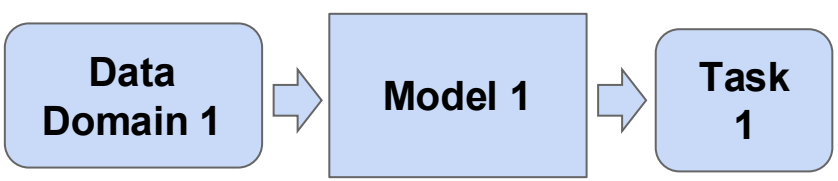
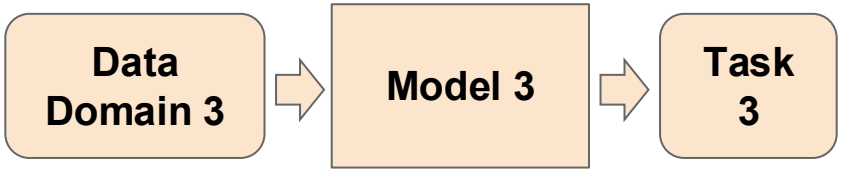
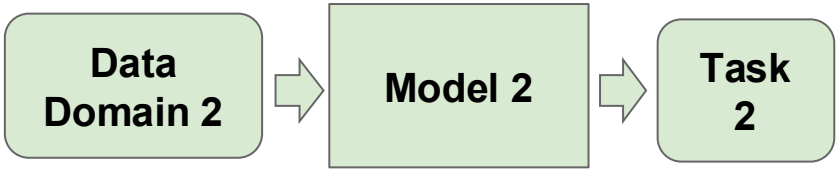
Lecture 16:
Vision + Language
(and Foundation Models)

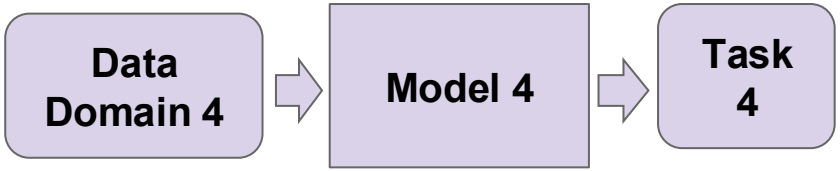
How have we been thinking about models in this class so far?

Train a *specialized* model for *each* task



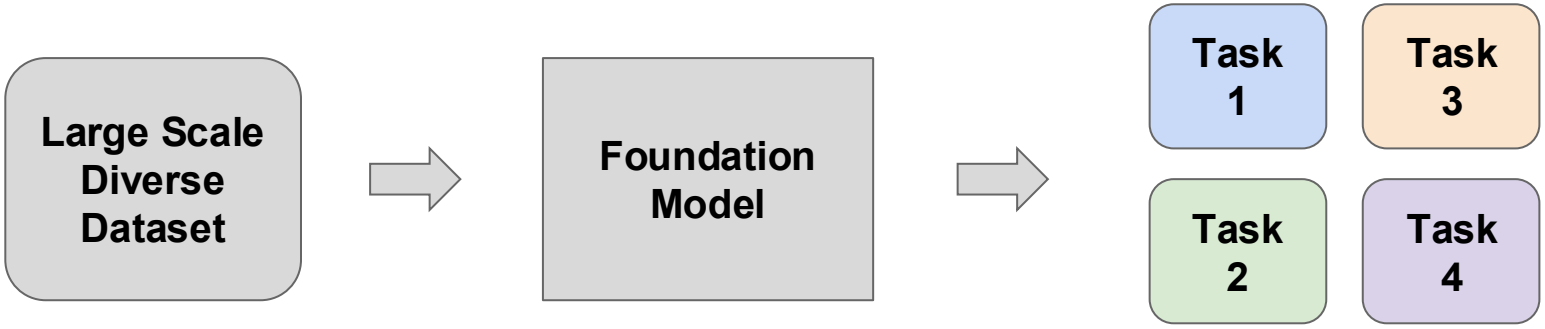






Another paradigm: Foundation Models

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



How do identify a model as a Foundation?

Always see with foundation models:

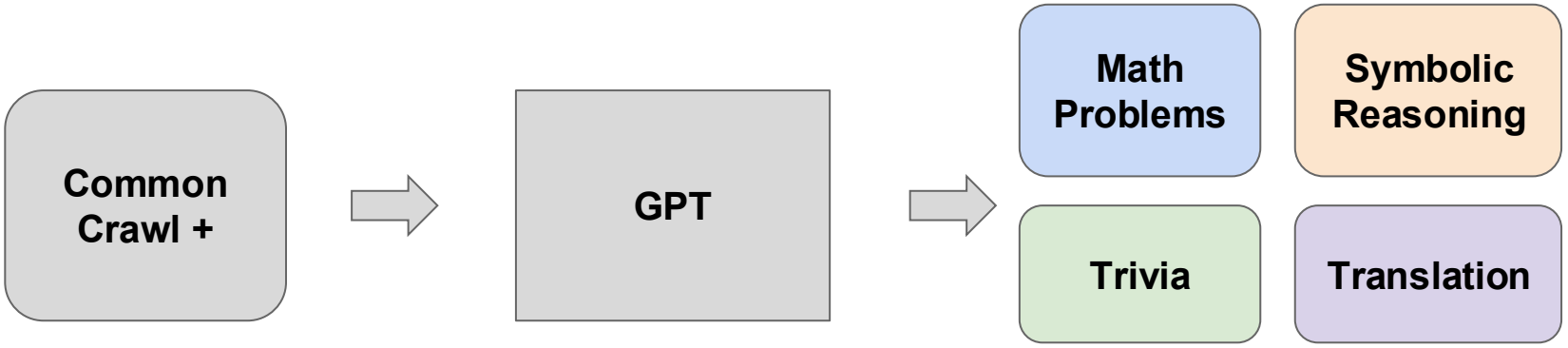
- general /robust to many different tasks

Often see with foundation models:

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

Foundation Models

Language



There are many classes of Foundation Models

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

There are many classes of Foundation Models

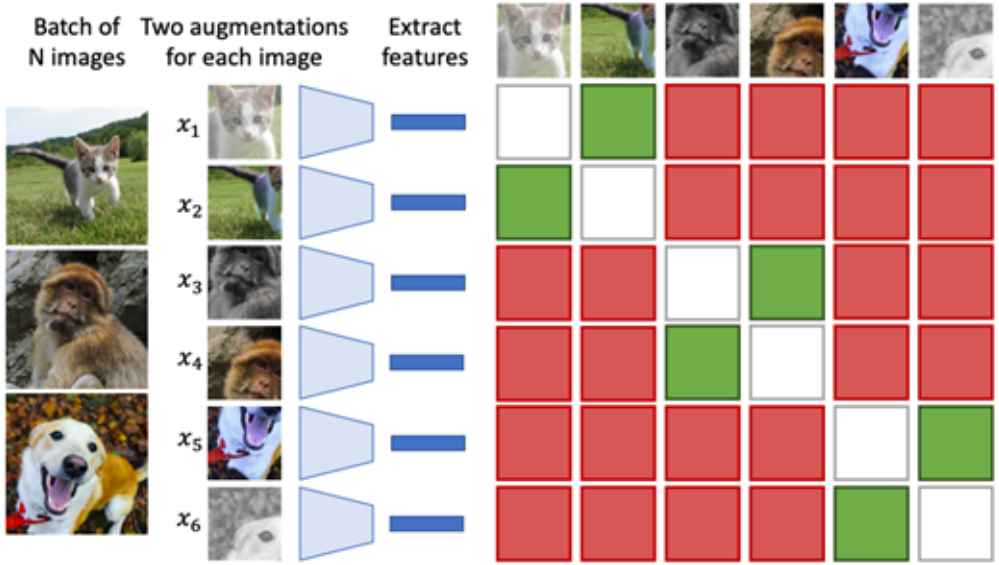
Language	Classification	LM + Vision	Chaining	And More!
ELMo	CLIP	LLaVA	LMs + CLIP	Segment Anything
BERT	CoCa	Flamingo	Visual Programming	Whisper
GPT		GPT		Dalle
T5		Gemini		Stable Diffusion
		Qwen		Imagen

The main topics discussed in lecture today

Let's start with the foundation models for classification

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

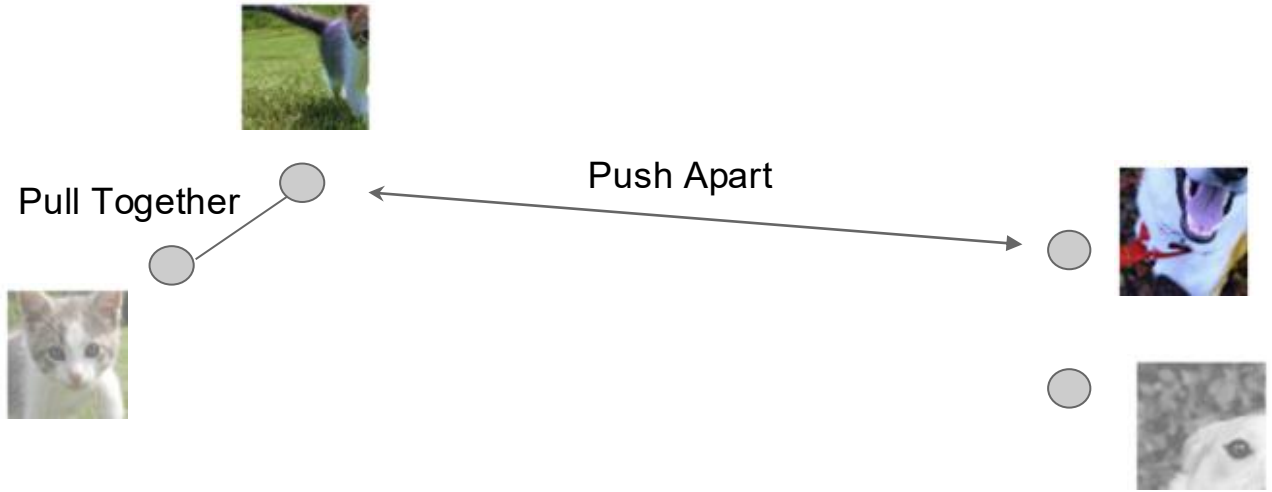
Recall this self-supervised objective from SimCLR



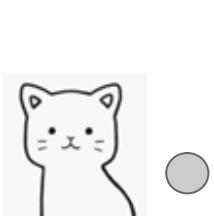
Use Self Supervised learning to learn good image features

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



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

“A cute fluffy cat”

“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”

How can we construct a joint visual-language embedding space?

Step #1: Collect a ton of data (~400M image-text pairs)

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

Step #1: Collect a ton of data (~400M image-text pairs)

> “To address this, we constructed a new dataset of 400 million (image, text) pairs collected from a variety of publicly available sources on the Internet. To attempt to cover as broad a set of visual concepts as possible, we ***search*** for (image, text) pairs as part of the construction process whose text includes one of a set of *500,000* ***queries*** We approximately class balance the results by including *up to 20,000 (image, text) pairs per query.*”

Quote from original CLIP paper – has since been replicated!

Xu et al, Demystifying CLIP Data, ICLR 2024

Step #2: Train a model

Step #2: Train a model

What loss function to use?

Step #2: Train a model

The authors describing prior work (e.g., image captioning):

> Both these approaches share a key similarity. They try to predict the *exact* words of the text accompanying each image. This is a difficult task due to the wide variety of descriptions, comments, and related text that co-occur with images. Recent work in contrastive representation learning for images has found that contrastive objectives can learn better representations than their equivalent predictive objective (Tian et al., 2019). Other work has found that although generative

Radford et al, Learning Transferable Visual Models from Natural Language Supervision

Step #2: Train a model

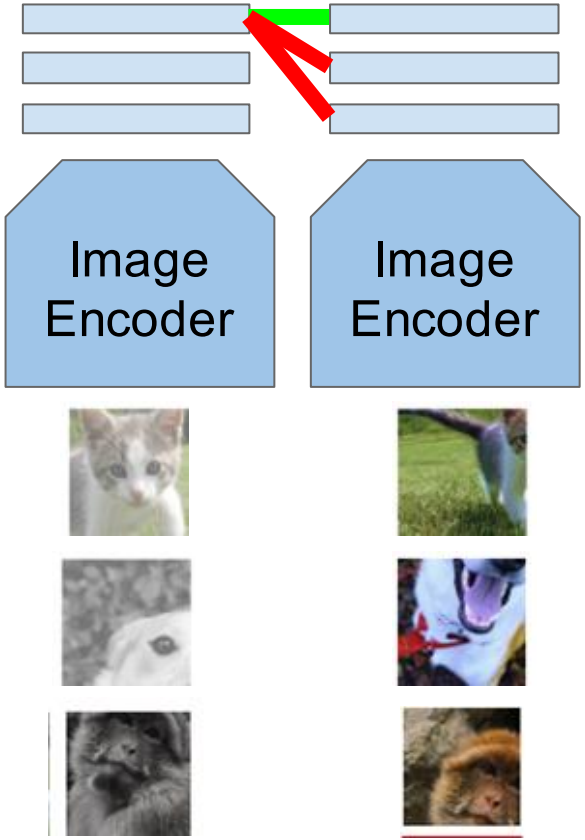
The authors describing prior work (e.g., image captioning):

> Both these approaches share a key similarity. They try to predict the *exact* words of the text accompanying each image. This is a difficult task due to the wide variety of descriptions, comments, and related text that co-occur with images. Recent work in contrastive representation learning for images has found that contrastive objectives can learn better representations than their equivalent predictive objective (Tian et al., 2019). Other work has found that although generative

Instead of predicting the EXACT words from an image, we only need to learn a model that matches the correct description to the image.

Radford et al, Learning Transferable Visual Models from Natural Language Supervision

SimClr



SimClr

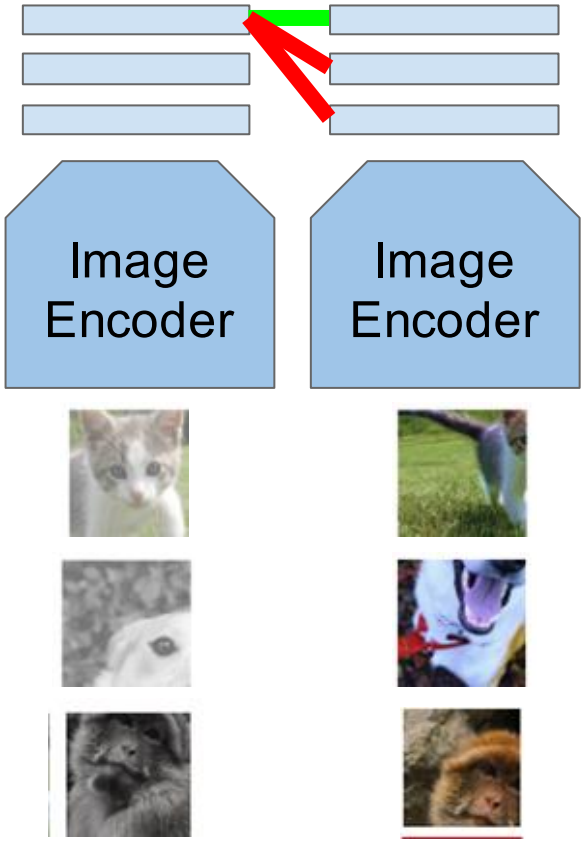


Image Encoder
Image Encoder

CLIP

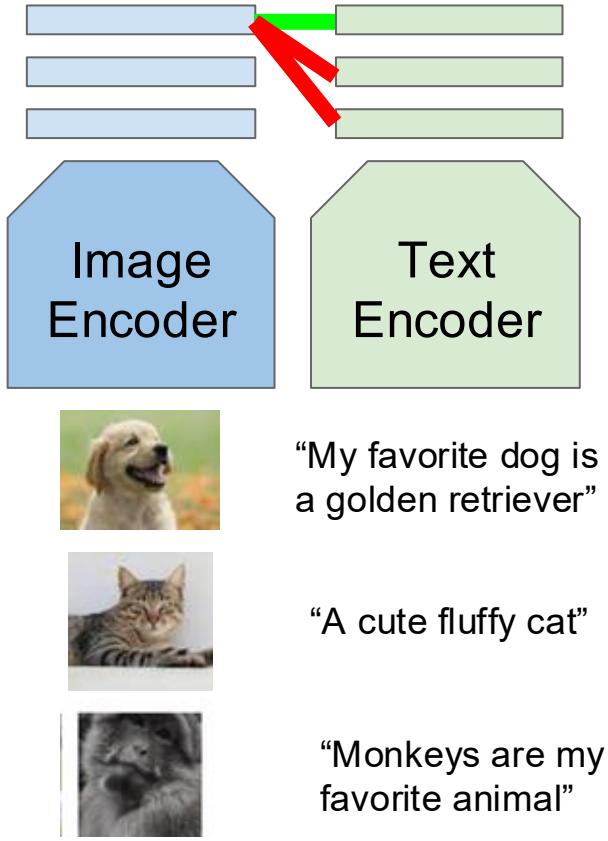
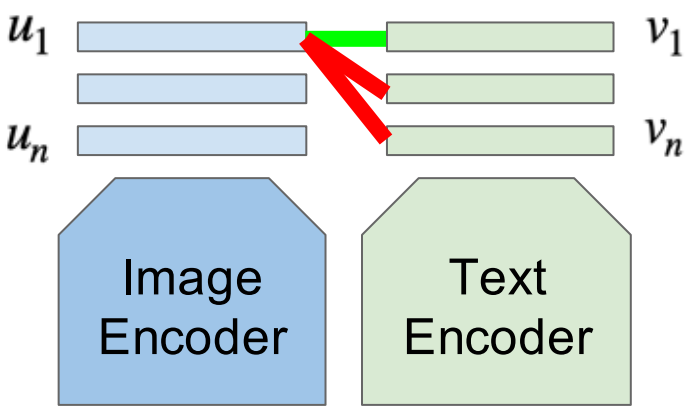


Image Encoder
Text Encoder
“My favorite dog is a golden retriever”
“A cute fluffy cat”
“Monkeys are my favorite animal”

CLIP is trained with the same contrastive objective (InfoNCE)

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



"My favorite dog is a golden retriever"

"A cute fluffy cat"

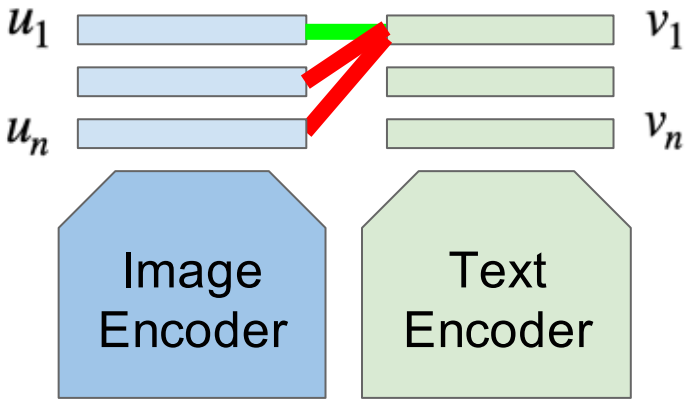
"Monkeys are my favorite animal"

***Some details not shown in slides (e.g. temperature parameter, vectors are L2 normalized)

CLIP Training Objective

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

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



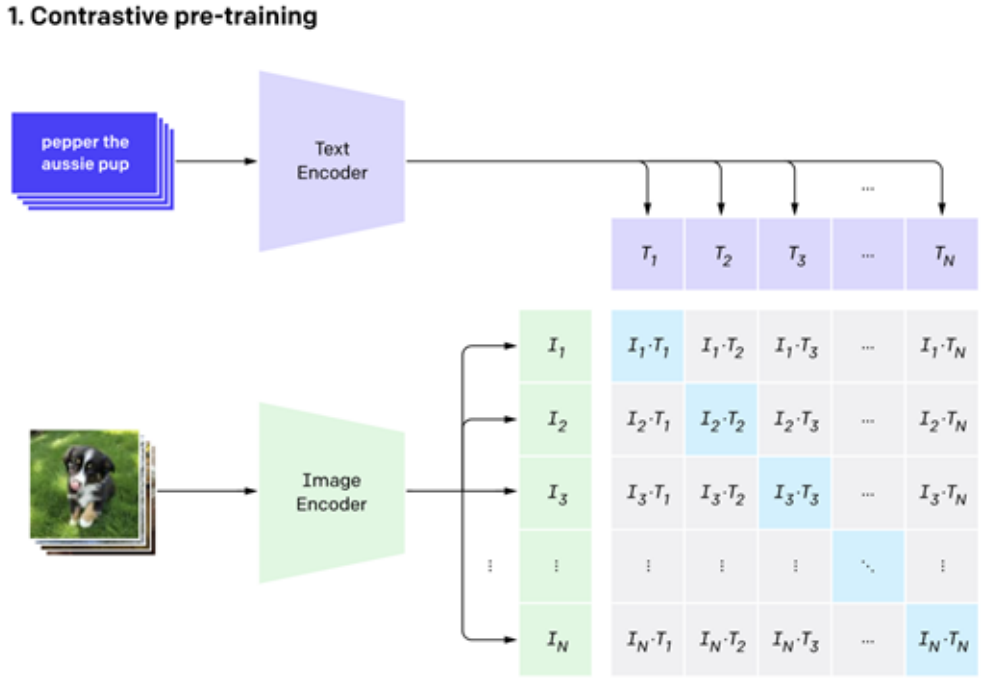
"My favorite dog is a golden retriever"

"A cute fluffy cat"

"Monkeys are my favorite animal"

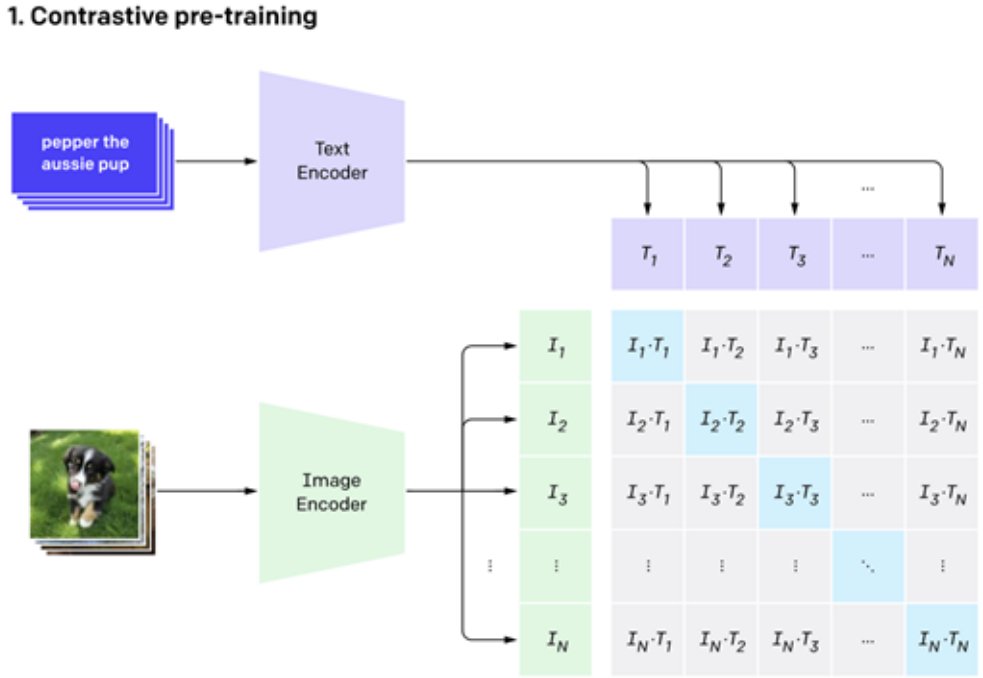
***Some details not shown in slides (e.g. temperature parameter, vectors are L2 normalized)

CLIP Training Objective



By minimizing this 2 way InfoNCE loss, we try to **maximize** the image-text **similarities along the diagonal**.

CLIP Training Objective

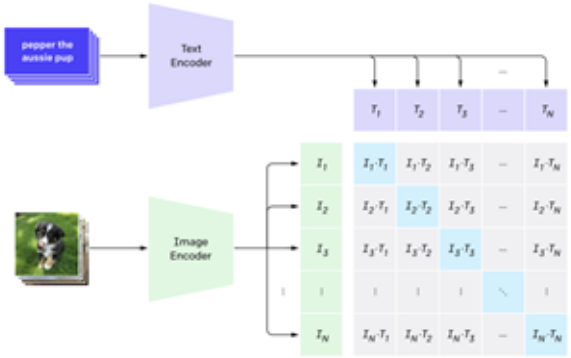


At the end of training, you have a model that can embed images and text

(and can give you a similarity score between an image and a text)

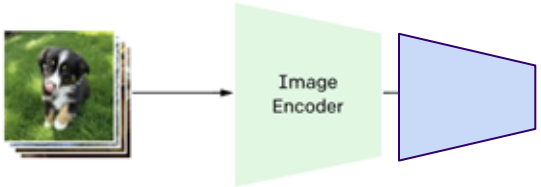
Using pre-trained models out of the box

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



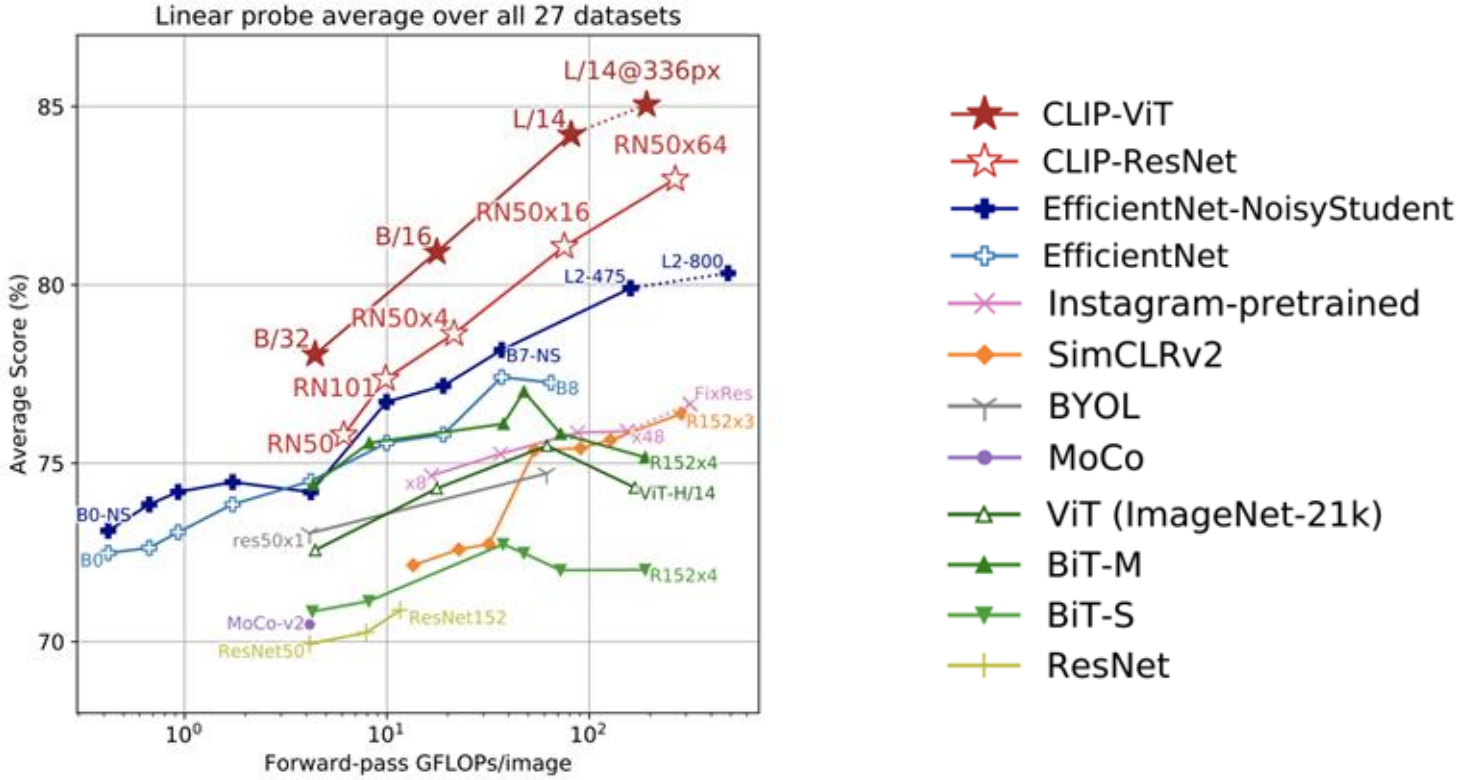
Pre-training tasks:
Contrastive Objective

Step 2: Transfer encoder to downstream tasks via linear classifiers



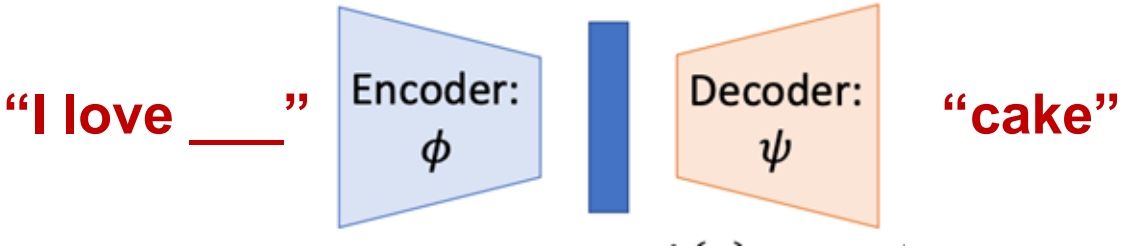
Downstream tasks:
Image classification, object detection, semantic segmentation

CLIP features w/ linear probe across multiple datasets

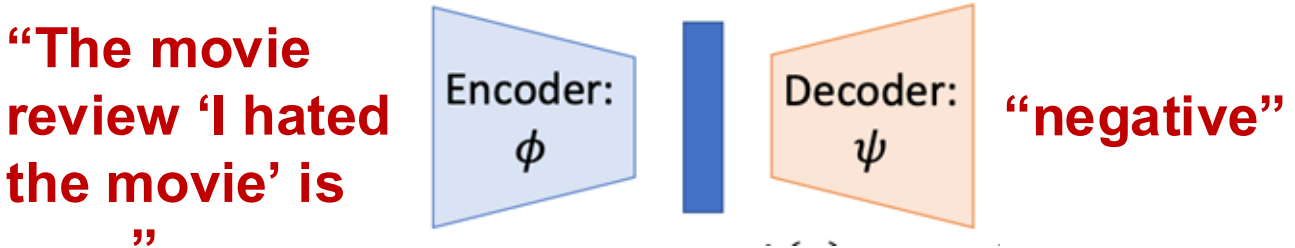


Big difference with language models: We can use LLMs zero-shot for new downstream tasks

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

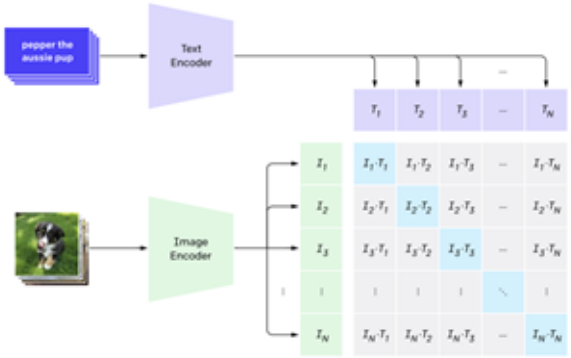


Step 2: Use the model out of the box in a creative way!



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

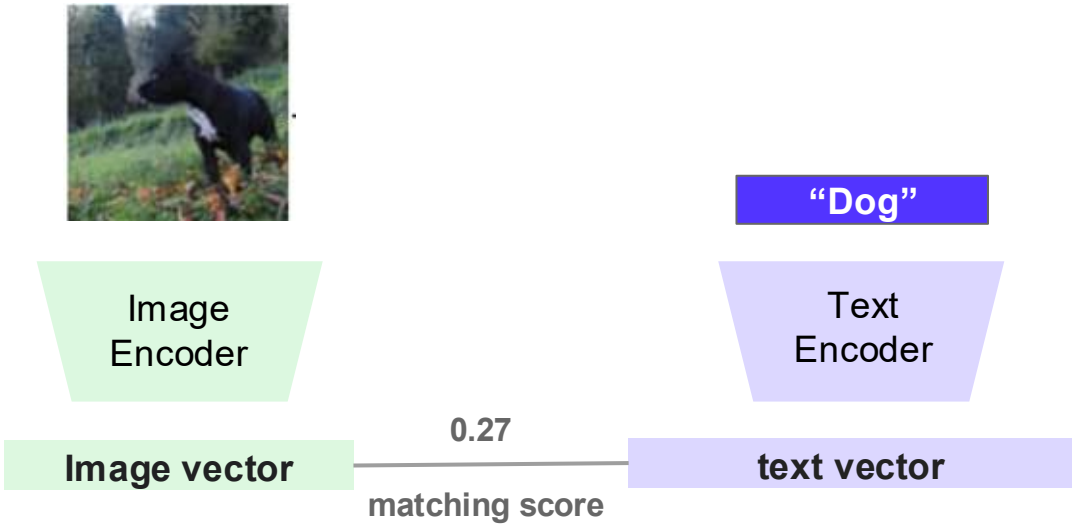


Pre-training tasks:
Contrastive Objective

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

Question: How to do 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

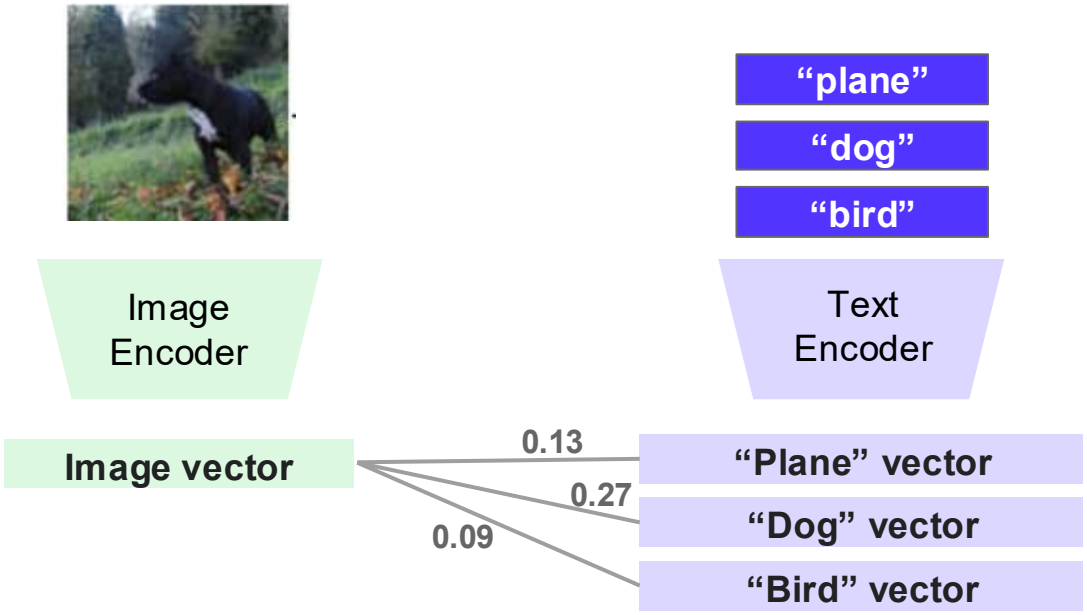
“Plane” vector

“Dog” vector

“Bird” vector

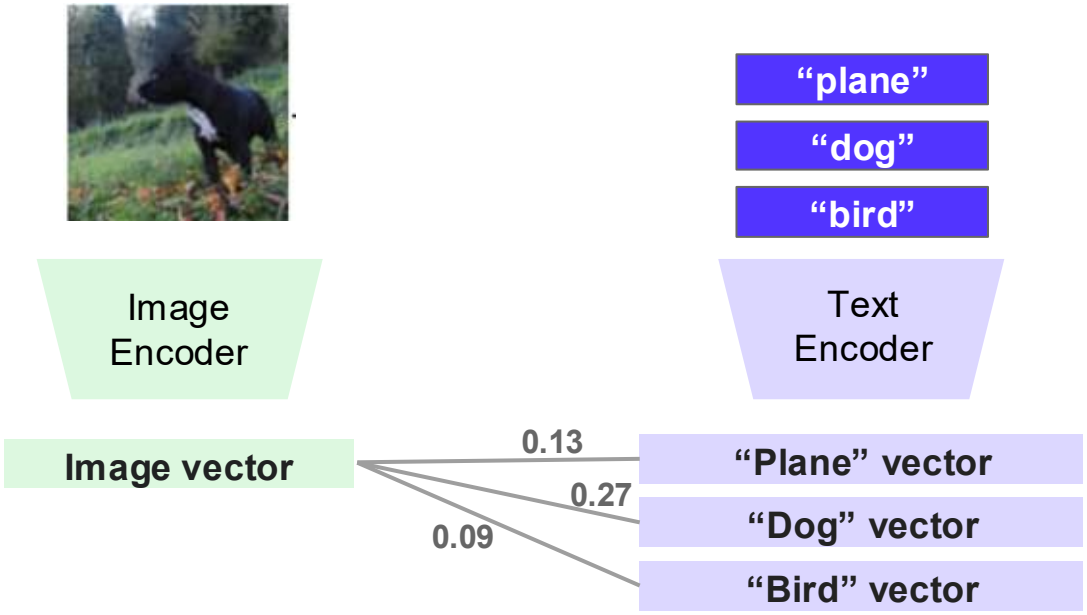
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Match a new image to the most similar vector



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

Similar to a 1-NN algorithm with the text 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]”

“A photo of a plane”

“A photo of a dog”

“A photo of a bird”

Image Encoder

Text Encoder

+1.3% on ImageNet

Image vector

0.13

0.27

0.09

“Plane” vector

“Dog” vector

“Bird” vector

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

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

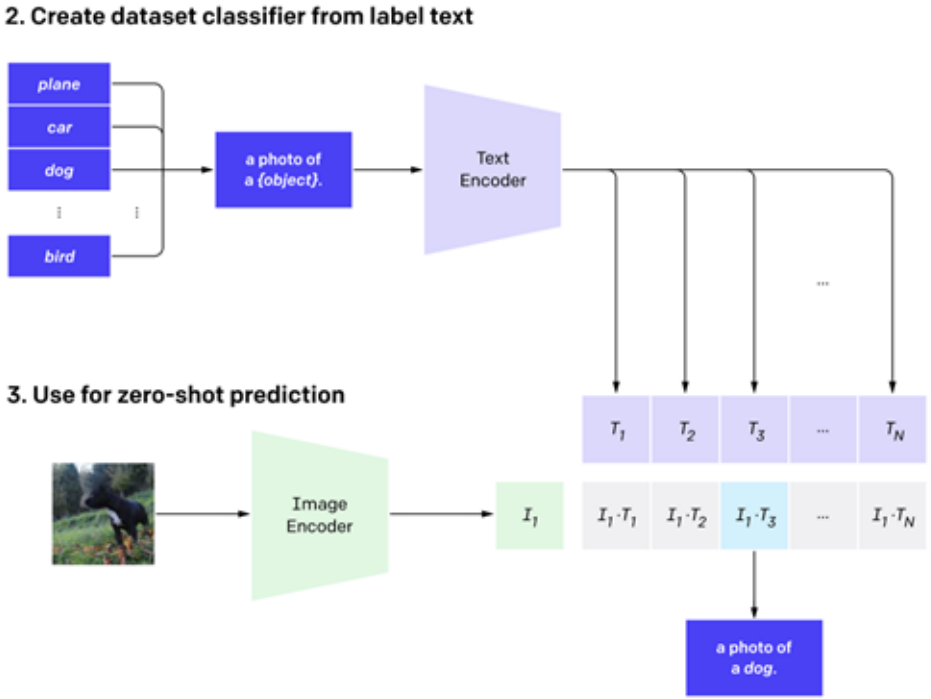
Mean “Plane” vector

Mean “Dog” vector

Mean “Bird” vector

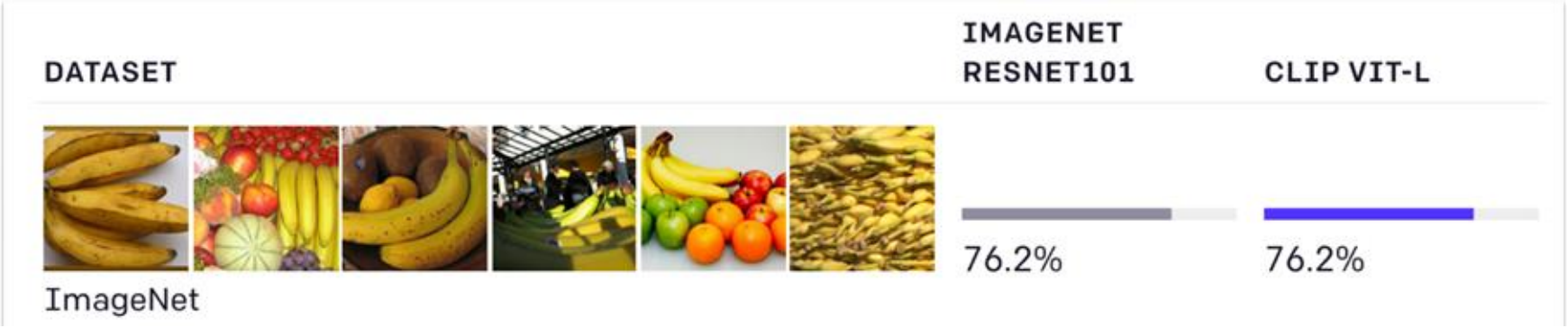
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

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"

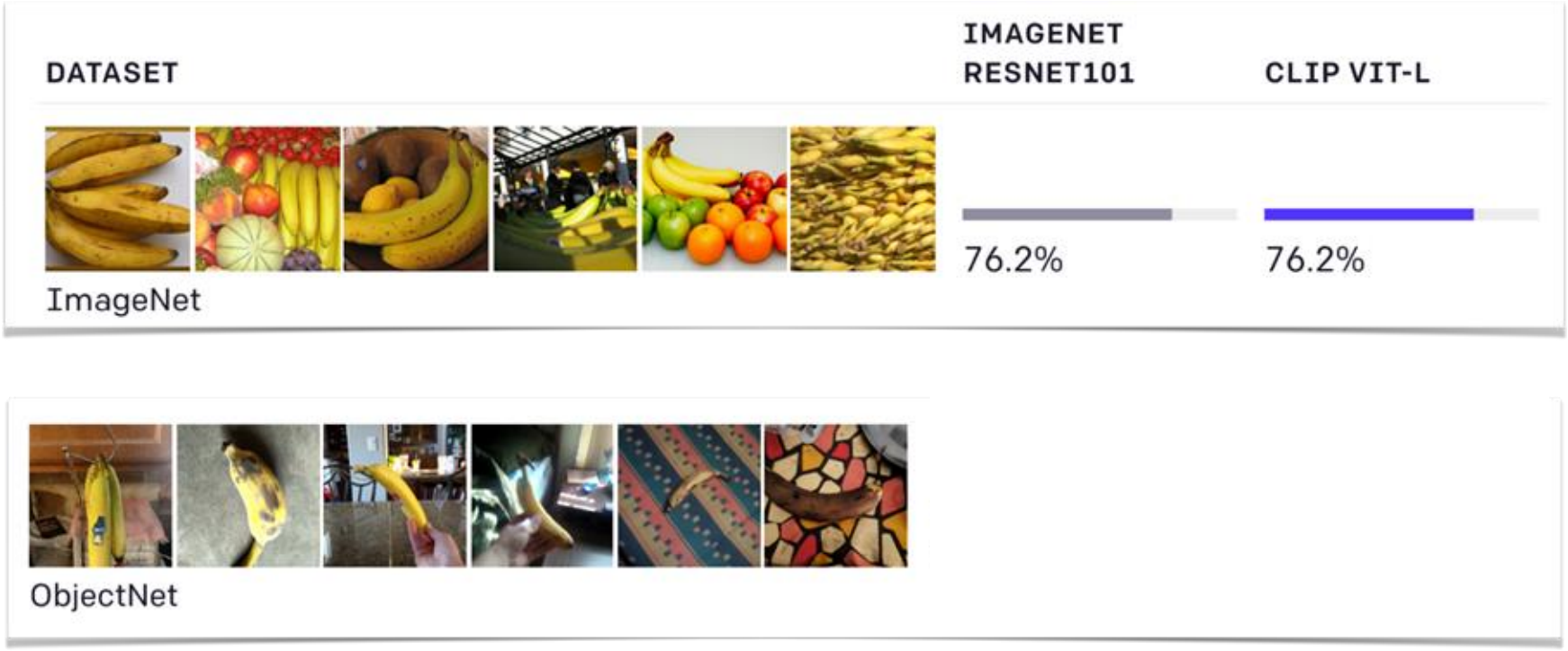
Exciting result after training on 400M image-text pairs



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

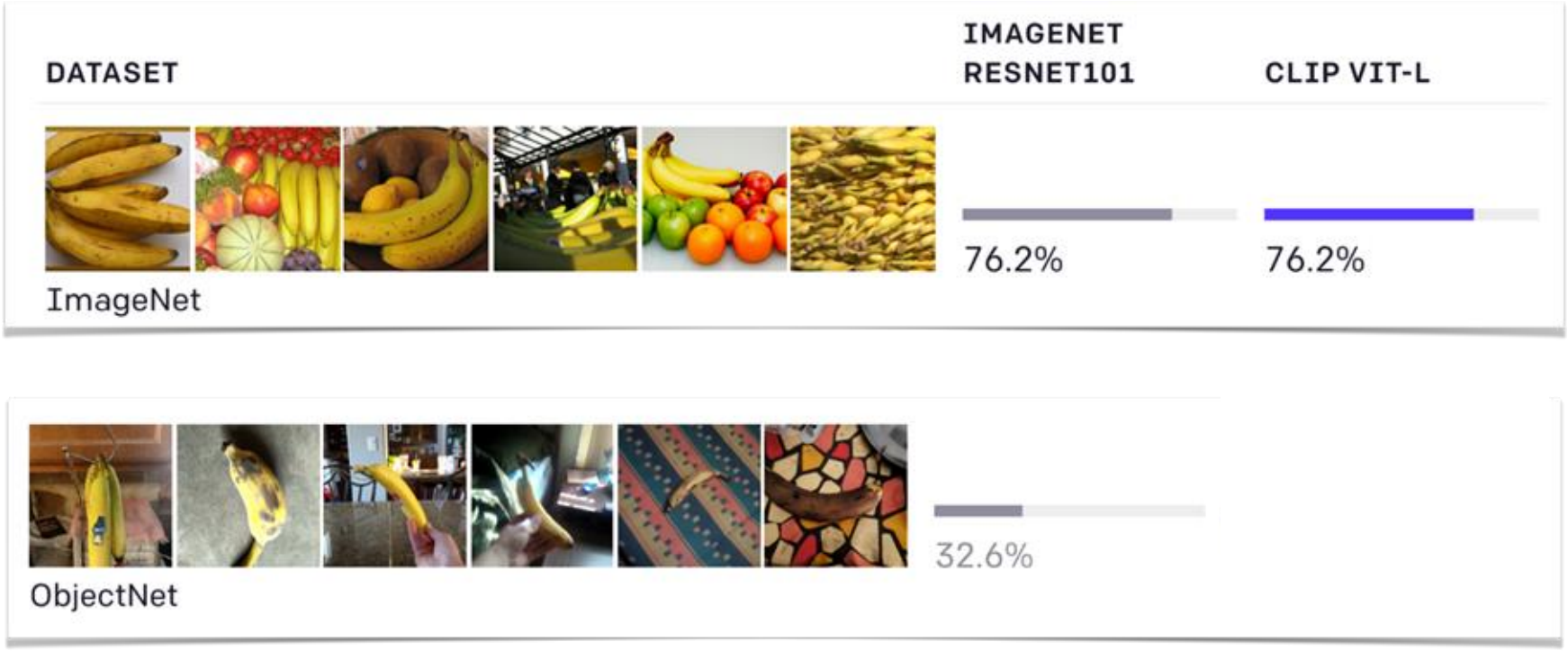
Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Here's where things get even more exciting



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

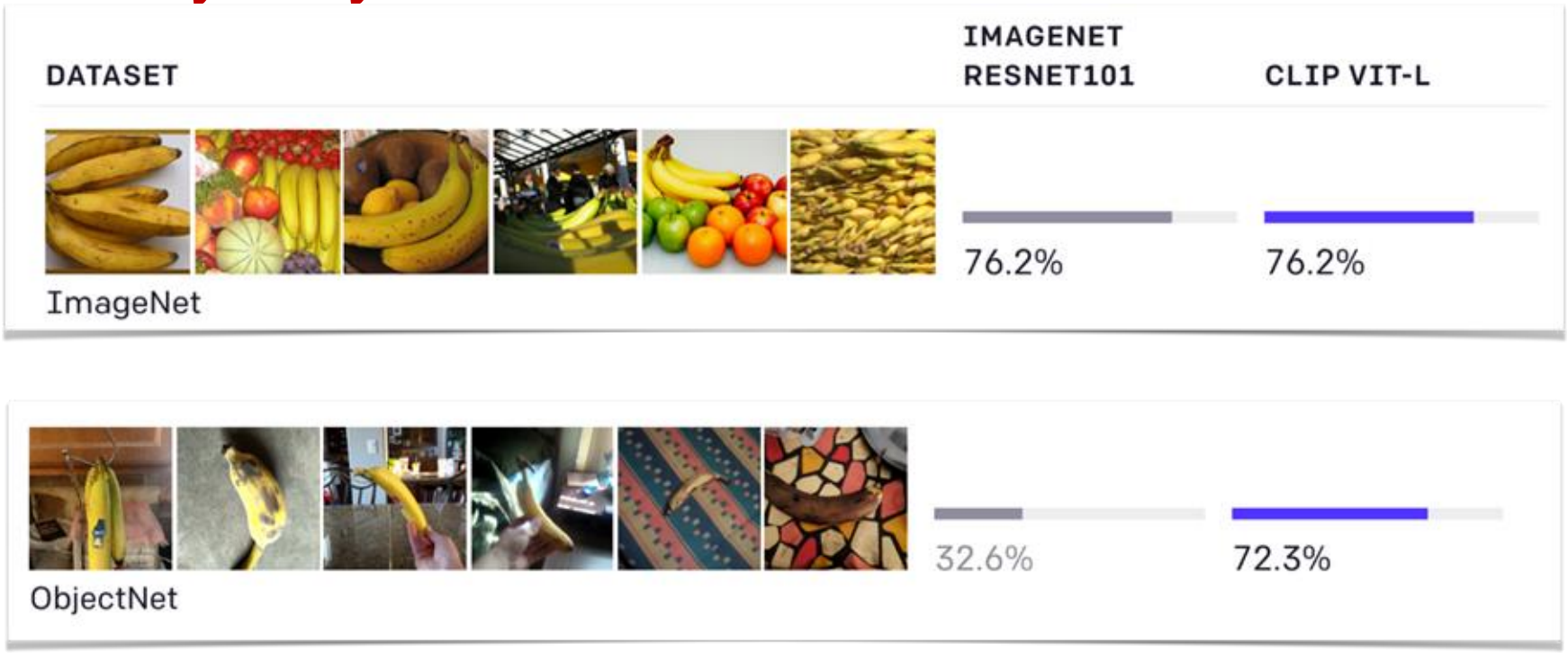
Training on ImageNet doesn't generalize to other datasets. ObjectNet contains the same categories but in weird viewpoints



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

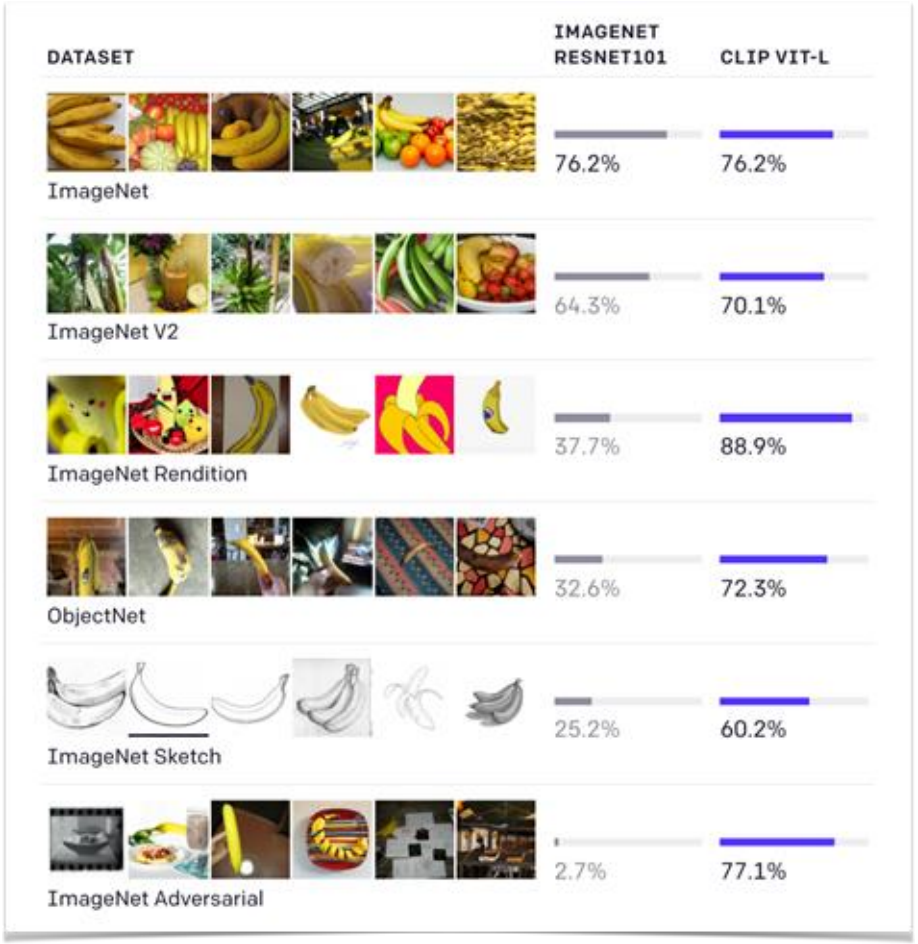
But CLIP zero-shot does so well!

Q. Why do you think that is?



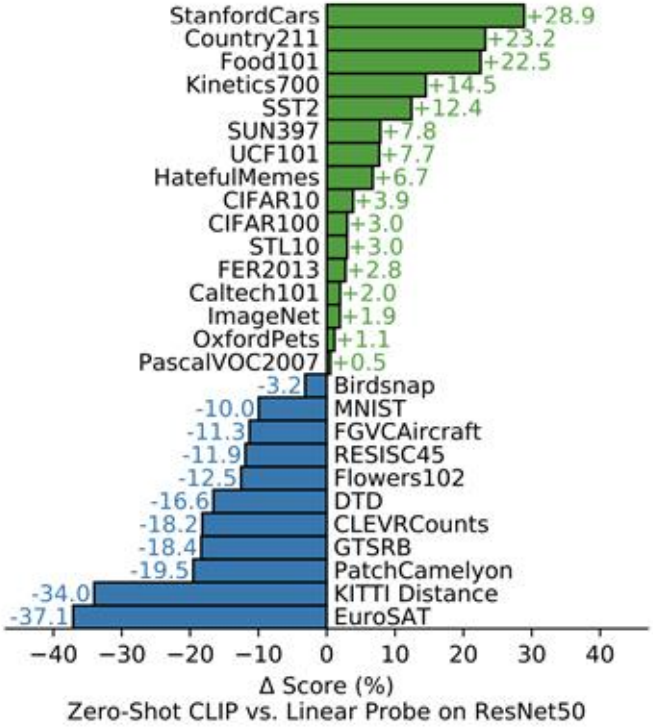
Radford et al "Learning Transferable Visual Models From Natural Language Supervision"

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

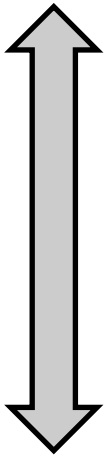


Radford et al “Learning Transferable Visual Models From Natural Language Supervision”

Difference in performance between linear probe vs zero-shot



Zero-shot is best



Linear classifier w/ CLIP features is best

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

Q: Why does CLIP perform so well?

How can no labels beat labels??

Q: Why does CLIP perform so well?

How can no labels beat labels??

Some possible answers:
1. “No labels” is a bit misleading
2. Massive pretraining scale
3. Possible test set leakage? Probably not: ~2% dataset leakage

Does CLIP’s Generalization Performance Mainly Stem from High Train-Test Similarity?
TLDR; Not really

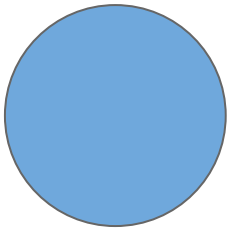
Q: Why does CLIP perform so well?

How can no labels beat labels??

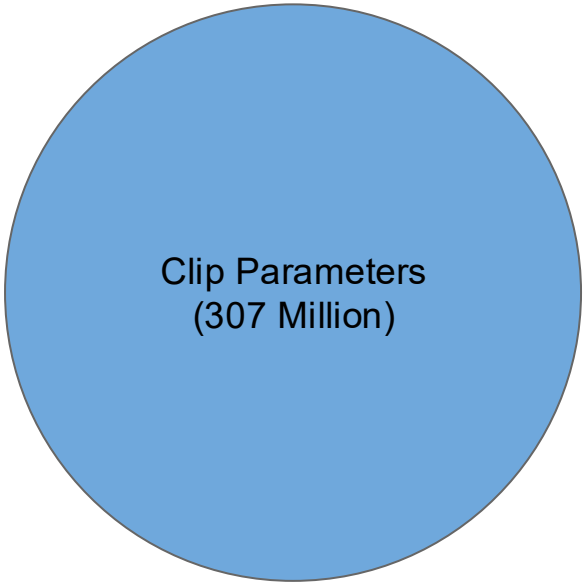
Some possible answers:
1. “No labels” is a bit misleading
2. Massive pretraining scale
3. Possible test set leakage? Probably not: ~2% dataset leakage

Does CLIP’s Generalization Performance Mainly Stem from High Train-Test Similarity?
TLDR; Not really

CLIP scaled up the model parameters with the transformer architecture

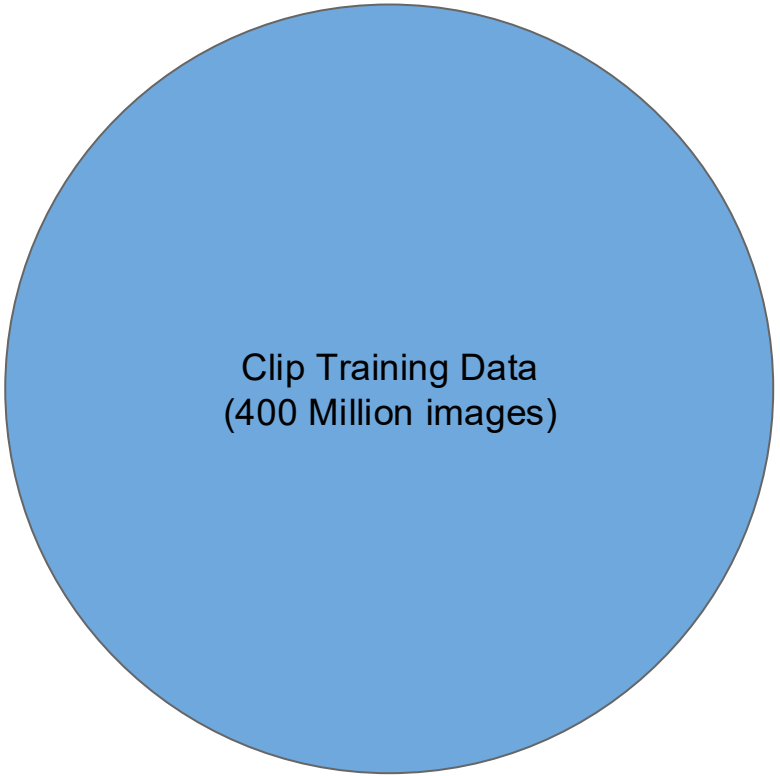


ImageNet ResNet Parameters
(44.5 Million)

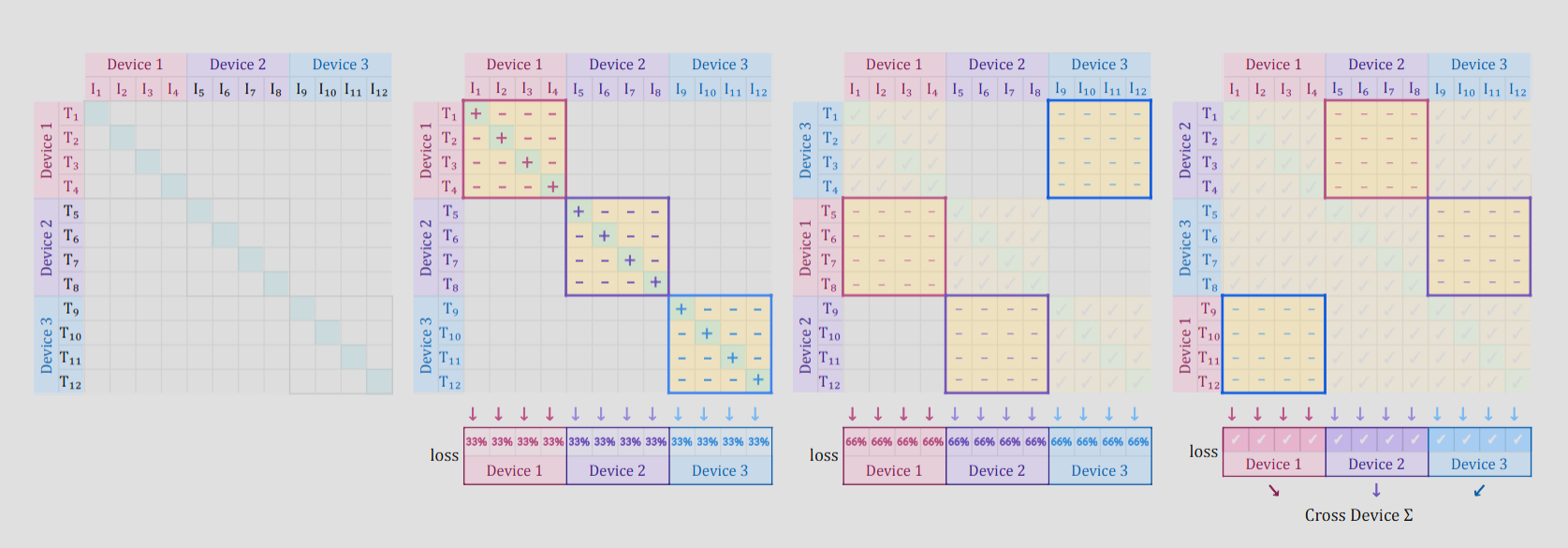


CLIP Scaled up the training data by scraping image-text pairs from the internet

ImageNet ResNet Training Data
(1.28 Million)

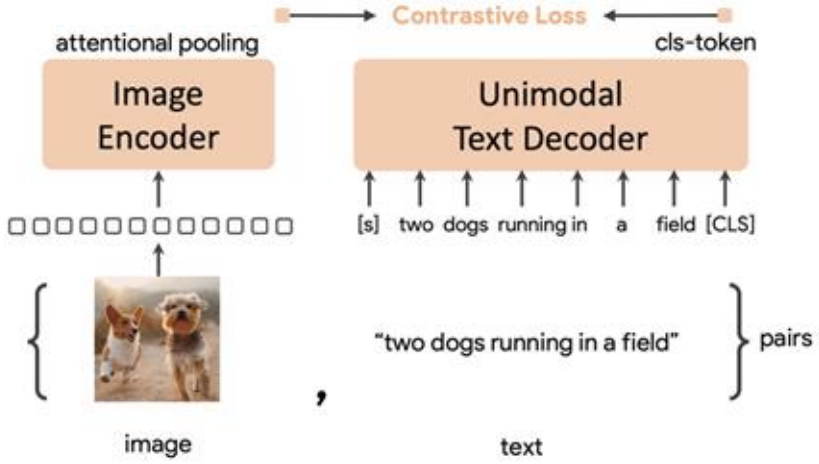


Common Variant: SigLIP uses Sigmoid Instead of Softmax



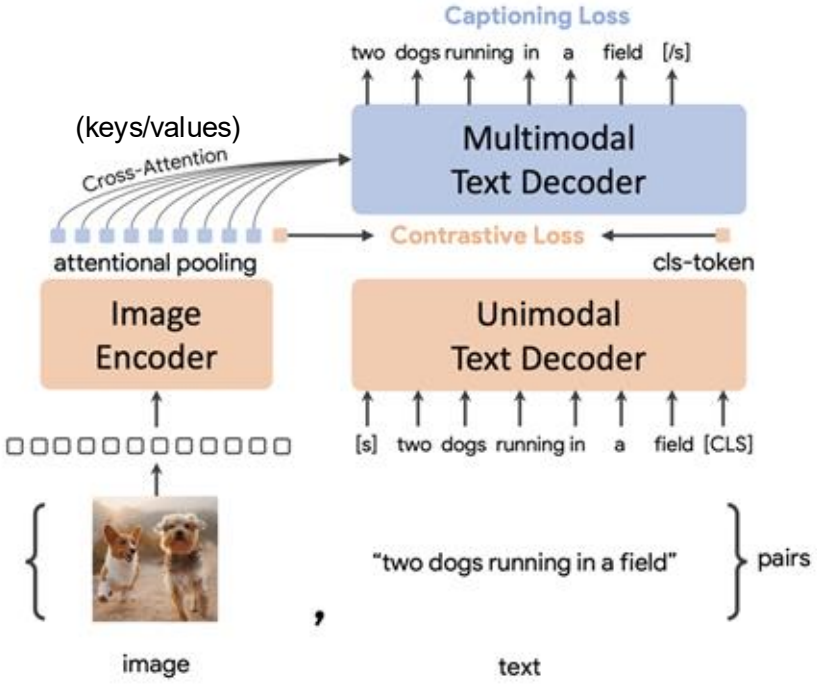
Very useful: does not require materializing the entire batch to compute loss per sample (can save memory)

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

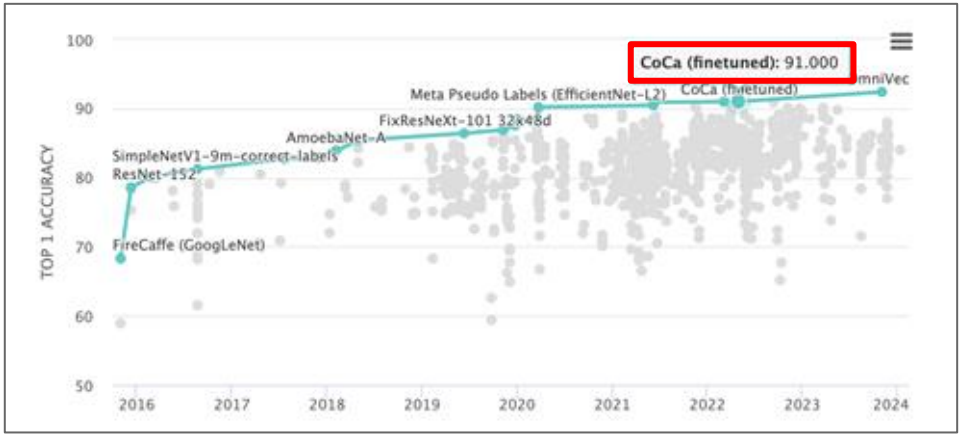
CoCa: Contrastive Captioners are Image-Text Foundation Models

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 dominate other models on ImageNet

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



Advantages of CLIP-style models

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

2. Open-vocabulary (zero-shot generalization)

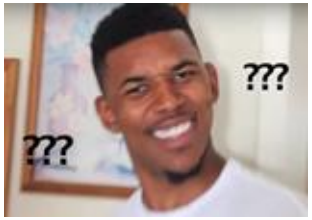
3. Can be chained with other models (CuPL)
   [we will discuss this later today]

April 2022, Tristan Thrush et al:

CLIP can’t distinguish between:

there is a mug in some grass

there is some grass in a mug



Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

Increasing batch size helps you understand fine-grained concepts

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

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

   Increasing batch size helps you understand fine-grained concepts

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

   Even in a batch of 32K, it's unlikely you see both "a mug in some grass" and "some grass in a mug"

Disadvantages of CLIP-style models

1. Rely too heavily on batch size to learn concepts

Winoground

there is a mug in some grass

there is some grass in a mug

"compositionality"

CREPE

Crepe on a skillet. 👁?

Boats on a skillet.

Crepe under a skillet.

Crepe on a dog.

ARO

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

Solution? Hard Negative Fine-Tuning

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
✗

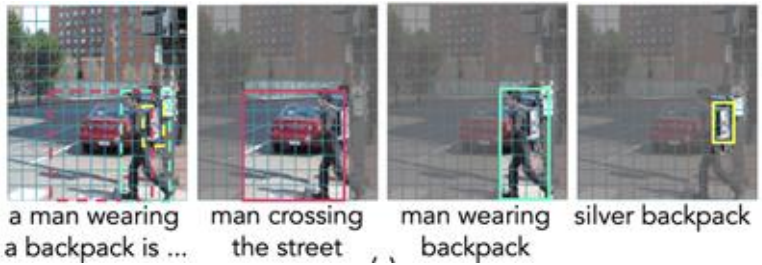
A brown dog and a black cat
✗

"hard positives"

Disadvantages of CLIP-style models

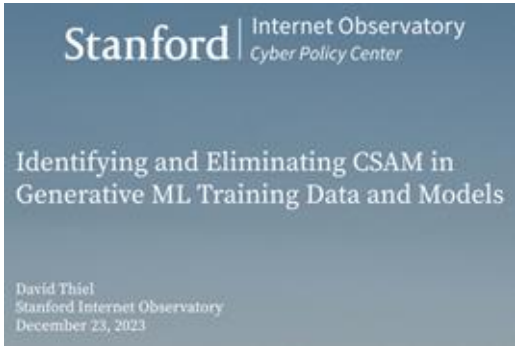
1. Rely too heavily on batch size to learn concepts
2. Image-level captions are insufficient supervision

Can 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 a single 5B dataset



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

Next, the Vision + Language Model Family

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

Next, the Vision + Language Model Family

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

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)

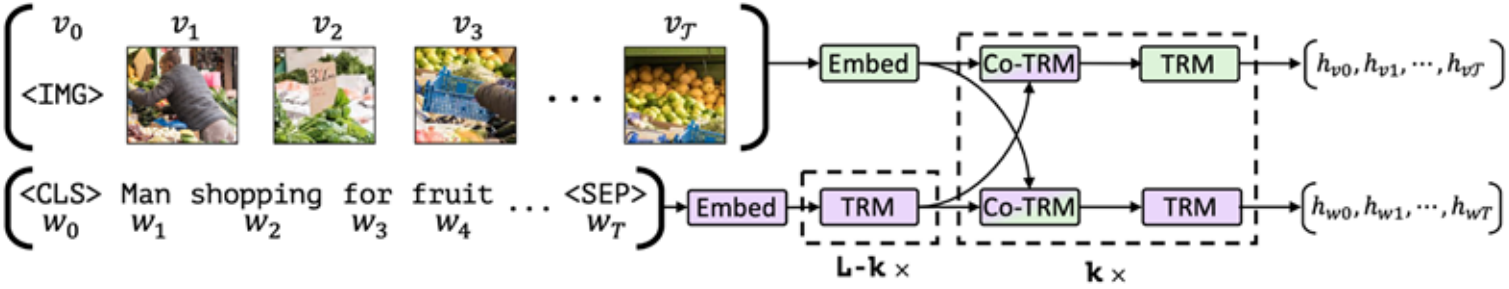
Can we build a model that can accept images and text as input, and then output text?

→ **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

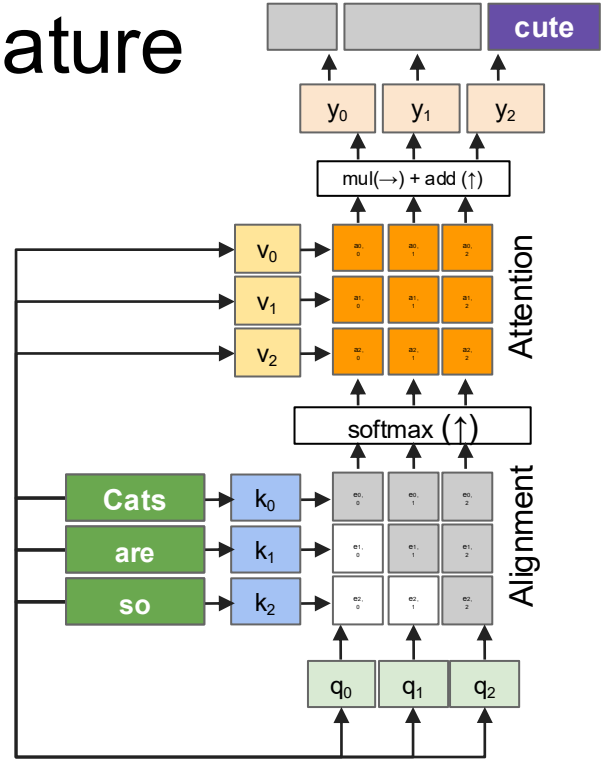
Vision-Language Models didn't start with LLaVA!

They go as far back as 2019 → ViLBERT

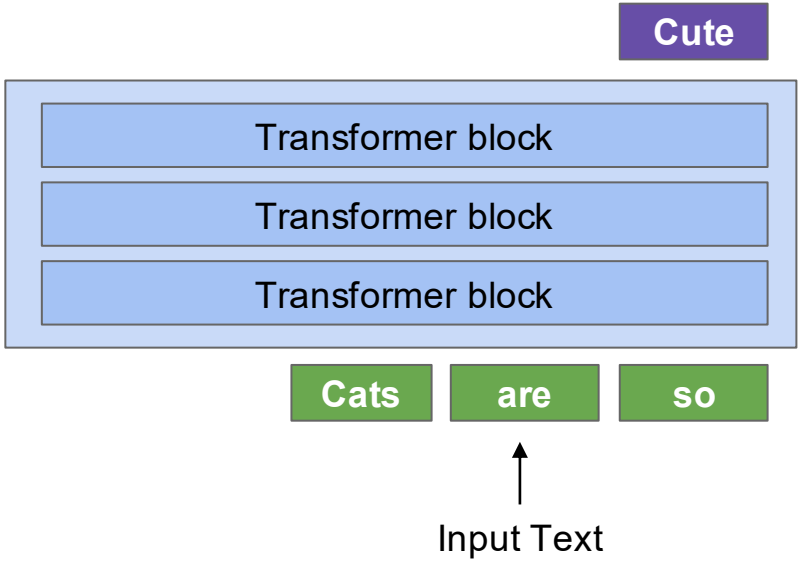
BUT, they had to finetune for each task separately,
with non-trivial task-specific methods (e.g., Mask-RCNN
bounding box re-ranking for RefCOCO)

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

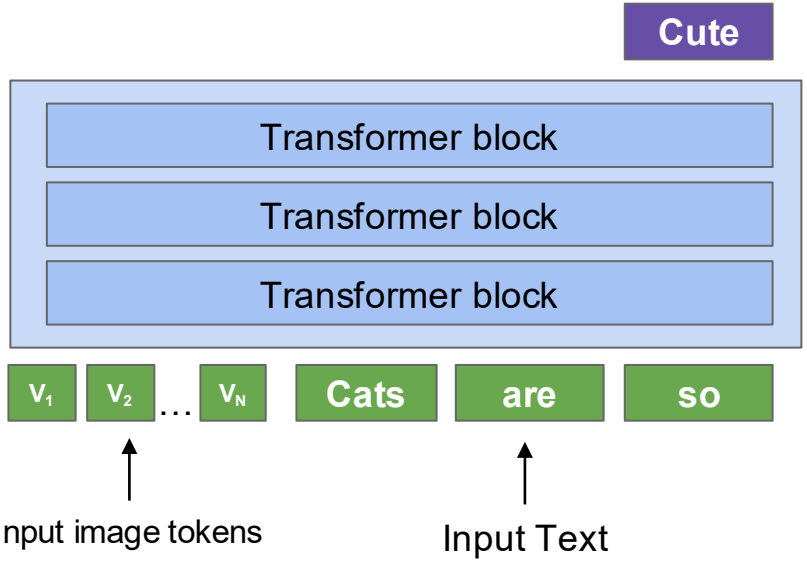
LLaVA uses the autoregressive nature of LLMs



Recall how transformers decode language

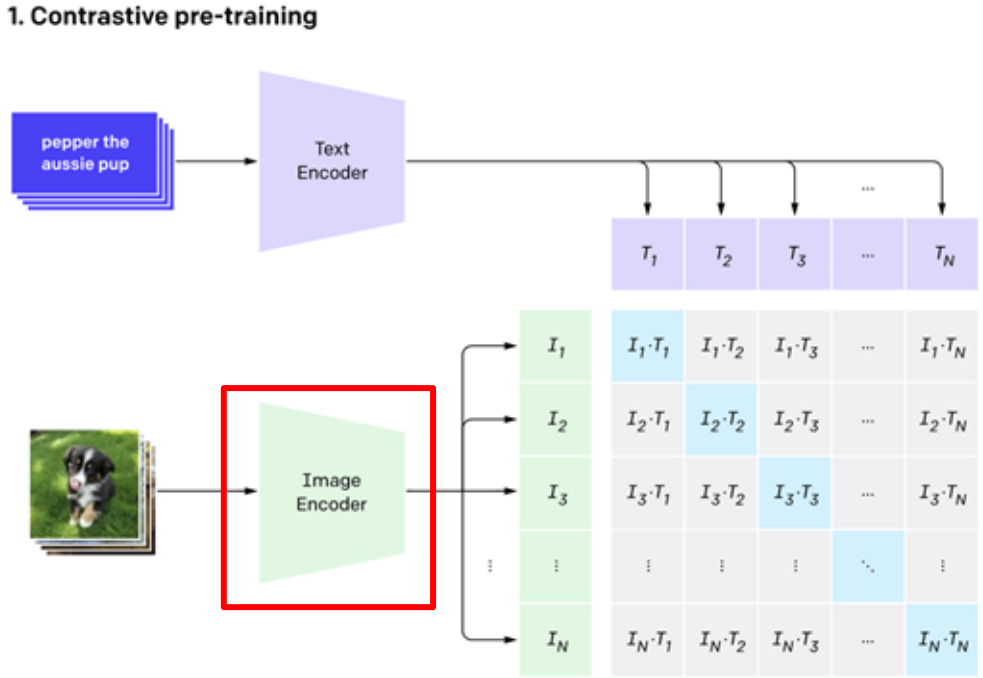


Key idea behind LLaVA – add visual information to the LLM



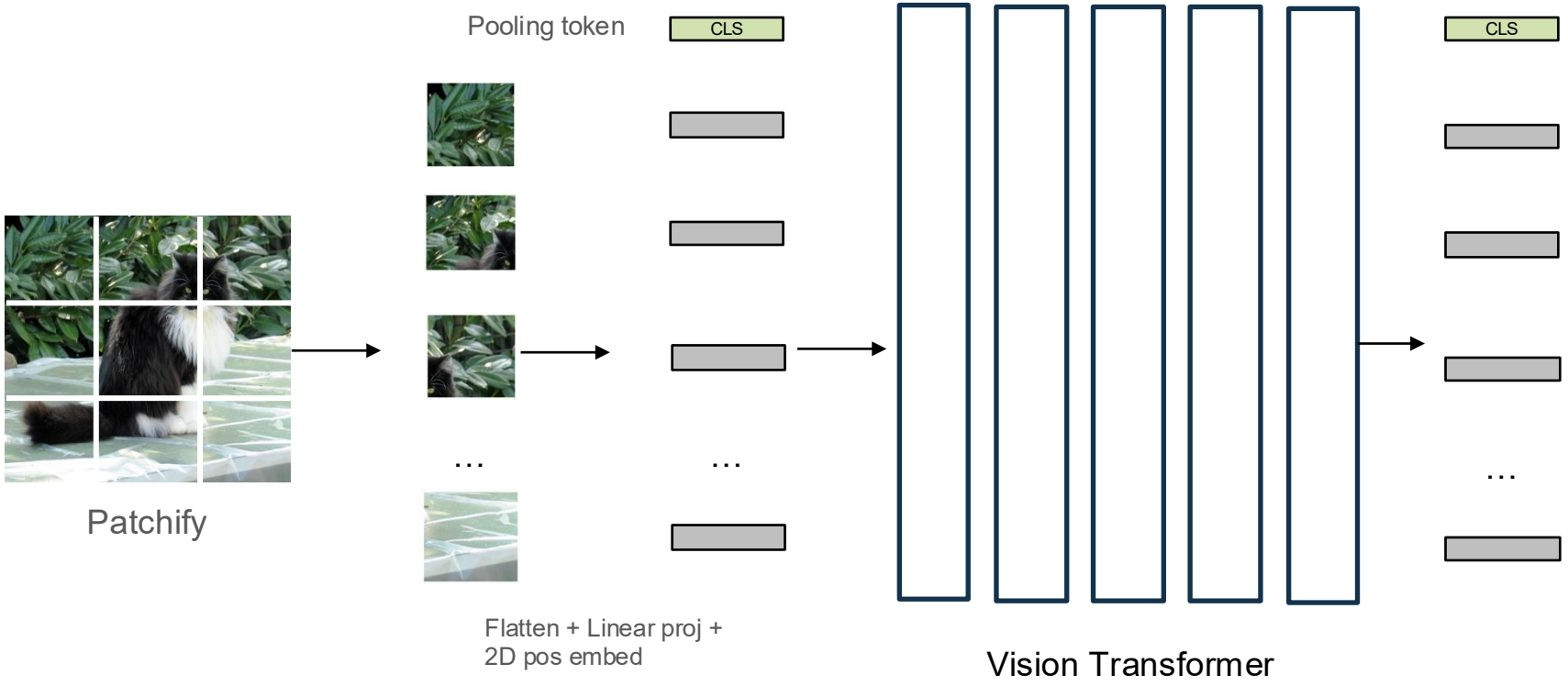
Q: Which image tokens work best here?

The CLIP encoder is a good option!



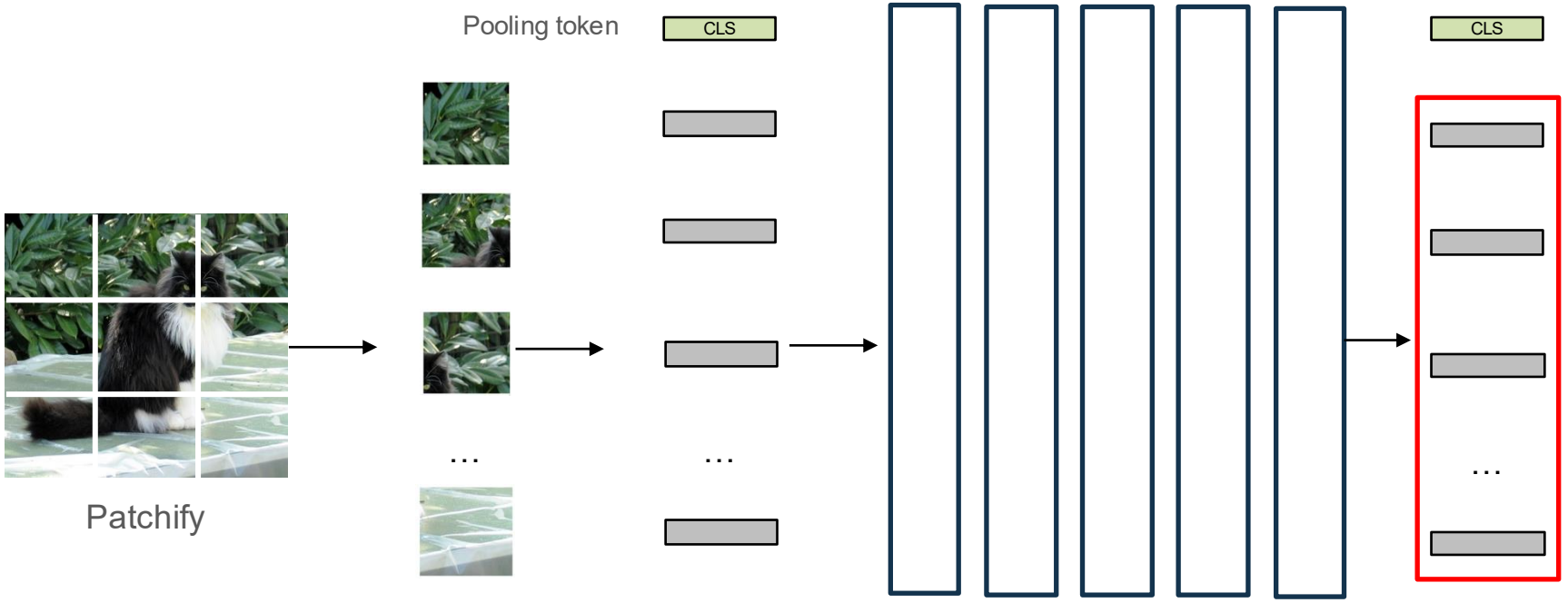
What features should we use from CLIP?

Extract Image Features from CLS token for contrastive learning

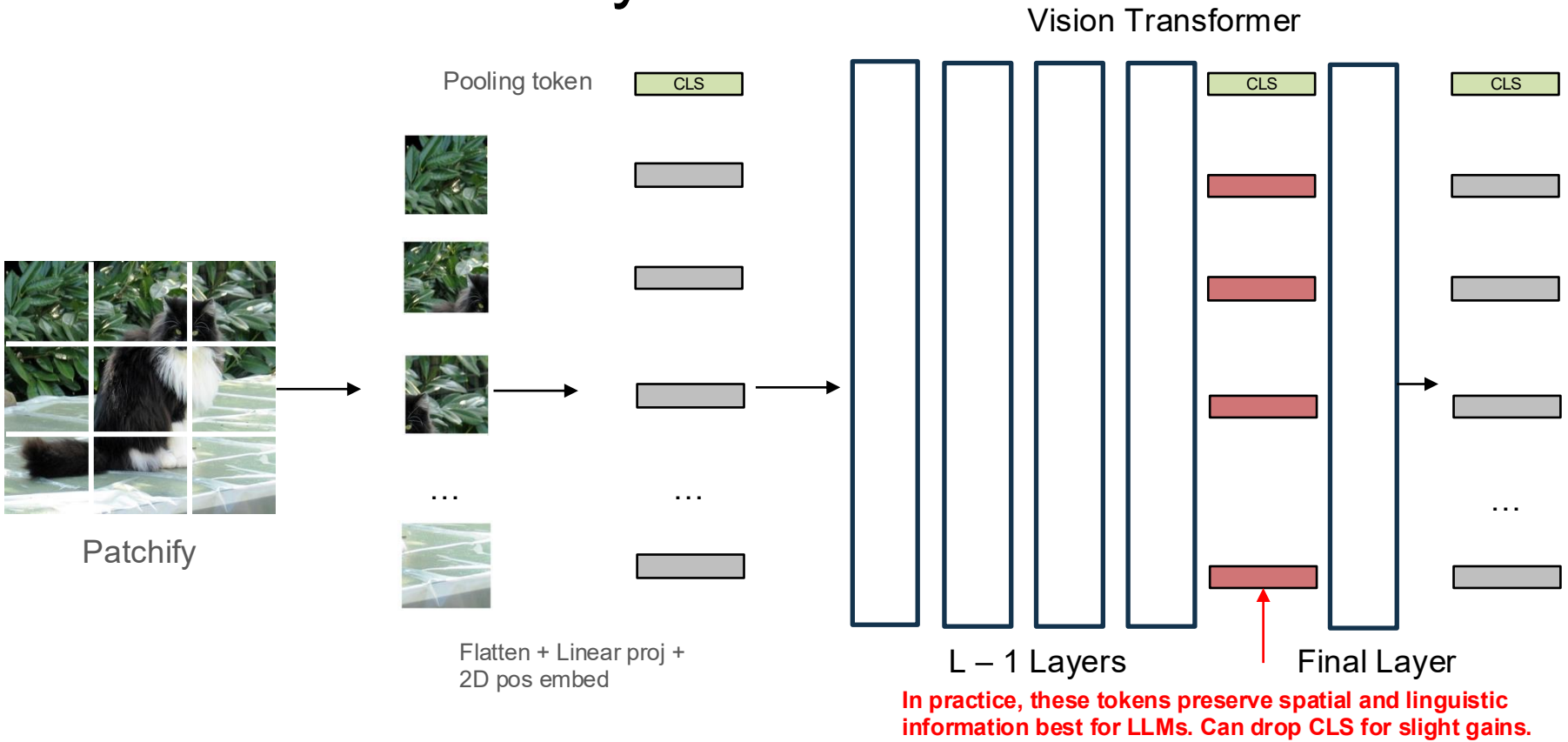


[Image source]

What features should we use from CLIP?

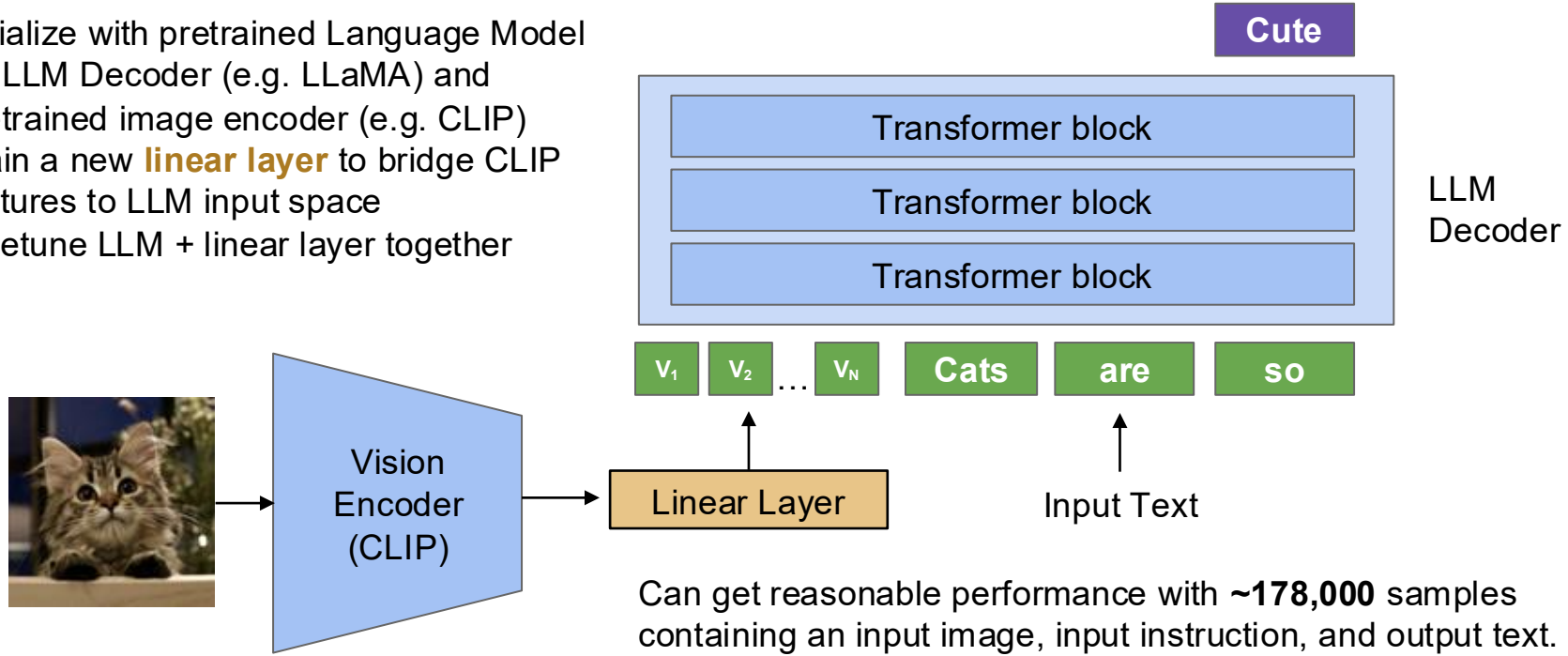


Use Penultimate Layer!



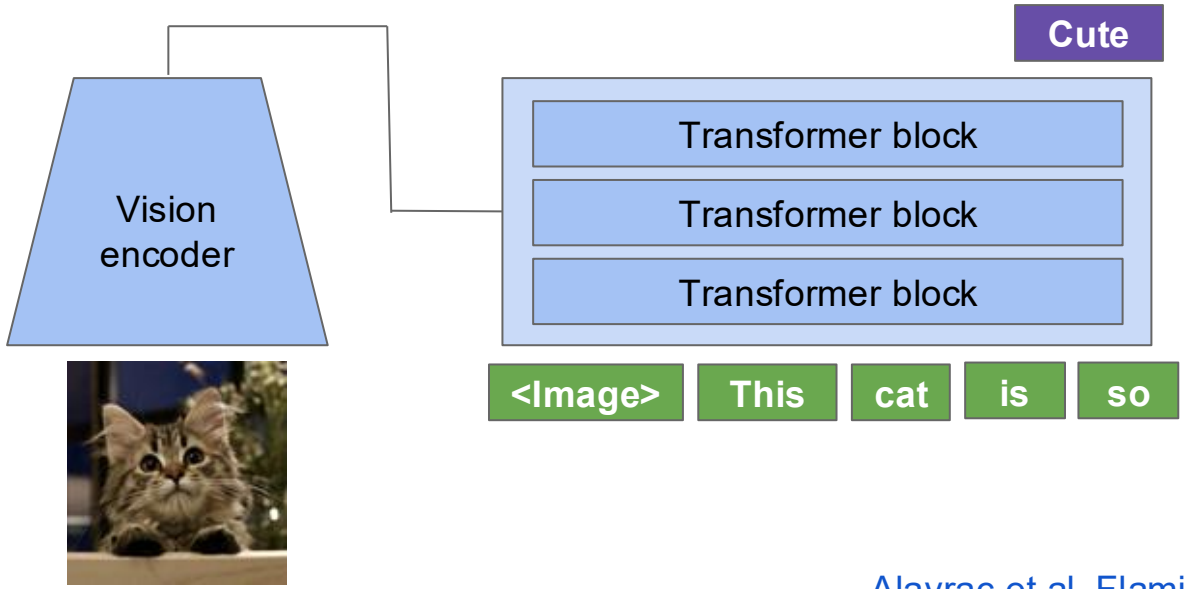
LLaVA – Overall Architecture + Training Recipe

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



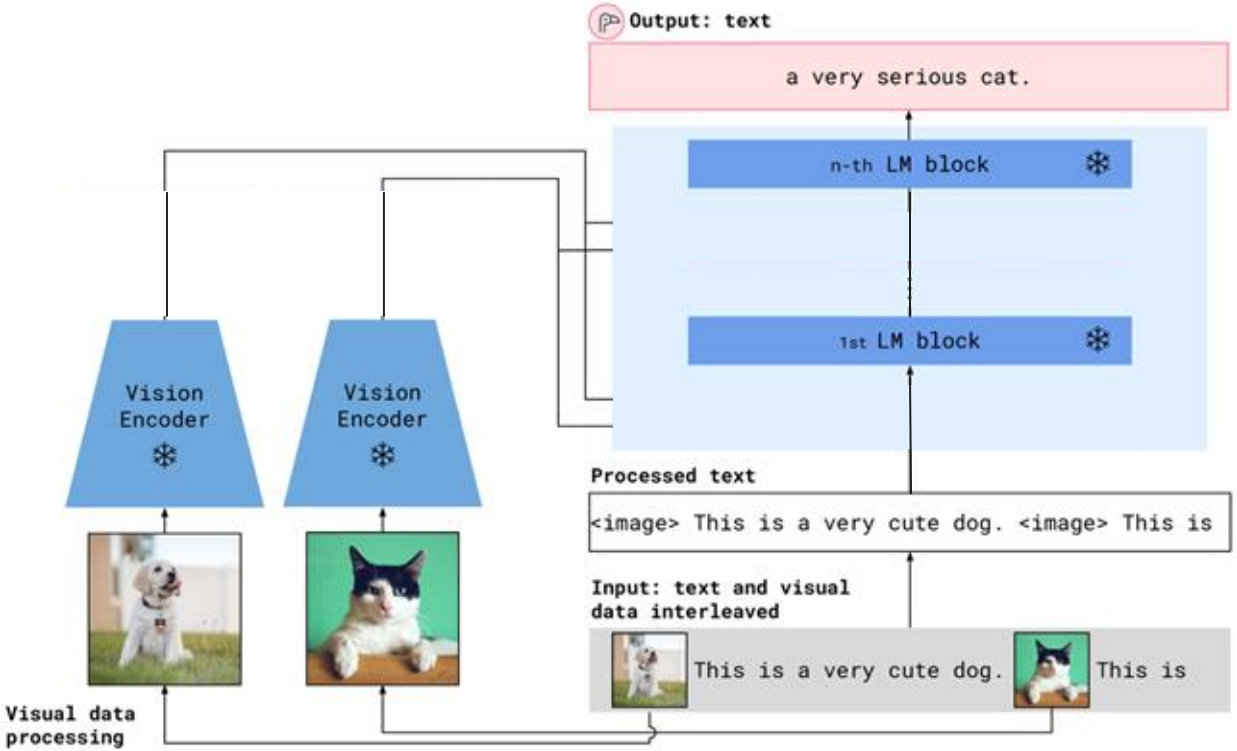
Can get reasonable performance with **~178,000** samples containing an input image, input instruction, and output text.

However, there are other ways to fuse information here. Flamingo proposed an alternative way



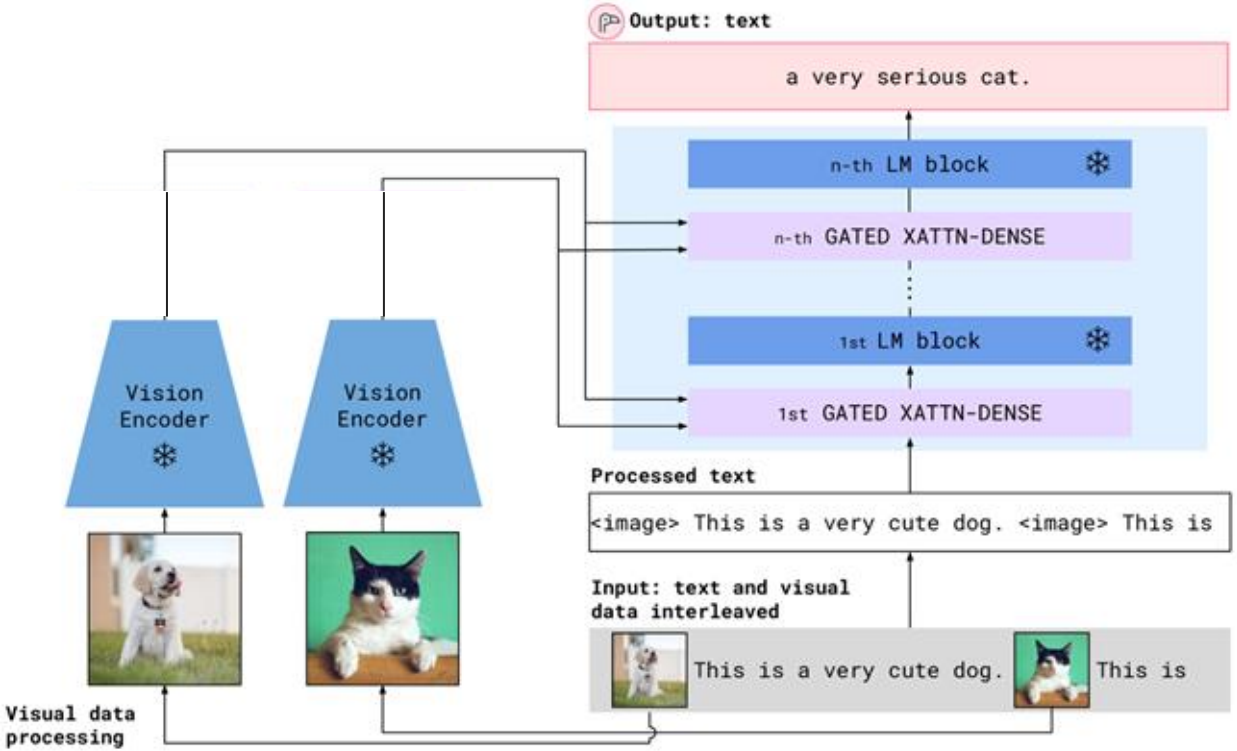
Alayrac et al. Flamingo, NeurIPS 2022

Pre-trained parts of Flamingo



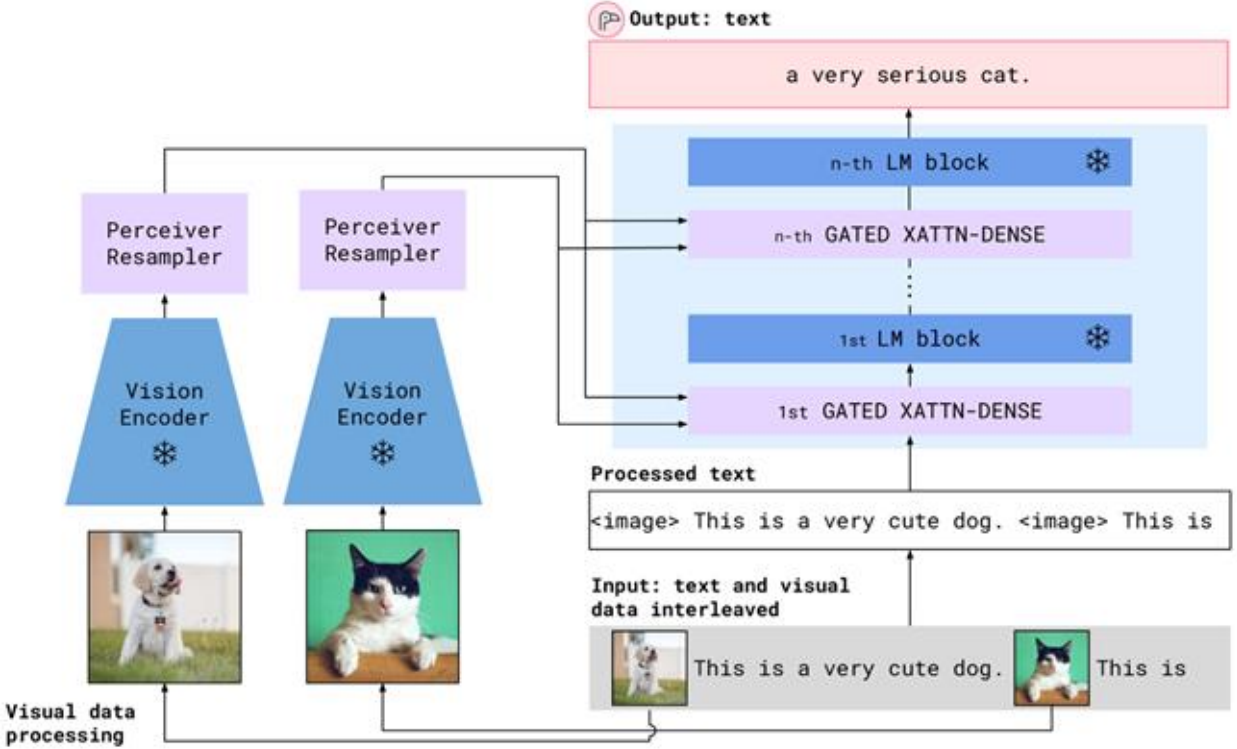
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

There are 2 learned parts in Flamingo



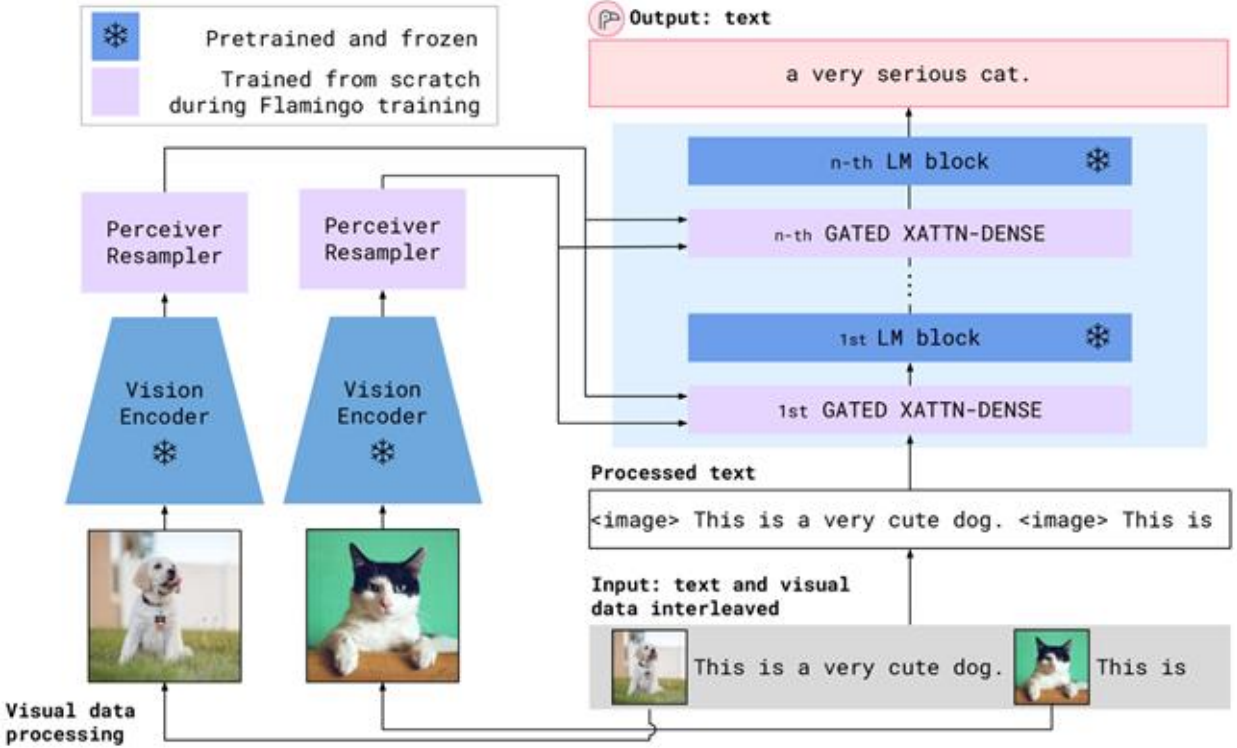
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

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



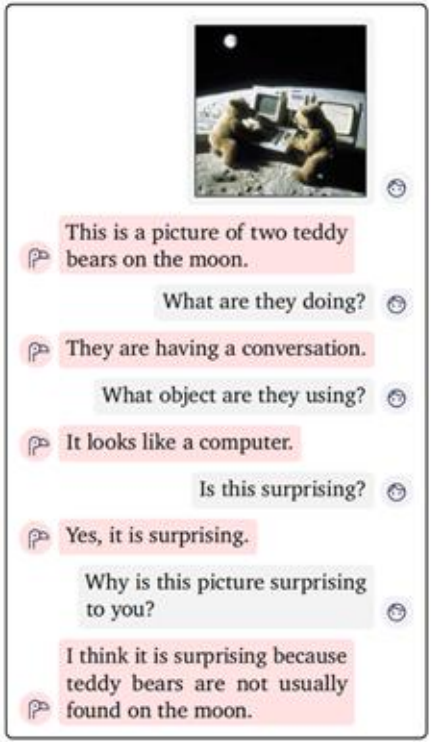
Alayrac et al “Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

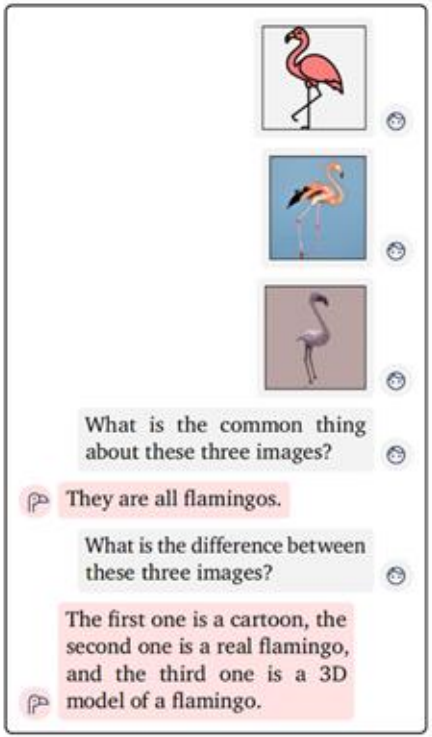
Flamingo arranges its training data similar to language modeling,

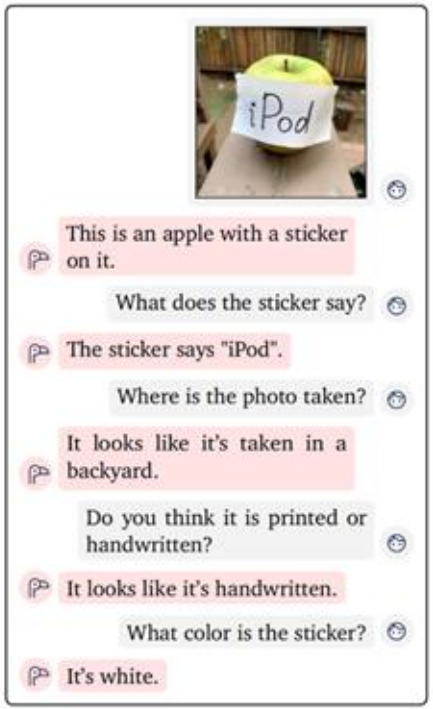
with special tags <image>, <eos> to indicate when a new image shows up or the text ends.

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

Flamingo results

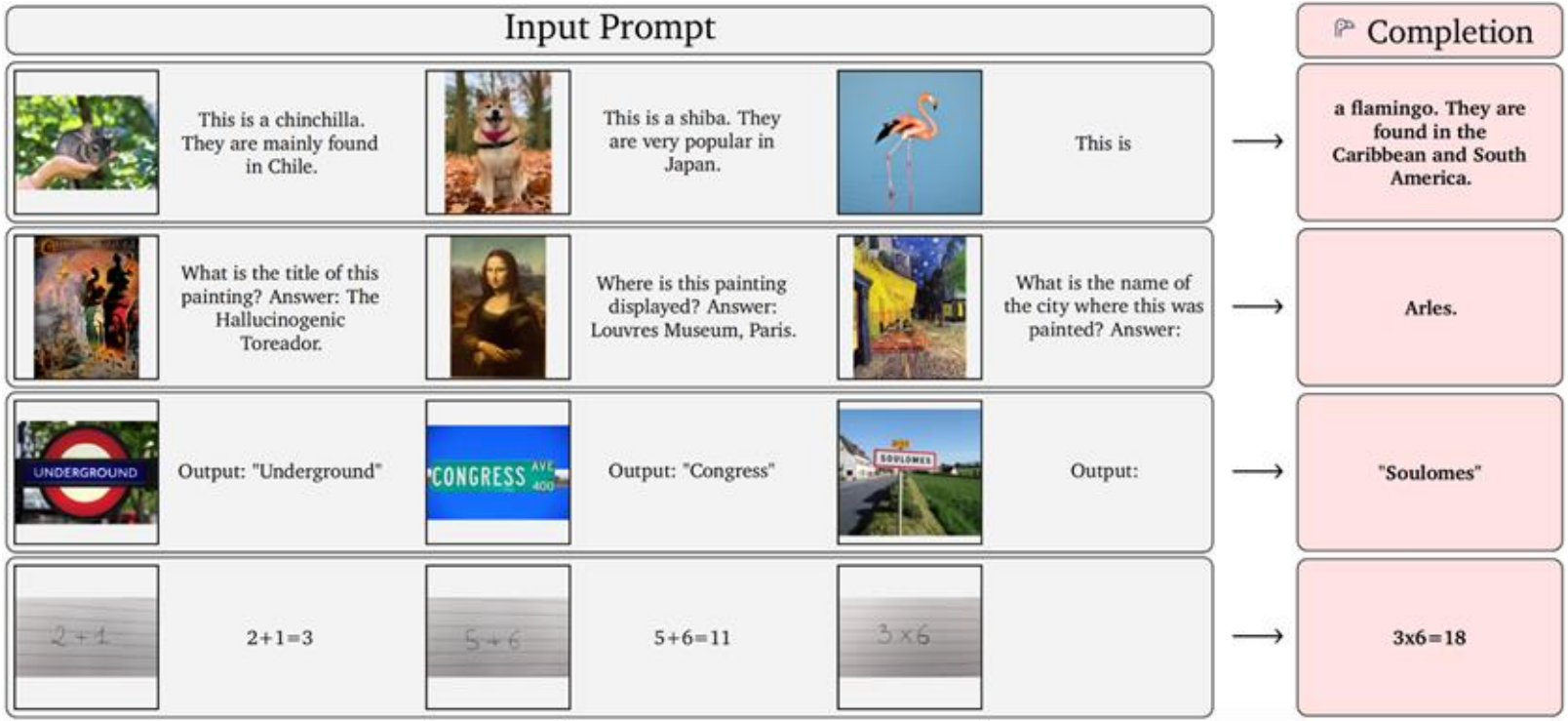






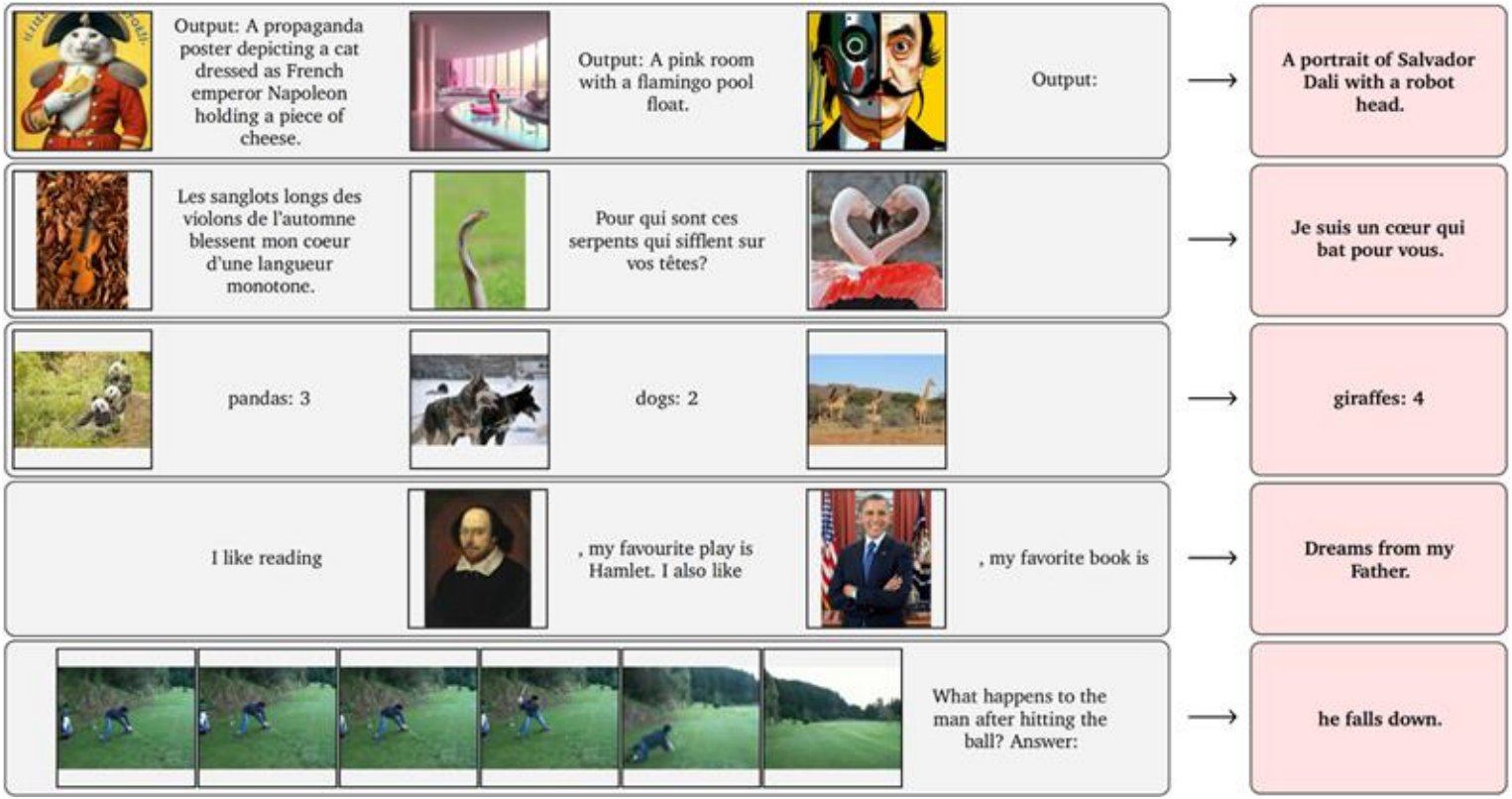
Alayrac et al "Flamingo: a Visual Language Model for Few-Shot Learning. 2022.

Flaming 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.

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)	-

What is SOTA in 2026 (so far)?

Gemini widely considered the best proprietary Vision-Language model

What about open source models?

Open Weight (can download + run locally)

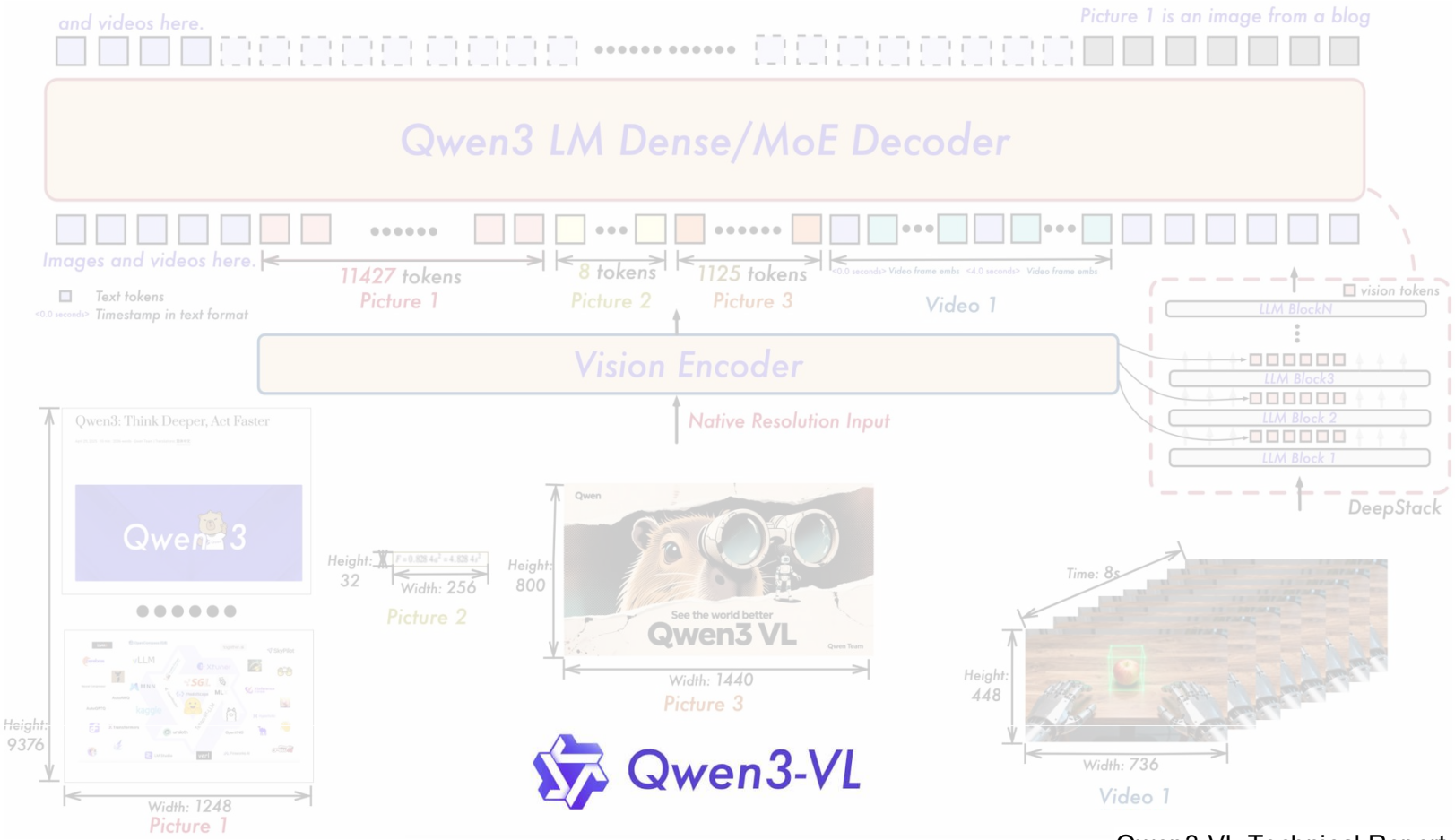
vs

Fully Open Source (can repro training)

Open Weight (can download + run locally)

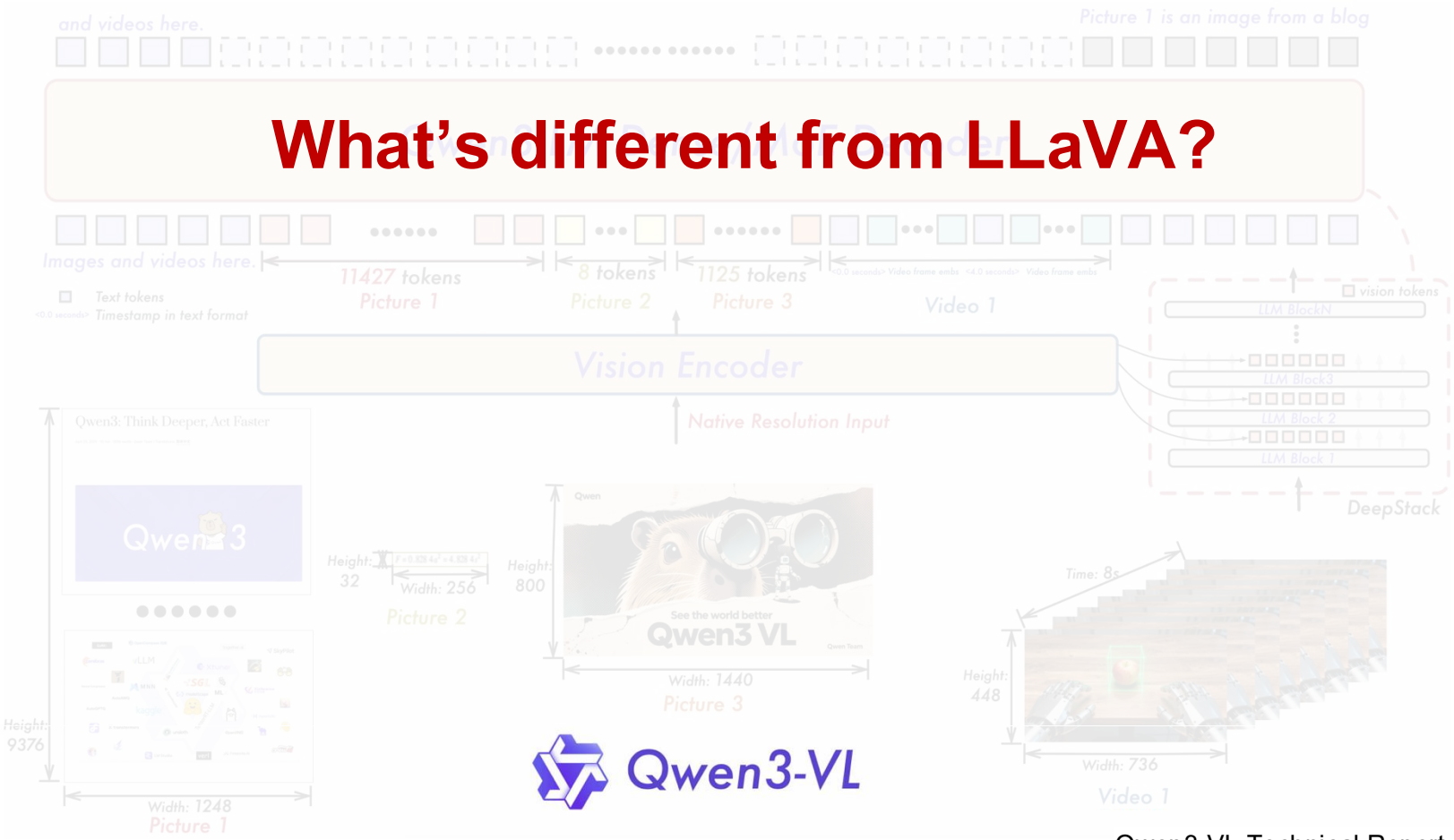
vs

Fully Open Source (can repro training)



Qwen3-VL Technical Report Dec 2025

What's different from LLaVA?

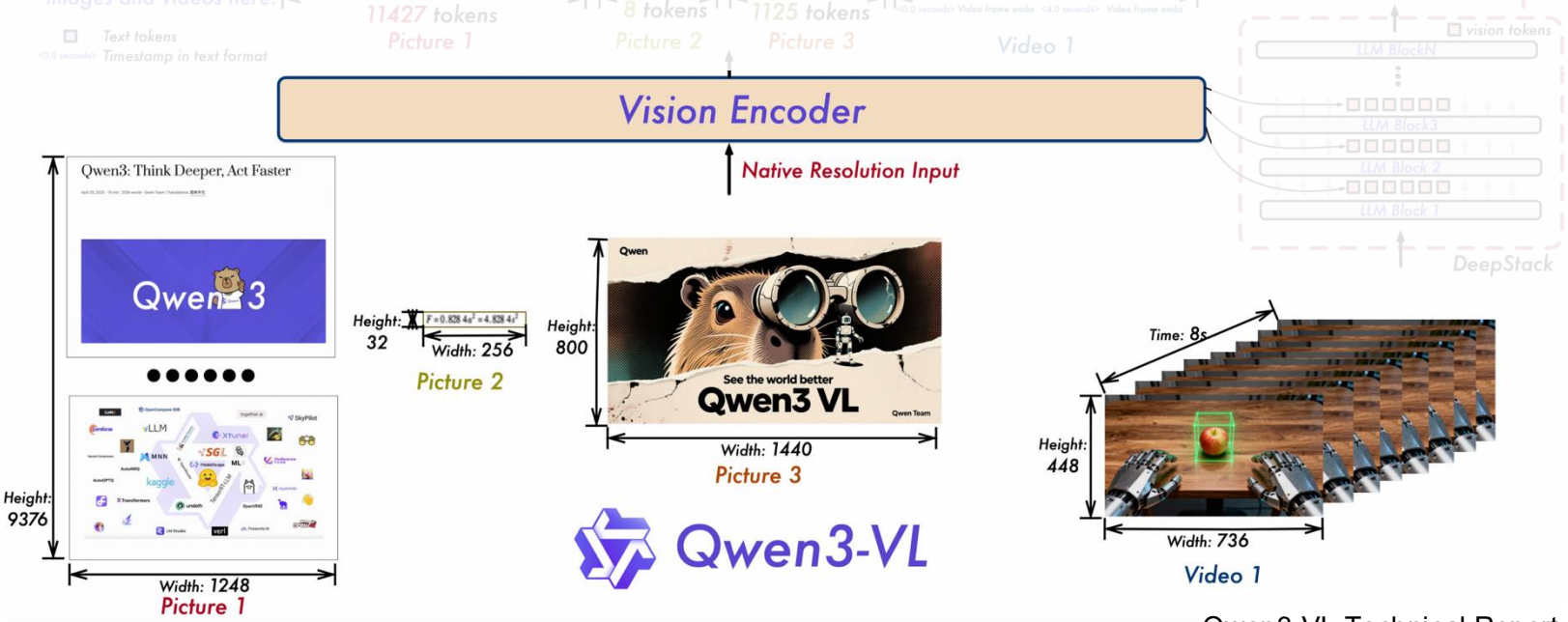


Qwen3-VL Technical Report Dec 2025

1. **Native image resolution → more tokens for larger images**

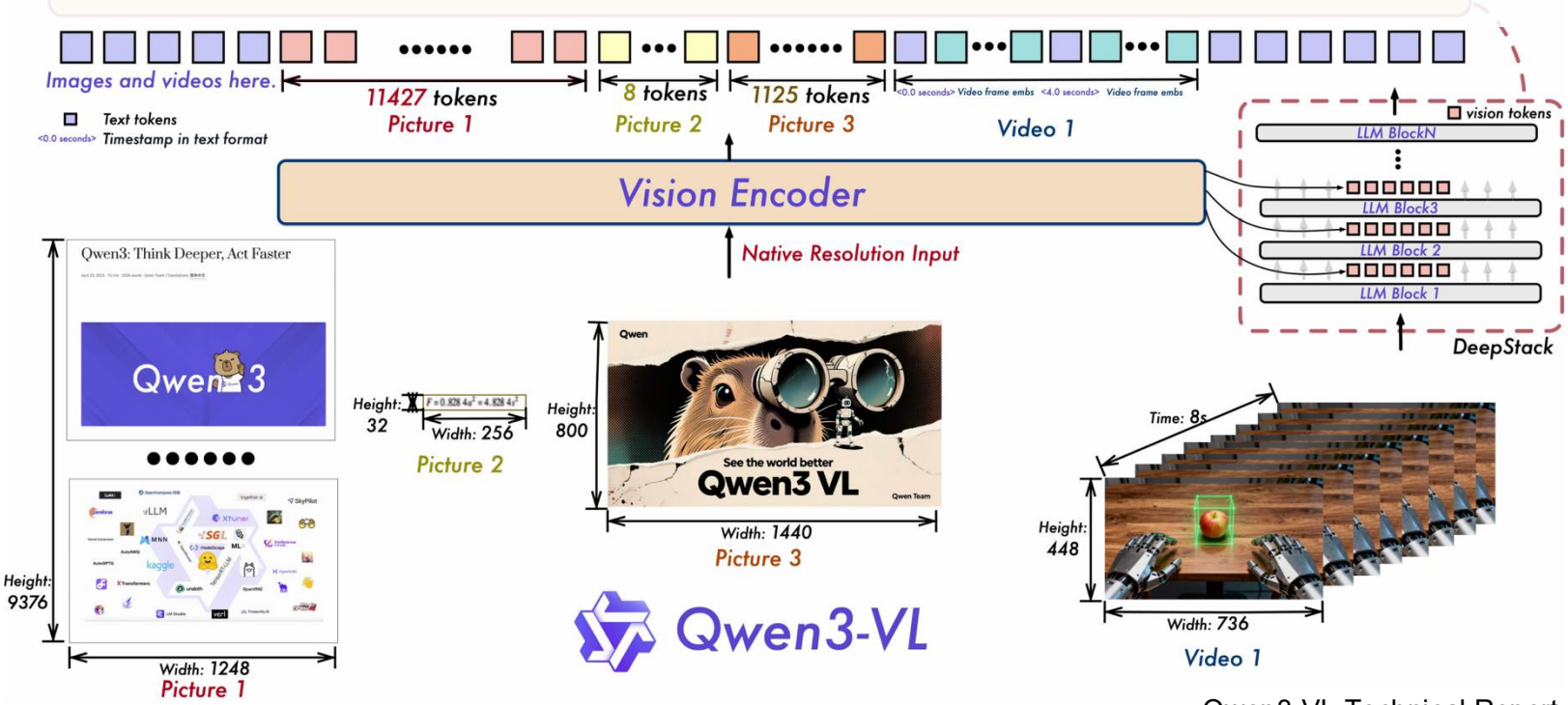
   (use 2D-RoPE for pos-embed to handle varying image sizes)

1. **Use SigLIP-2 Vision Encoder (instead of CLIP)**



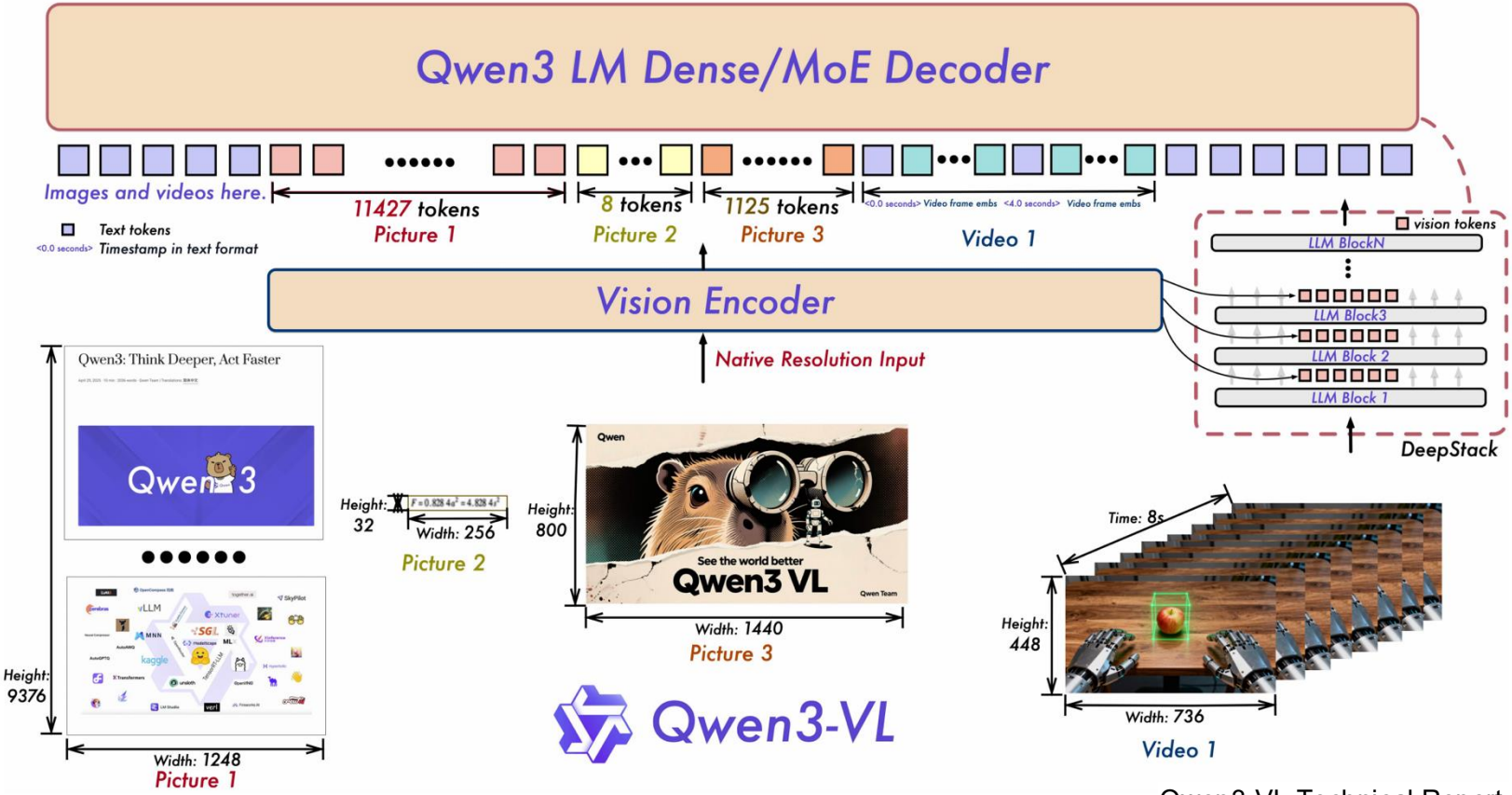
Qwen3-VL Technical Report Dec 2025

3. For videos, frame times are given in text <0.0 seconds>
4. Embeddings from layers 8,16,24 of SigLIP-2 are used



Qwen3-VL Technical Report Dec 2025

5. Four training phases: (1) bridge vision encoder + (3) increase context



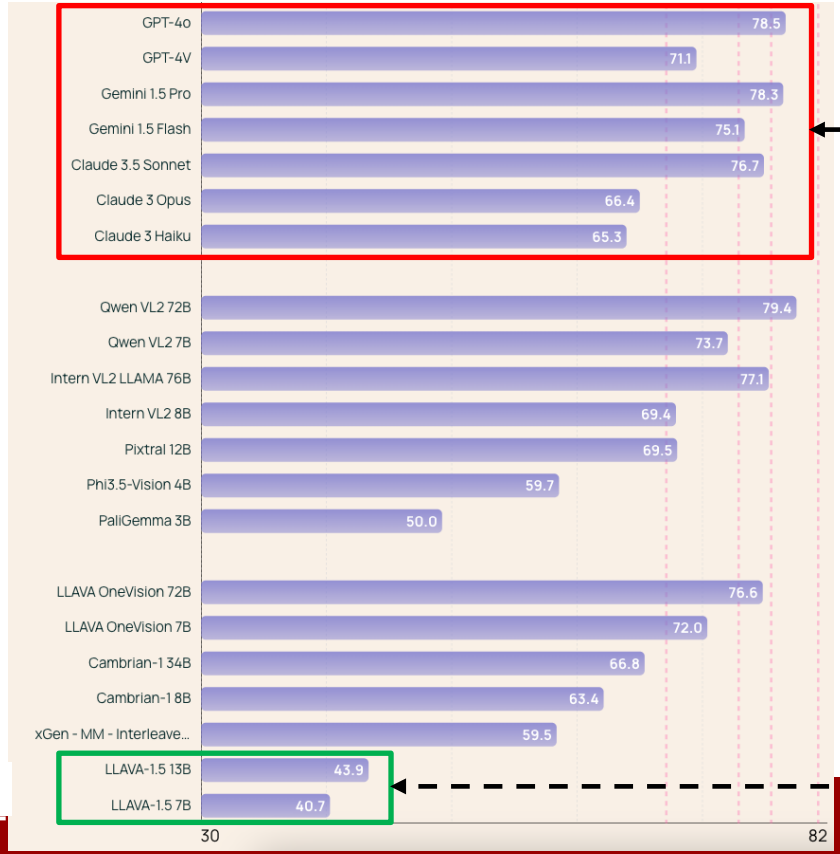
Qwen3-VL Technical Report Dec 2025

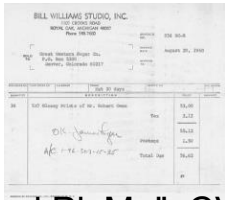
Open Weight (can download + run locally)

vs

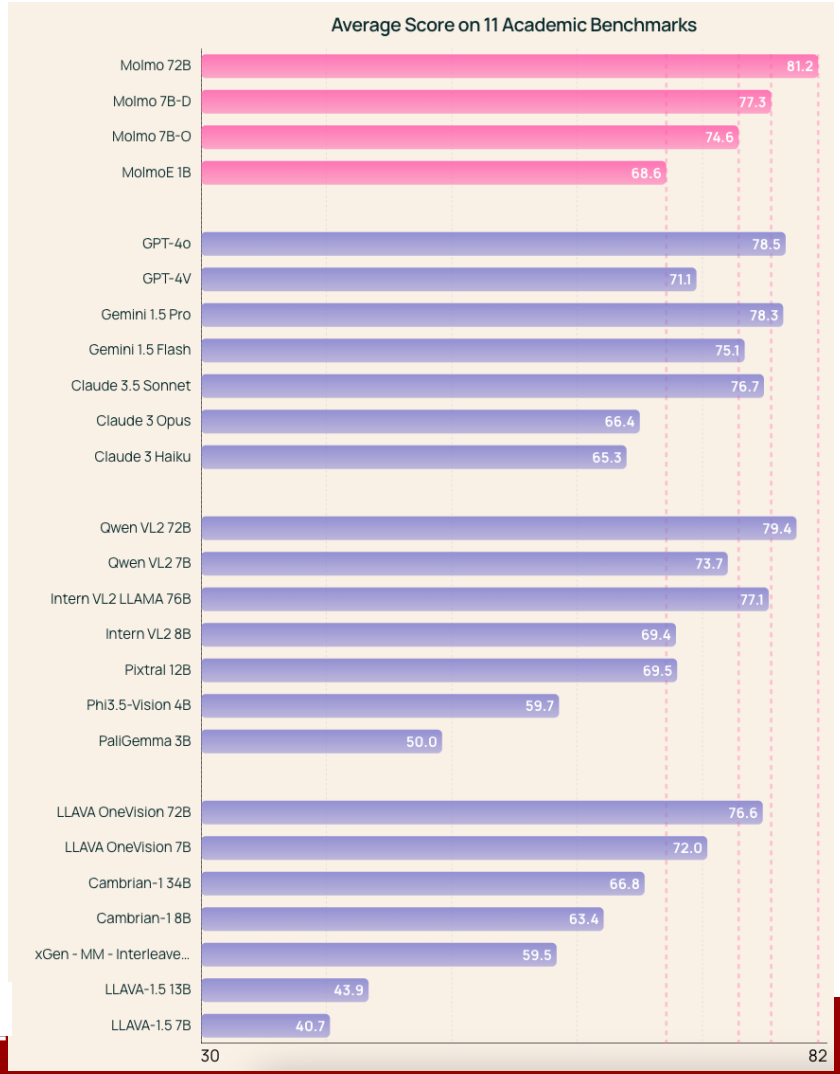
Fully Open Source (can repro everything)

There are open-weight models but they are largely distilled (e.g., from GPT)





Deitke et al. "Molmo and PixMo". CVPR 2025



Open
Weights
Data
Code
Evals

API Only

Open Weights

Distilled

Open

Completely **Open**
Open Weights
Open Data
Open Code
Open Evals

Deitke et al. “Molmo and PixMo”. CVPR 2025

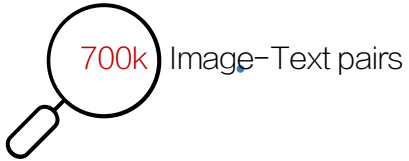
Quality vs quantity tradeoff

LLAMA 3.1V

6 Billion Image-Text pairs

Molmo is trained with

PixMo



Deitke et al. “Molmo and PixMo”. CVPR 2025

Internet data is often incidental
Human annotated data is more intentional

pink, japan, aesthetic image

love this winter picture by person

Deitke et al. “Molmo and PixMo”. CVPR 2025

PixMo data is detailed:

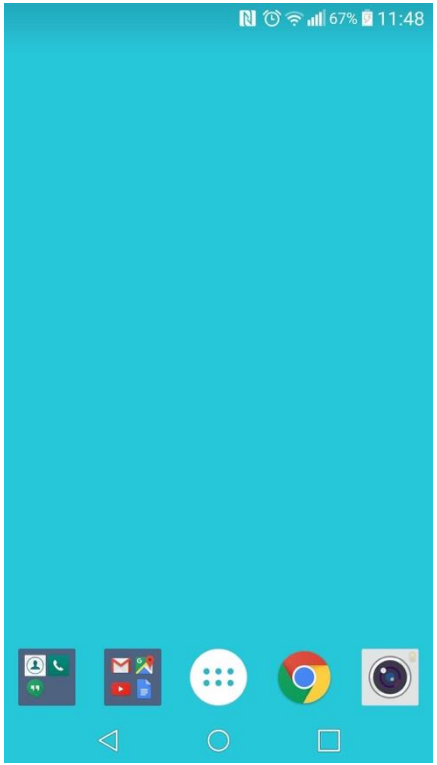
This photograph captures a well-organized work desk set prominently in the middle of the frame. The desk is large and rectangular, made from a polished, rich wood that spans horizontally across the image. Its structure is supported by four distinctive A-shaped legs, adding an elegant touch. On the desk, 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.

Deitke et al. "Molmo and PixMo". CVPR 2025

Collecting dense captions is hard!!!



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

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

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

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

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

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

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

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

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

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

Annotators were asked to speak for 60 to 90 seconds about an image

Authors automatically convert speech into text for pretraining

Deitke et al. “Molmo and PixMo”. CVPR 2025

Foundation Models

Language	Classification	LM + Vision	Chaining
ELMo	**CLIP**	**LLaVA**	**LMs + CLIP**
BERT	**CoCa**	**Flamingo**	**Visual Programming**
GPT		GPT	
T5		Gemini	
		Qwen 3.5 VL	

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

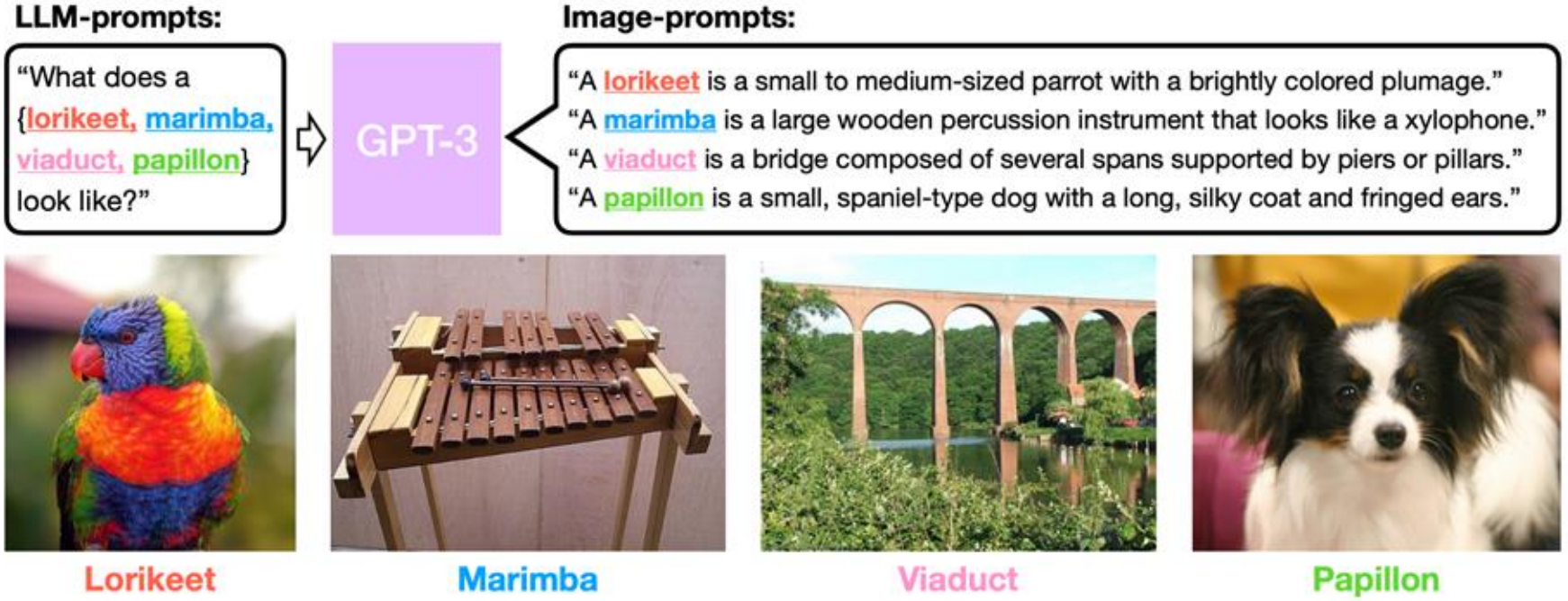
1. Get an LLM to generate a description.
2. Classify using the description

"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)



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:

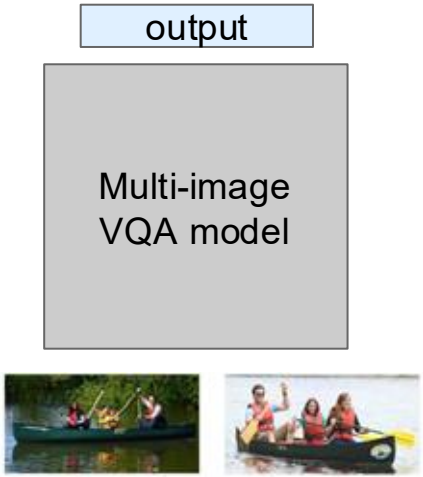
RIGHT:

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
```

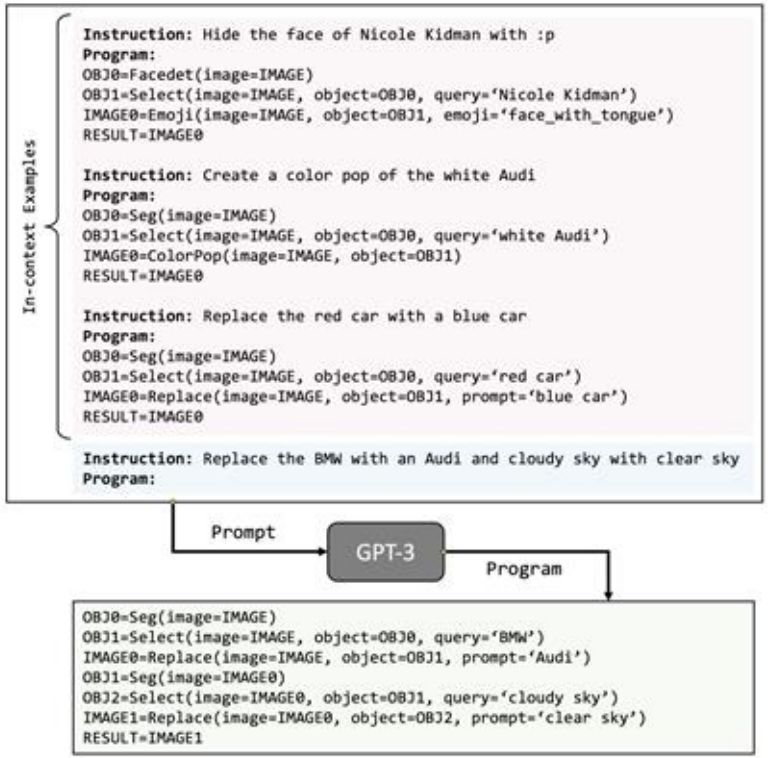
General to 2 images now, but not beyond that

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

VisProg (visual programming)

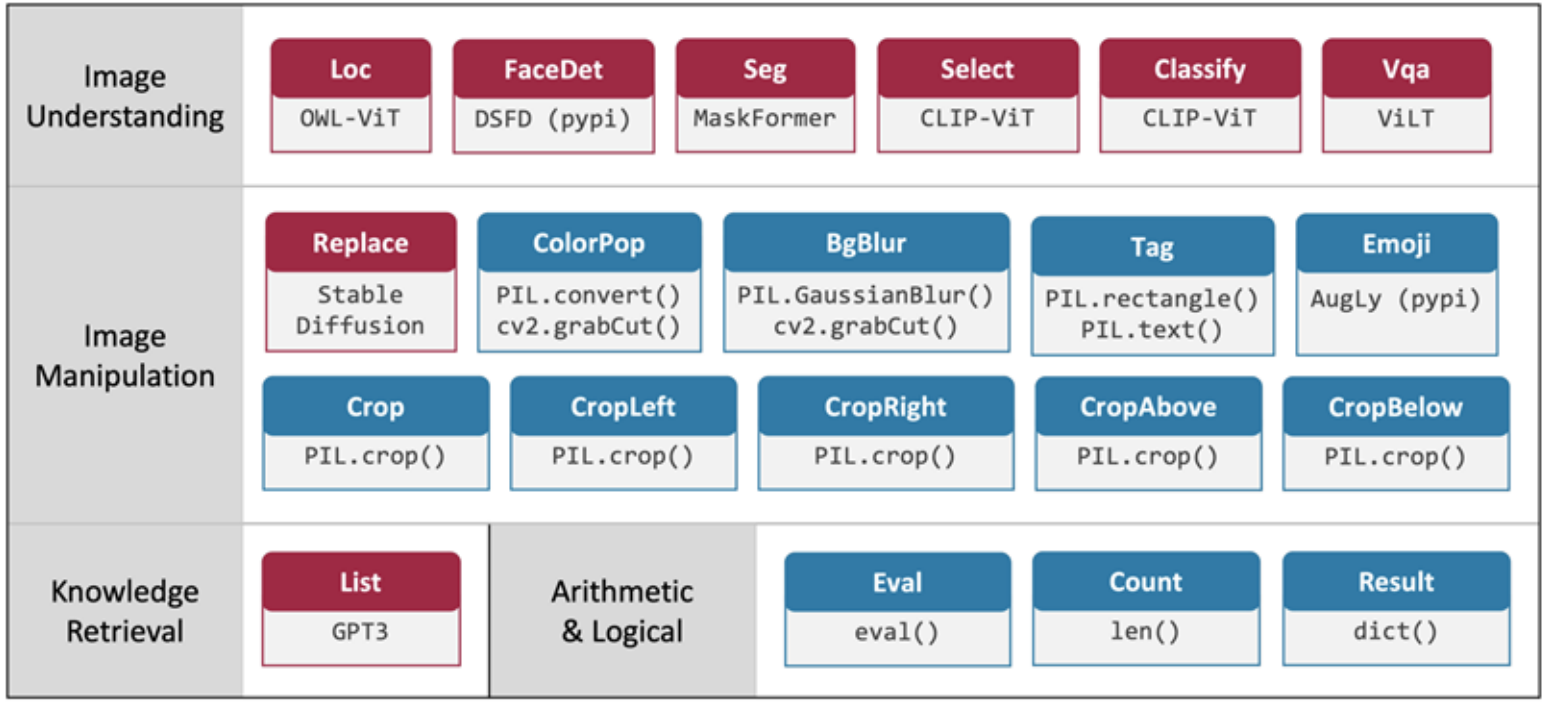
Gupta et al “Visual Programming: Compositional visual reasoning without training”. 2023.

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

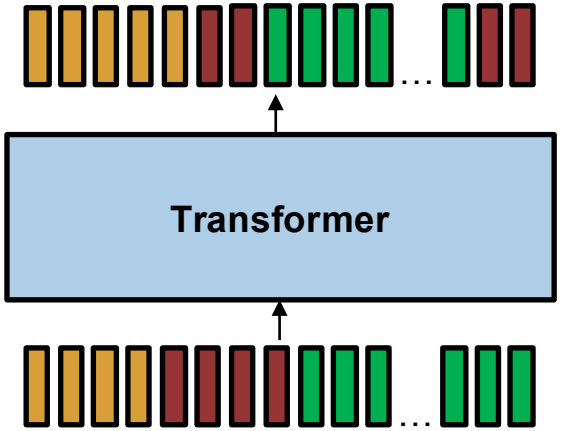
Program:

```
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.

Final Note: Omni Models (Beyond Vision + Language)

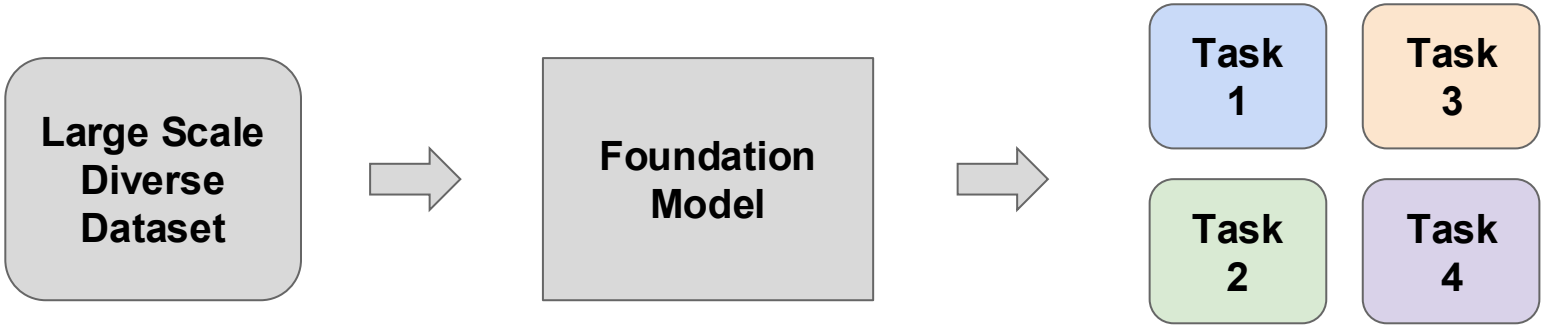
Started with GPT-4o in 2024
Recently, Thinking Machines + Gemini have introduced their own models as well



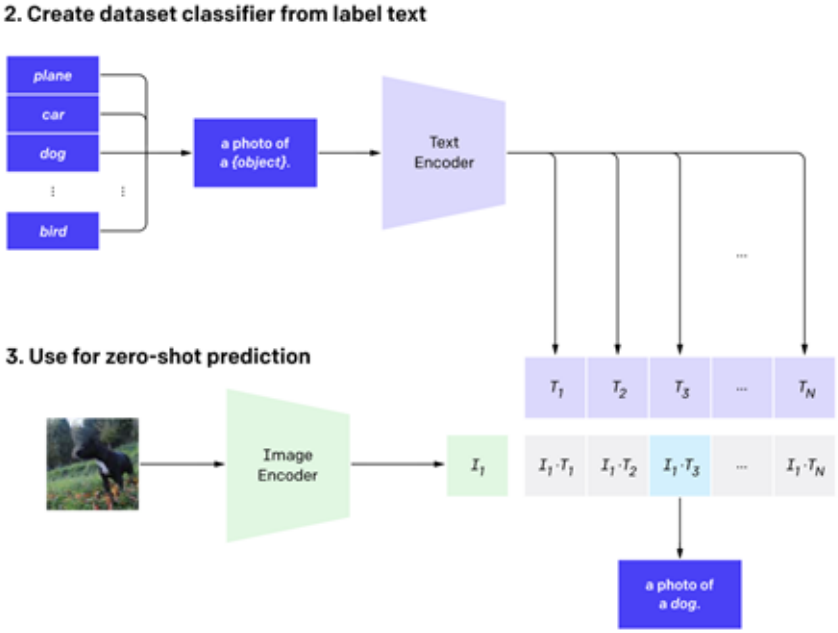
Train a model to unput and output in **text**, **audio**, and **video**.

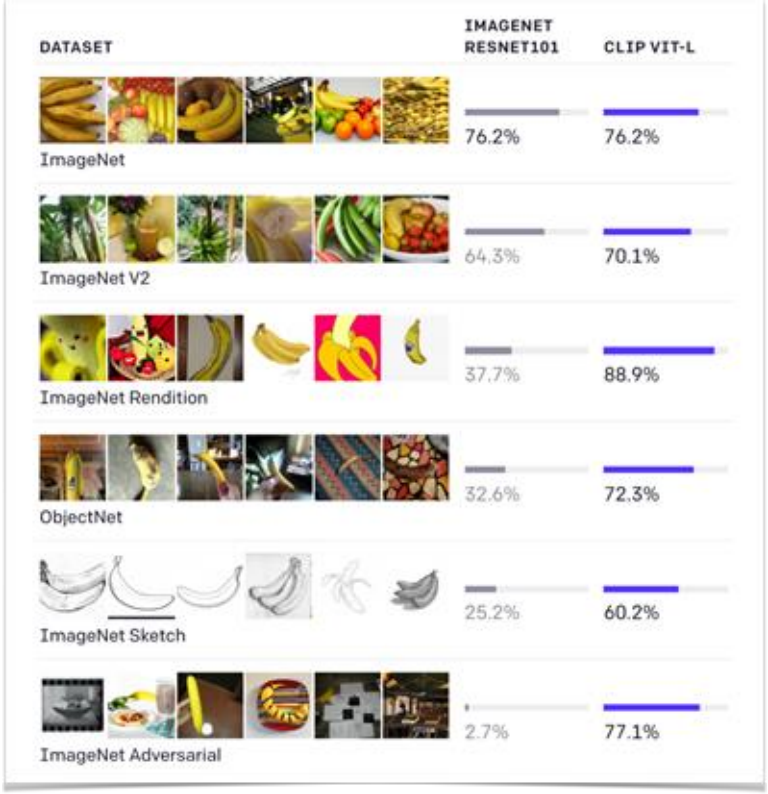
[Thinking Machines post from 2 weeks ago]
[Google blog post from 3 hours ago]

Summary

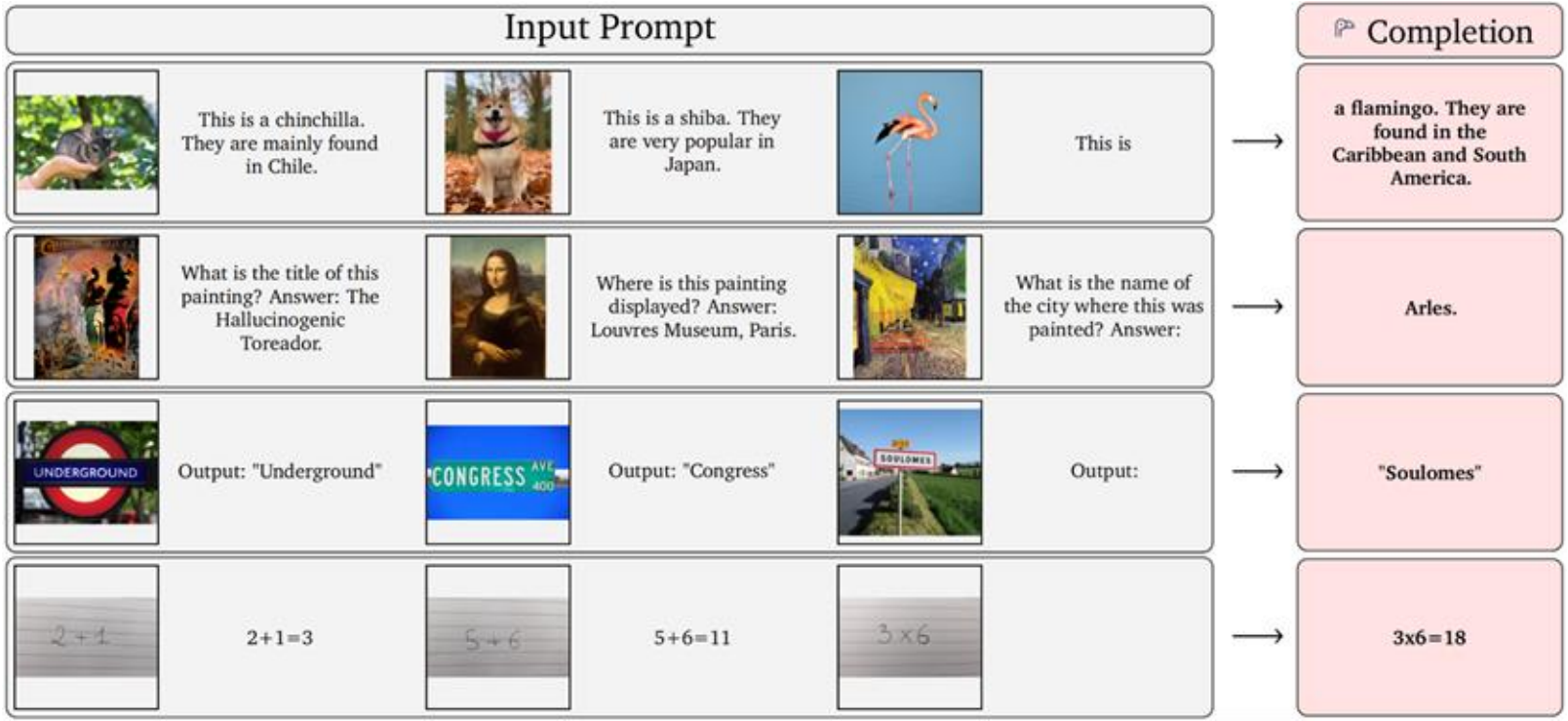

Large Scale Diverse Dataset
Foundation Model
Task 1
Task 3
Task 2
Task 4

Summary

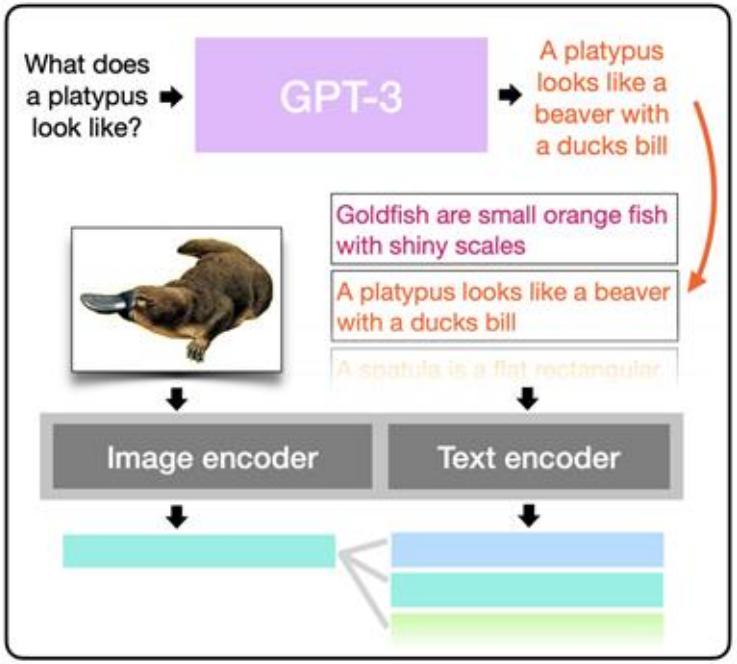

2. Create dataset classifier from label text
plane
car
dog
bird
a photo of a {object}.
Text Encoder
3. Use for zero-shot prediction
Image Encoder
a photo of a dog.

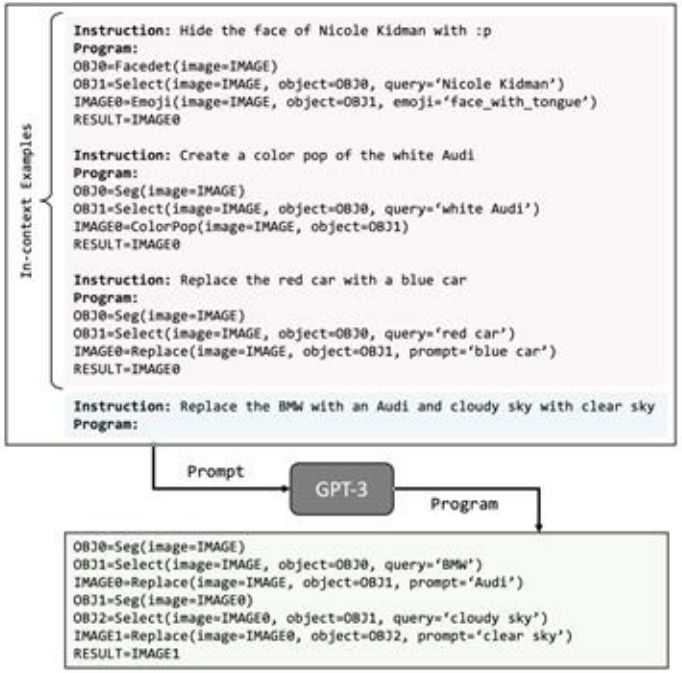

DATASET
IMAGENET RESNET101
CLIP VIT-L
ImageNet
76.2%
76.2%
ImageNet V2
64.3%
70.1%
ImageNet Rendition
37.7%
88.9%
ObjectNet
32.6%
72.3%
ImageNet Sketch
25.2%
60.2%
ImageNet Adversarial
2.7%
77.1%

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
Output: "Congress"
Output:
"Soulomes"
2+1
2+1=3
5+6
5+6=11
3x6
3x6=18

Summary


What does a platypus look like?
GPT-3
A platypus looks like a beaver with a ducks bill
Goldfish are small orange fish with shiny scales
A platypus looks like a beaver with a ducks bill
Image encoder
Text encoder


In-context Examples
Instruction: Hide the face of Nicole Kidman with :p
Program:
OBJ0=Facedet(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='Nicole Kidman')
IMAGE0=Emoji(image=IMAGE, object=OBJ1, emoji='face_with_tongue')
RESULT=IMAGE0
Instruction: Create a color pop of the white Audi
Program:
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='white Audi')
IMAGE0=ColorPop(image=IMAGE, object=OBJ1)
RESULT=IMAGE0
Instruction: Replace the red car with a blue car
Program:
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='red car')
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt='blue car')
RESULT=IMAGE0
Instruction: Replace the BMW with an Audi and cloudy sky with clear sky
Program:
Prompt
GPT-3
Program
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='BMW')
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt='Audi')
OBJ1=Seg(image=IMAGE0)
OBJ2=Select(image=IMAGE0, object=OBJ1, query='cloudy sky')
IMAGE1=Replace(image=IMAGE0, object=OBJ2, prompt='clear sky')
RESULT=IMAGE1

Next time: World Models