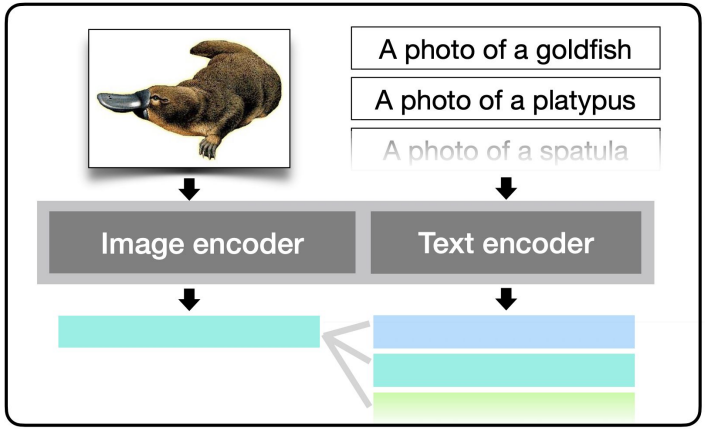
Zero-Shot with SAM

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

Foundation Models

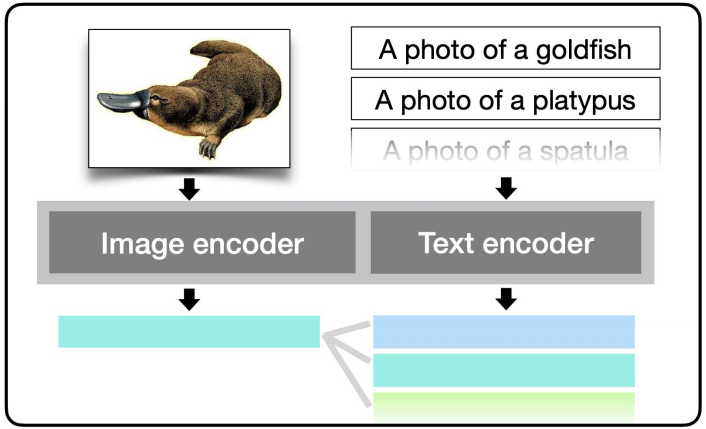
Language	Classification	LM + Vision	And More!	Chaining
ELMo	**CLIP**	**Flamingo**	**Segment Anything**	**LMs + CLIP**
BERT	**CoCa**	GPT-4V	Whisper	**Visual Programming**
GPT		Gemini	Dalle	Tool use
T5			Stable Diffusion	
			Imagen	

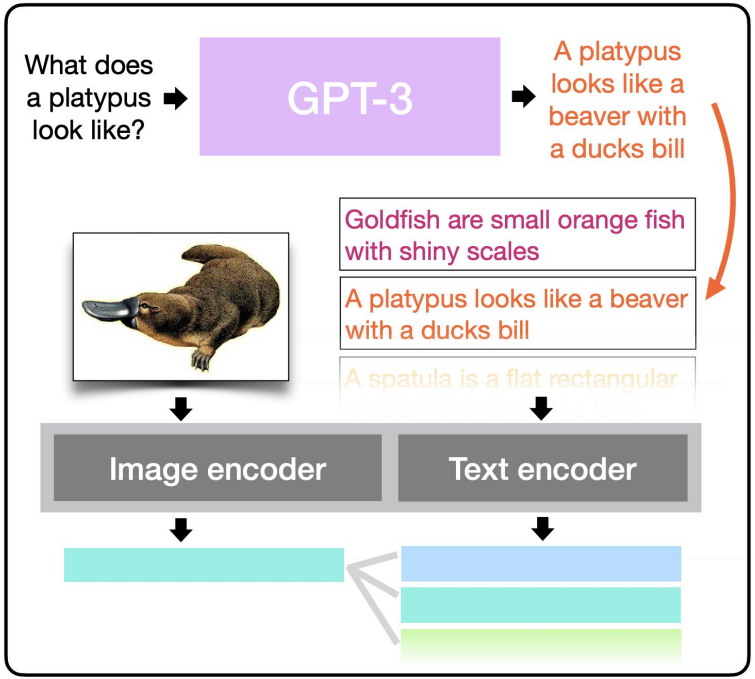
CuPL (CUstomized Prompts via Language models)



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

CuPL (CUstomized Prompts via Language models)

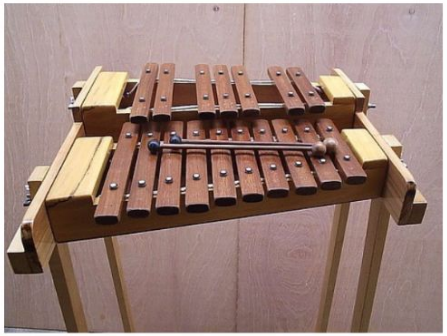
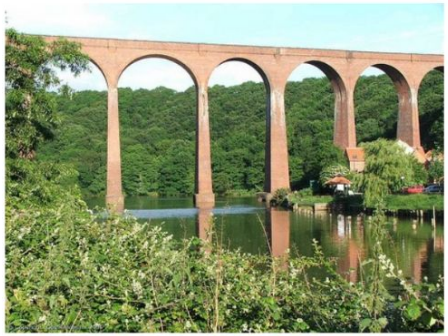




Pratt et al “What does a platypus look like? Generating customized prompts for zero-shot image classification”. 2023.

CuPL (CUstomized Prompts via Language models)

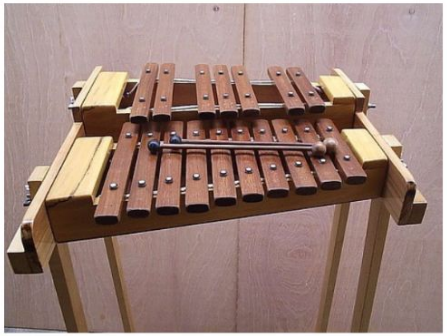
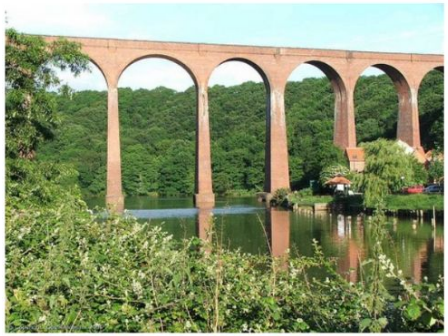
A photo of a marimba
A photo of a viaduct
A photo of a papillon
A photo of a lorikeet


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

CuPL (CUstomized Prompts via Language models)

"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."

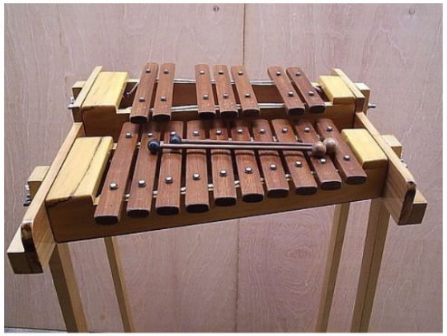




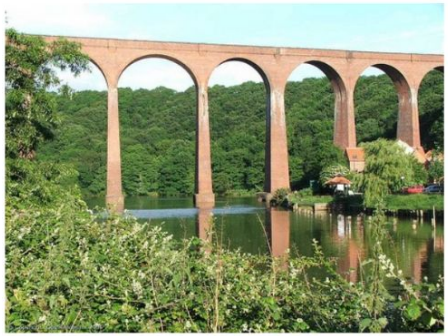

Pratt et al "What does a platypus look like? Generating customized prompts for zero-shot image classification". 2023.

CuPL (CUstomized Prompts via Language models)

"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."


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)

LLM-prompts:

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

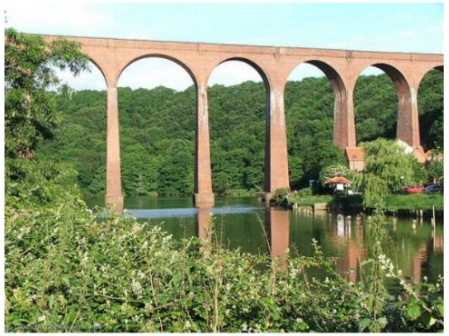
GPT-3

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	mean
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	73.43
# hw	80	8	8	2	1	2	1	34	1	48	28	18	18	18	1	
CuPL (full)	76.69	61.70	77.63	73.31	93.36	36.11	93.81	93.45	79.67	78.36	60.63	71.69	95.84	78.57	51.11	74.80
Δ std	+1.15	+6.50	+0.10	+4.00	+0.28	+3.23	+0.48	+0.21	+1.14	+0.91	+0.56	+0.59	+0.25	+0.31	+0.63	
# hw	5	6	9	3	3	2	2	3	2	5	4	5	3	4	3	

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

VisProg (visual programming)

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:

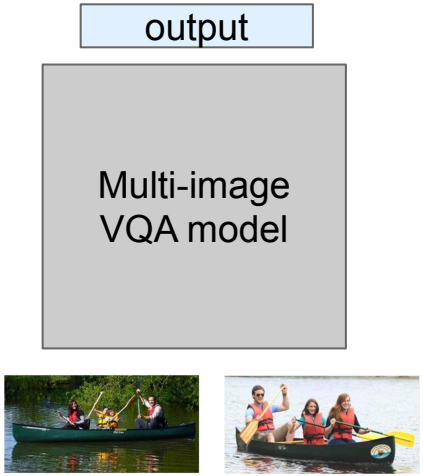
RIGHT:

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

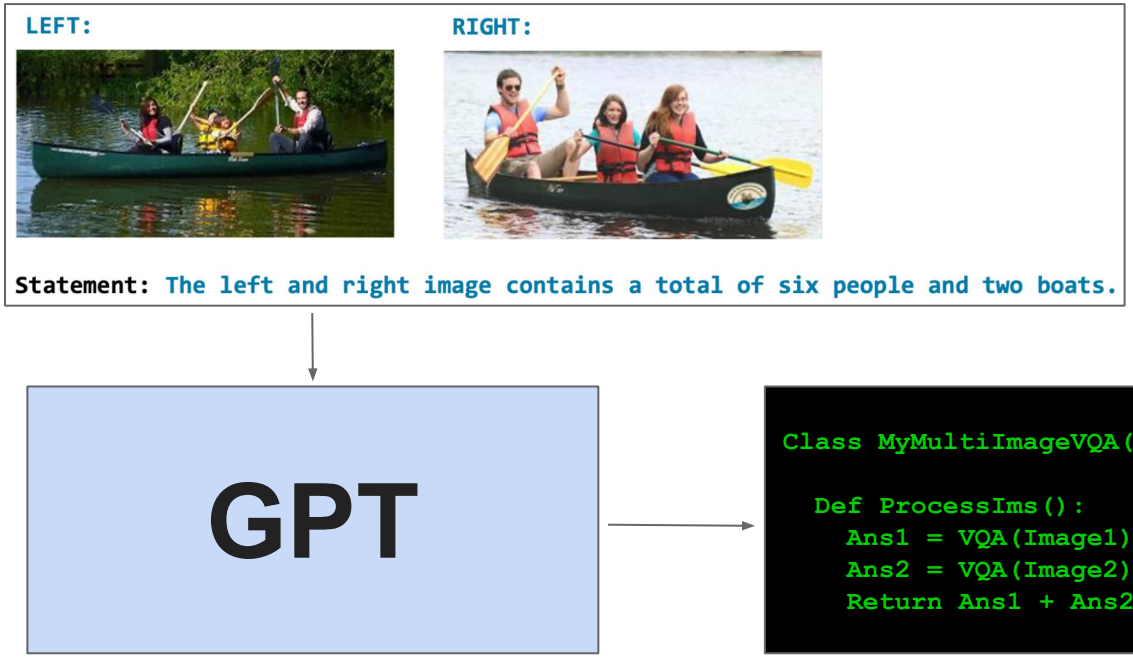
```
Class MyMultiImageVQA():

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

General to 2 images now, but not beyond that

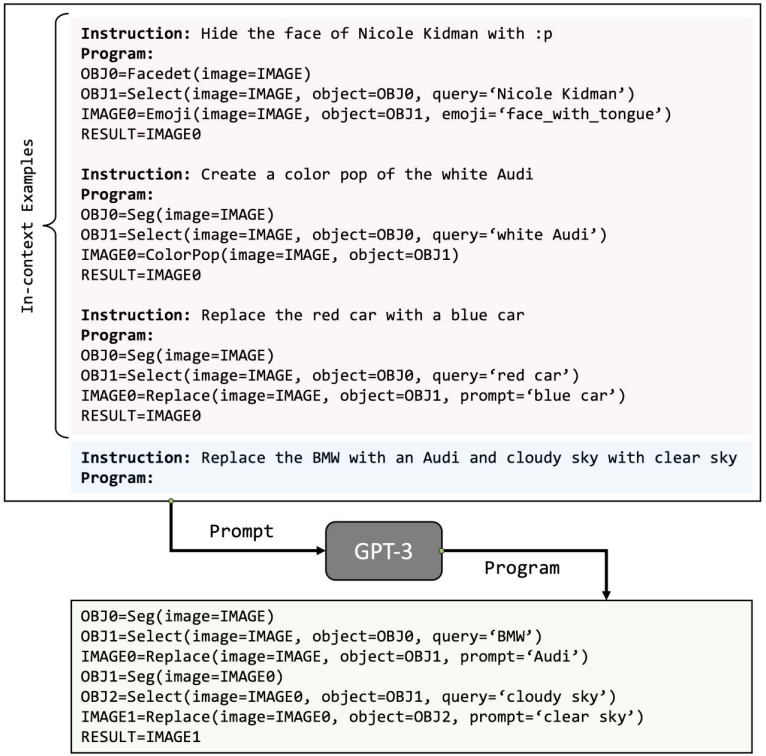
Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

VisProg (visual programming)



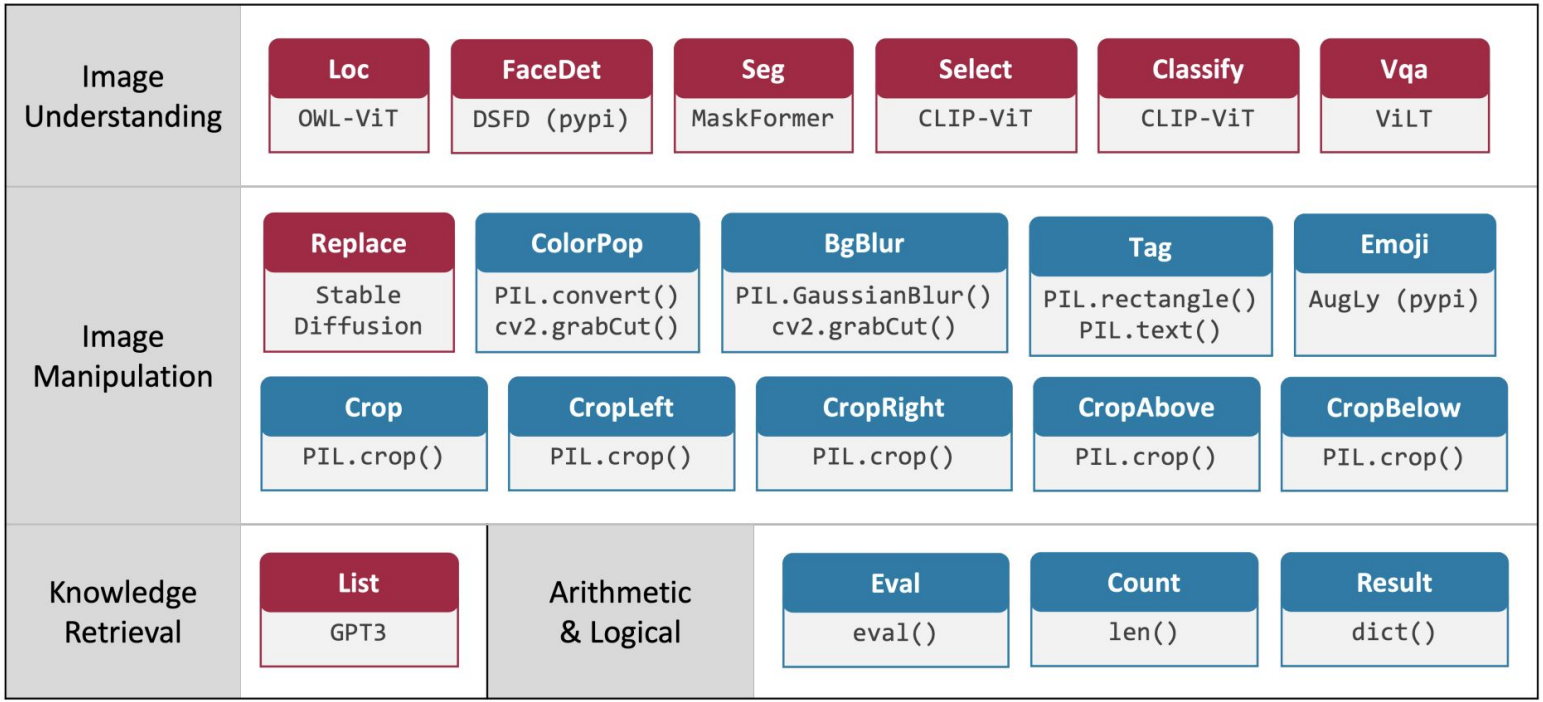
Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

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

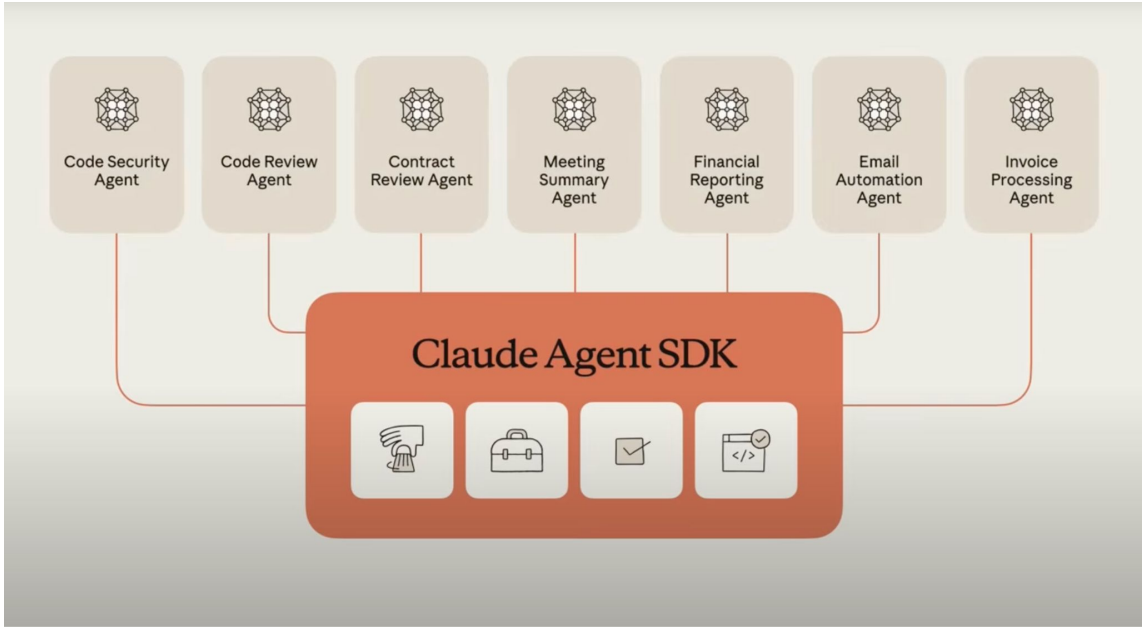
Instruction: Replace desert with lush green grass

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
```

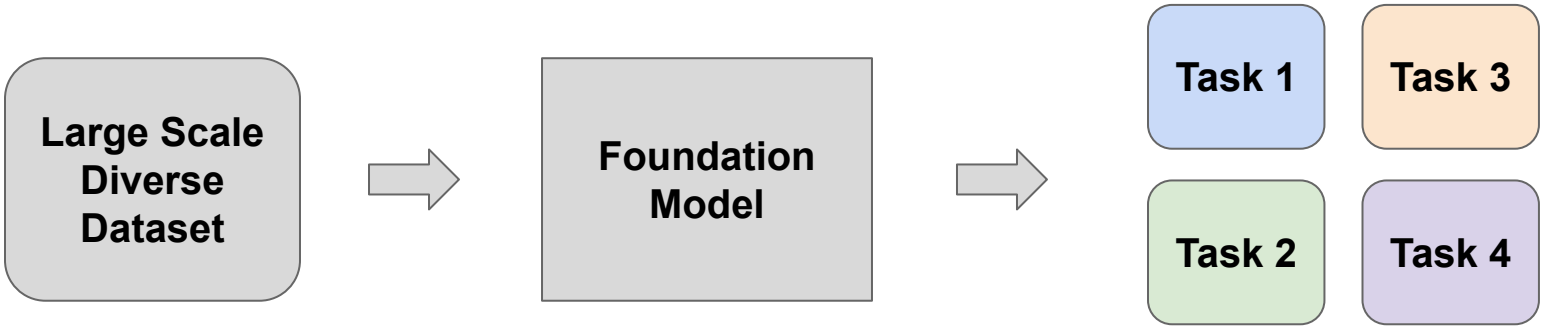
Gupta et al "Visual Programming: Compositional visual reasoning without training". 2023.

Tool Use! Agents! Code!

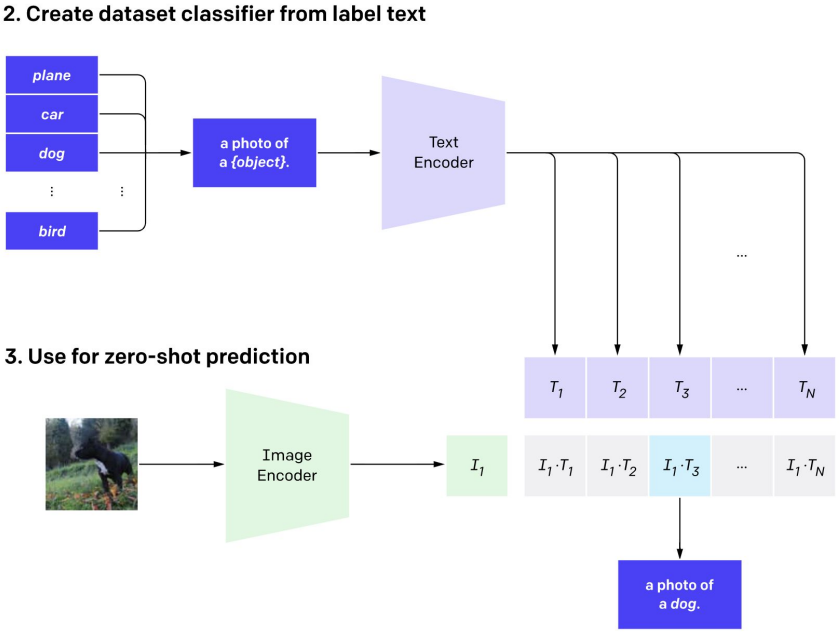


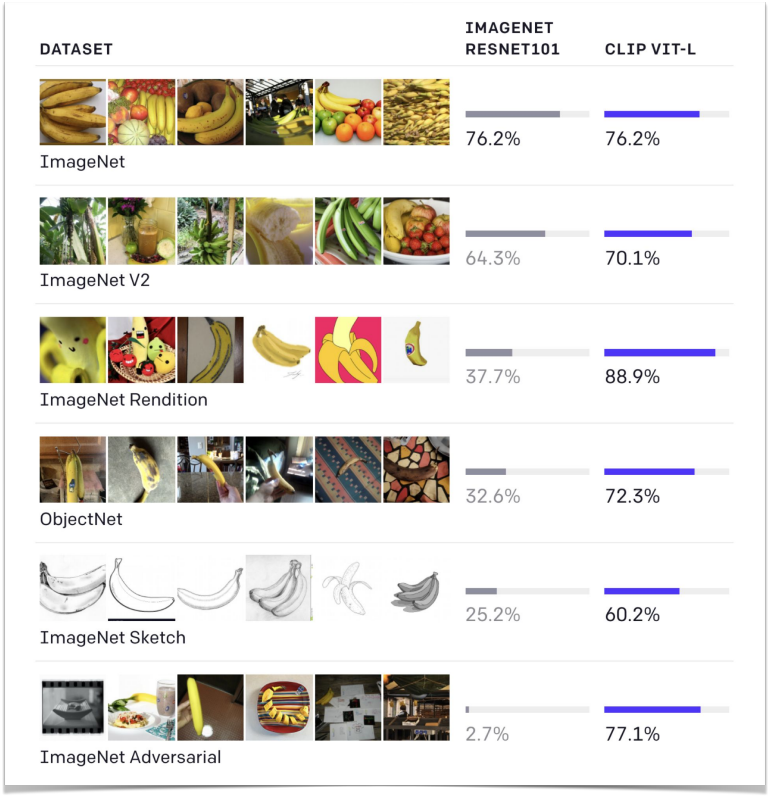
https://www.anthropic.com/news/enabling-claude-code-to-work-more-autonomously

Summary

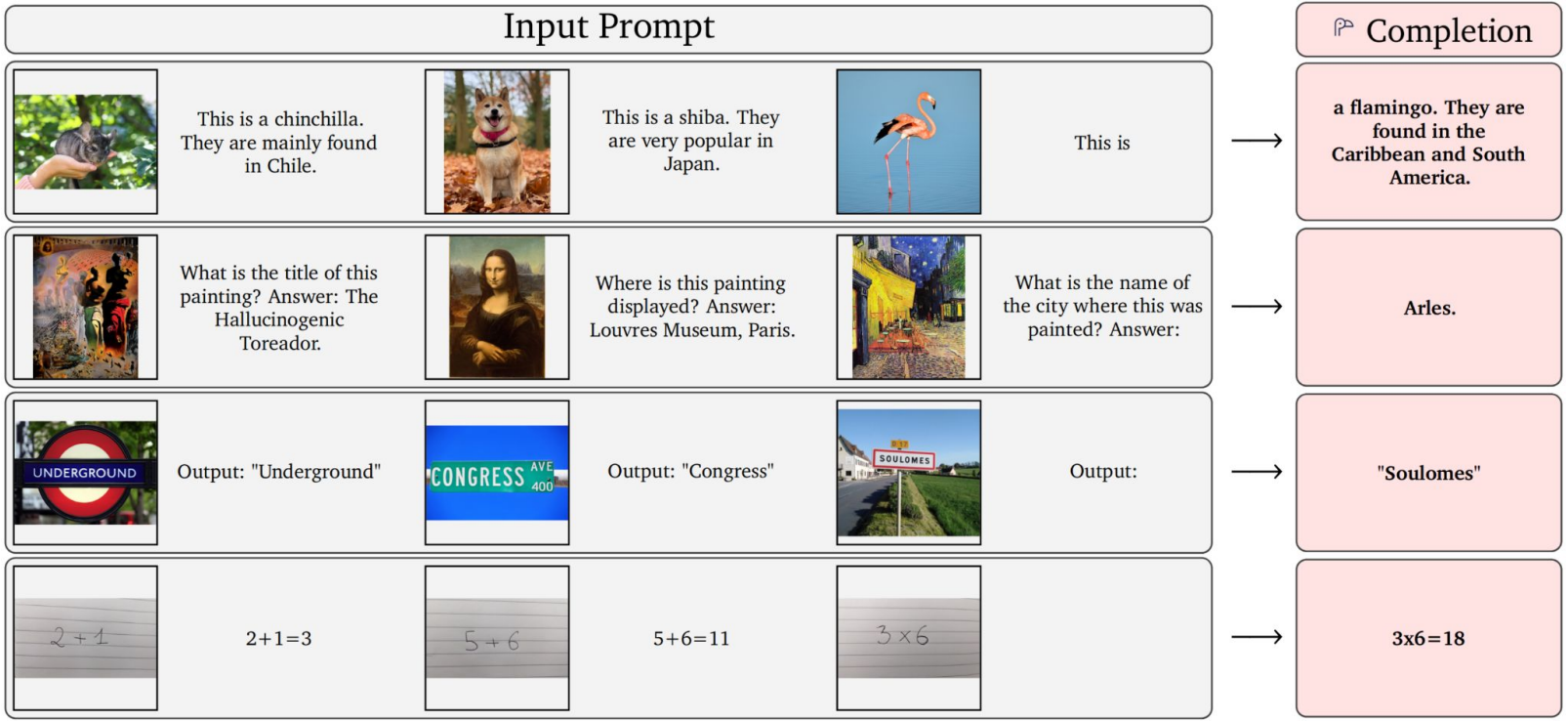

Large Scale Diverse Dataset
Foundation Model
Task 1
Task 3
Task 2
Task 4

Summary


2. Create dataset classifier from label text
plane
car
dog
bird
a photo of a {object}.
Text Encoder
3. Use for zero-shot prediction
Image Encoder
I_1
T_1 T_2 T_3 ... T_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
a photo of a dog.


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.
UNDERGROUND
Output: "Underground"
CONGRESS AVE 400
Output: "Congress"
SOULOMES
Output:
"Soulomes"
2+1
2+1=3
5+6
5+6=11
3x6
3x6=18

Summary