Vision and VLMs

CS6960 MultiModal LLM Agents

Kenneth Marino

Slide credits: Trevor Darrell, XuDong Wang, Ryan Tabrizi, Guanghong Tao, Matt Gormley, Abhinav Gupta, Justin Johnson others

Announcements

- HW0 due this Friday
  - It’s really short and easy
  - See Syeda’s note in announcements / read the new instructions
  - Basically just asking you to read the intro unit to the HF Agents tutorial and write some boilerplate code
- Material now being posted on Canvas Modules
  - Lecture Notes
  - Uploading Recordings to Youtube
  - Student pres: upload your slides (but we won’t record)
- Waitlist (you should be able to join) - can give out codes

Recommended Readings

- Unfortunately a lot less material for this stuff than RL/LLMs
- Some good keynotes
  - https://www.youtube.com/watch?v=PdsKNtEofFY
  - http://incompleteideas.net/book/RLbook2020.pdf
- Courses with notes
  - Berkeley Large Scale Vision and Language Models https://cs294-43-fall2024.pages.dev/
  - If you need to understand these models in detail
  - Stanford Deep CV course: https://cs231n.stanford.edu/schedule.html
  - Matt Gormley's Generative Model course: https://www.cs.cmu.edu/~mgormley/courses/10423-f24/
  - More general computer vision course
- Papers of some influential VLMs
  - Flamingo: https://arxiv.org/abs/2204.14198
  - CLIP: https://arxiv.org/abs/2103.00020
  - LLAVA: https://arxiv.org/abs/2304.08485
  - Closed source VLMs harder to find stuff for, but open source technical reports (Llama, Qwen, AI2 Olmo) talk a lot about these details

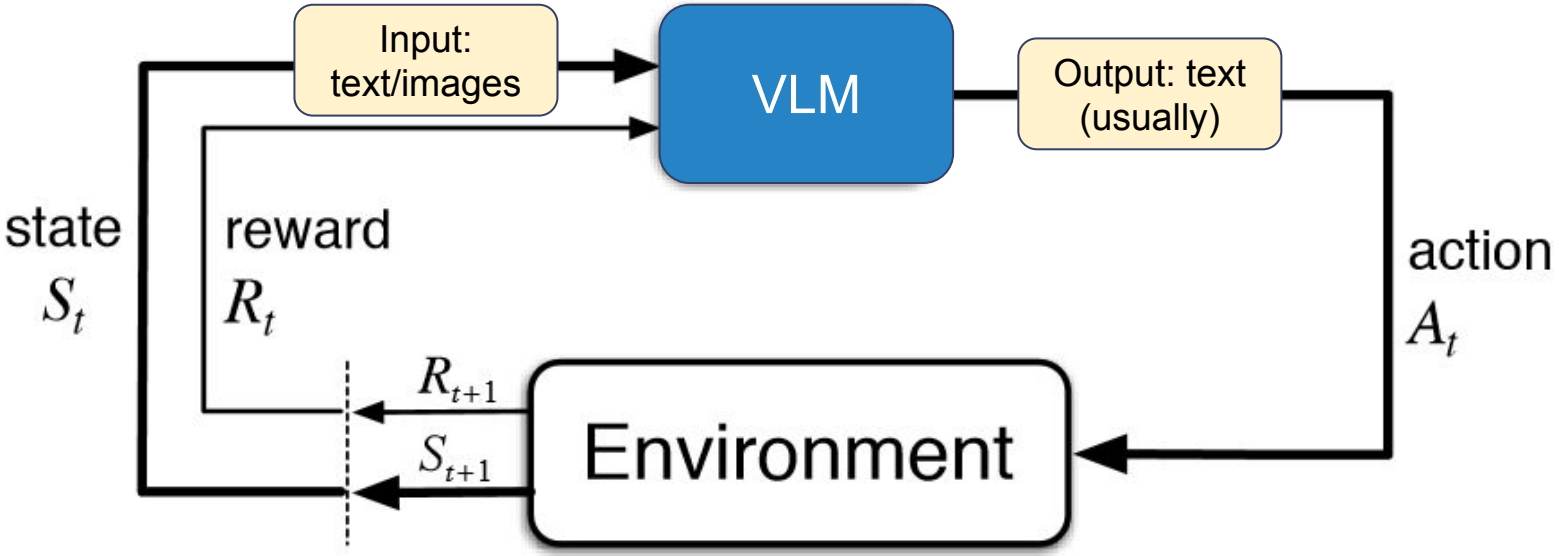
Warning (again)

- Again I can’t possibly cover everything about VLMs in a single lecture
- I’m also talking a lot about vision and visual recognition in general (which is its own course)

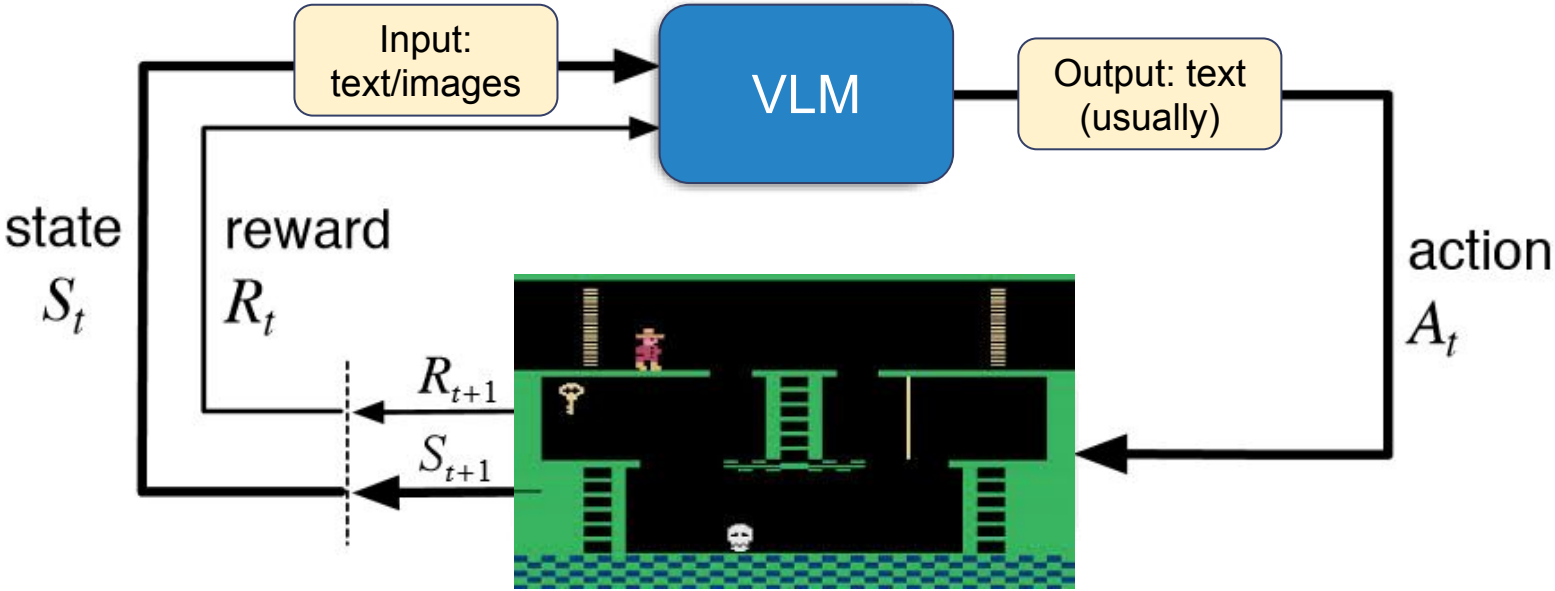
Any Questions

Recall: What does VLM agent actually look like?

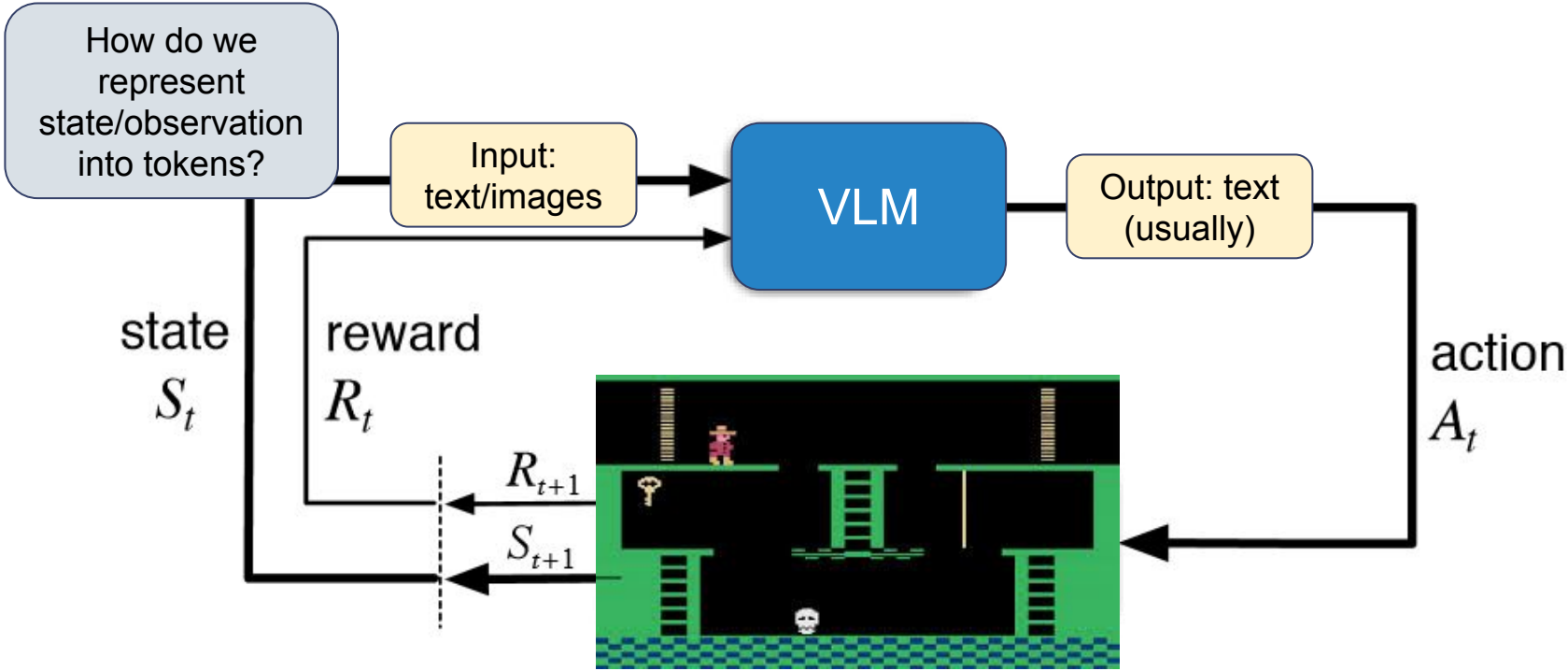
Recall: What does VLM agent actually look like?



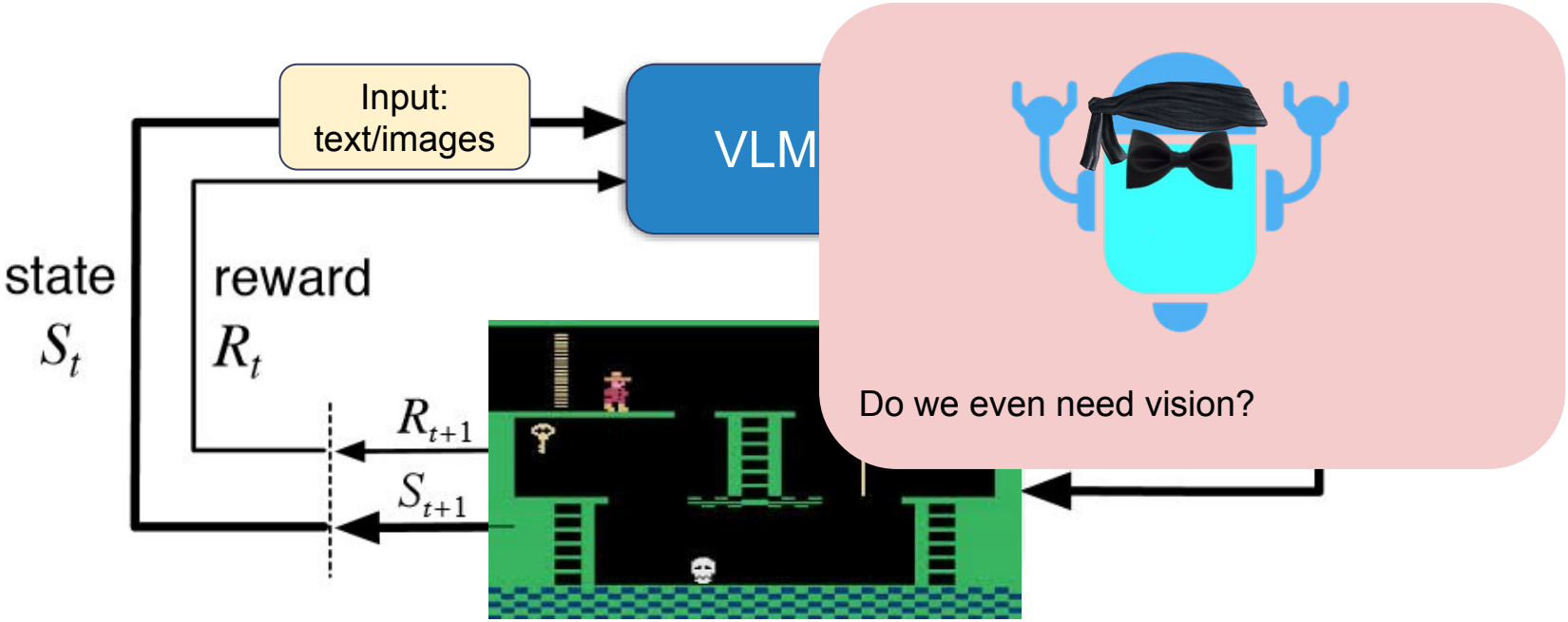
Recall: What does VLM agent actually look like?



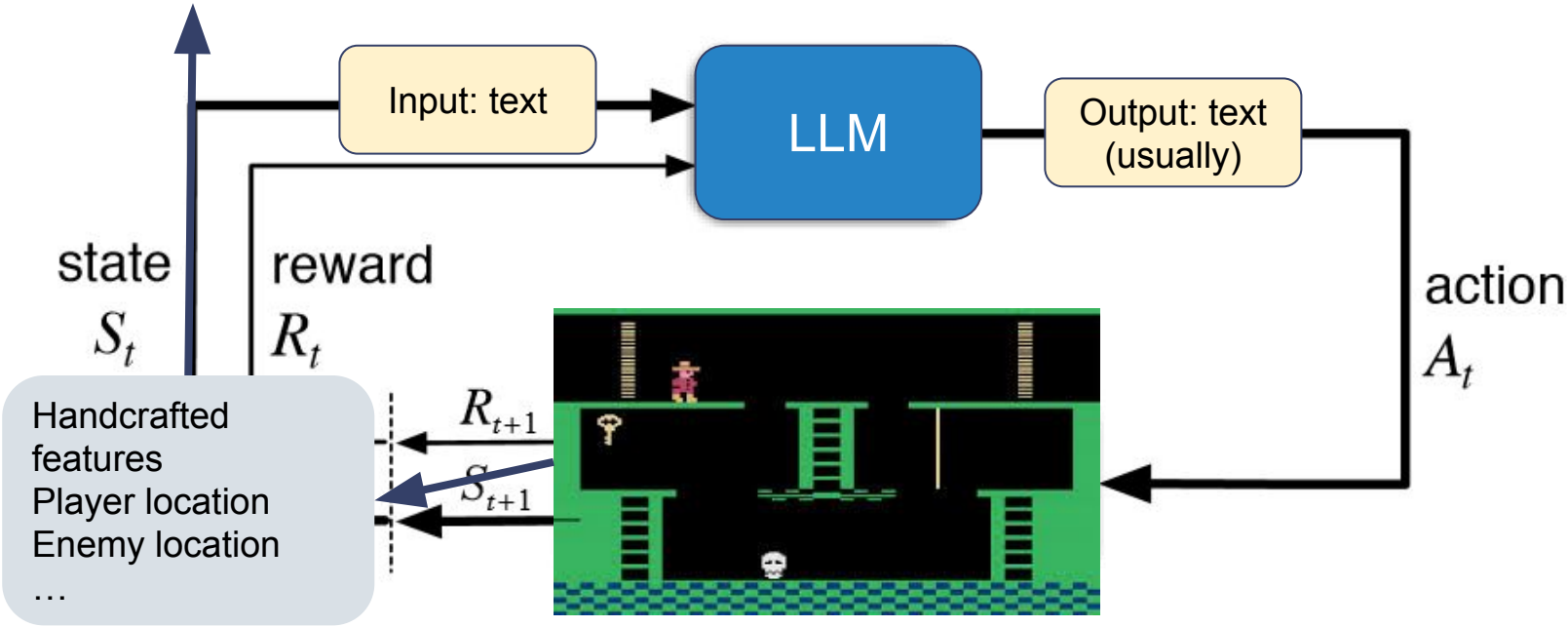
Recall: What does VLM agent actually look like?



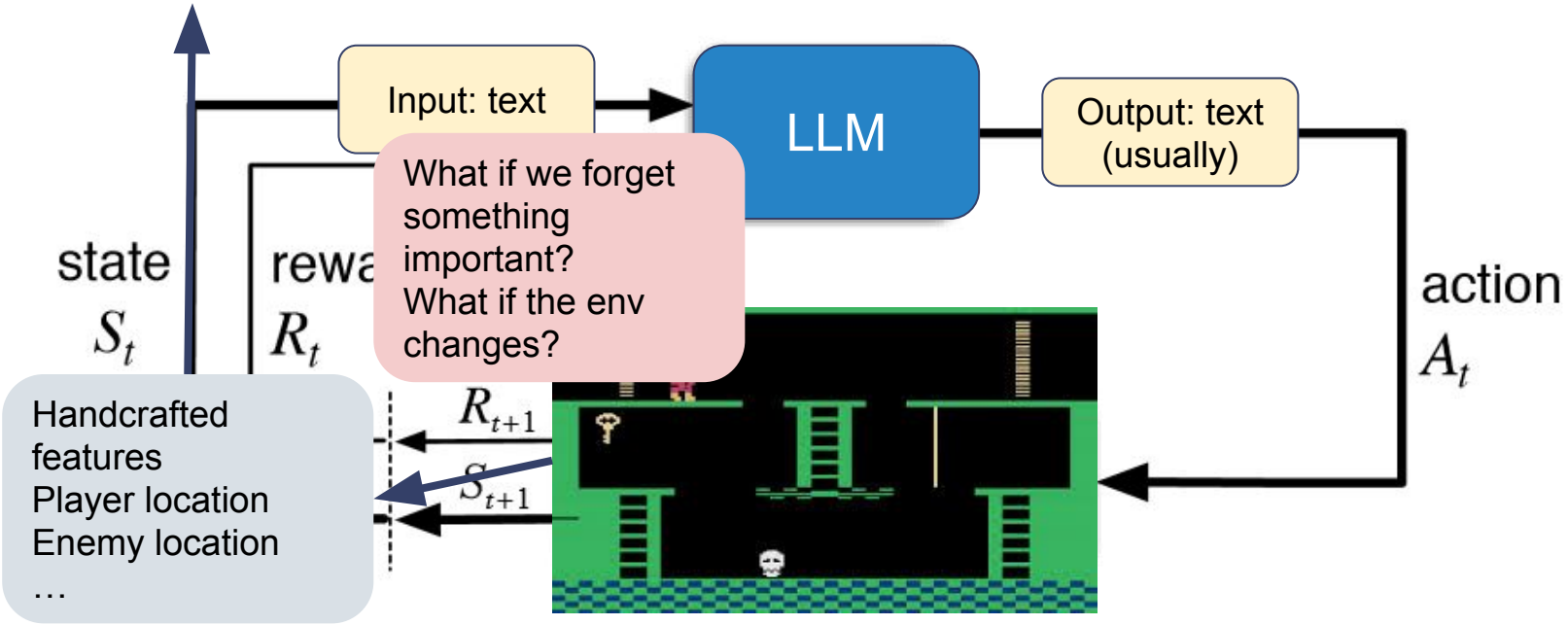
Recall: What does VLM agent actually look like?



Recall: What does VLM agent actually look like?

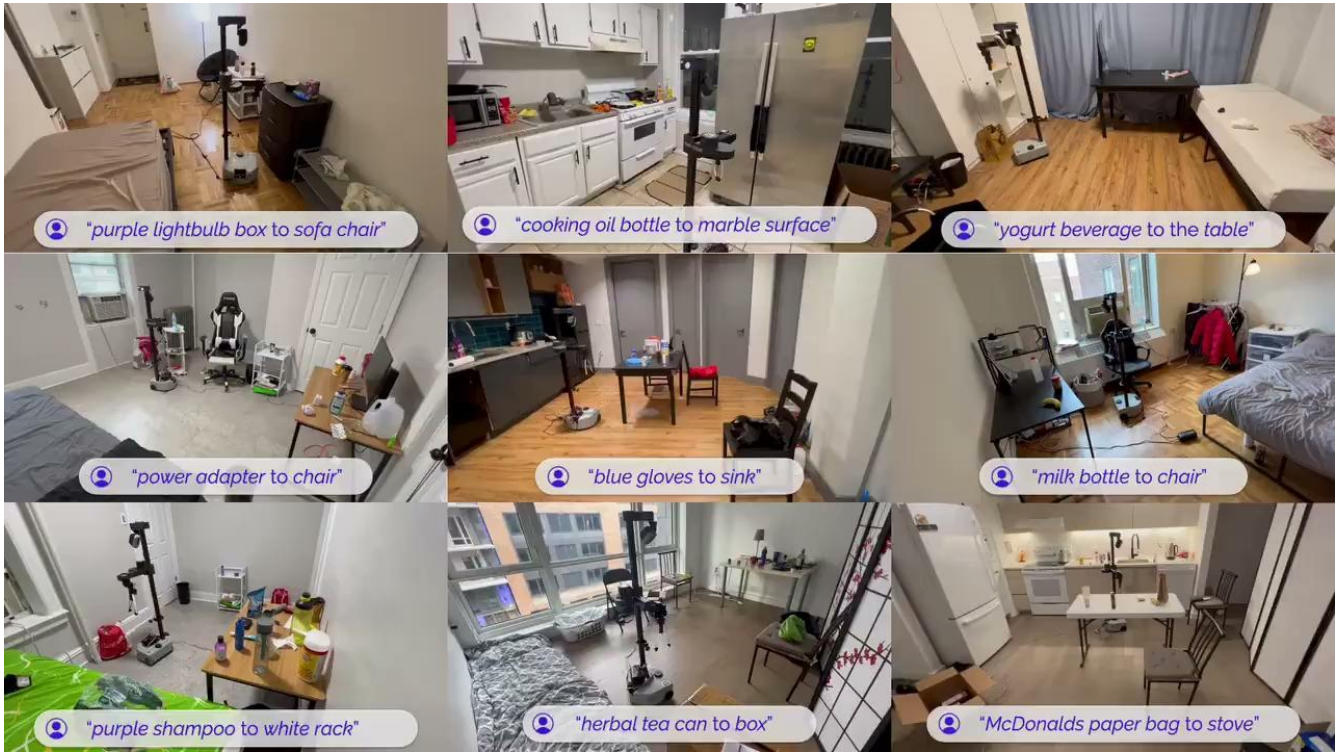


Recall: What does VLM agent actually look like?



Robotics

How would we describe everything in this scene using language?



Computer Vision

In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to “spend the summer linking a camera to a computer and getting the computer to describe what it saw”. We now know that the problem is slightly more difficult than that. (Szeliski 2009, Computer Vision)

Computer Vision: Goal

To create autonomous systems that "understand" visual data

Computer Vision: Goal

To create autonomous systems that "understand" visual data

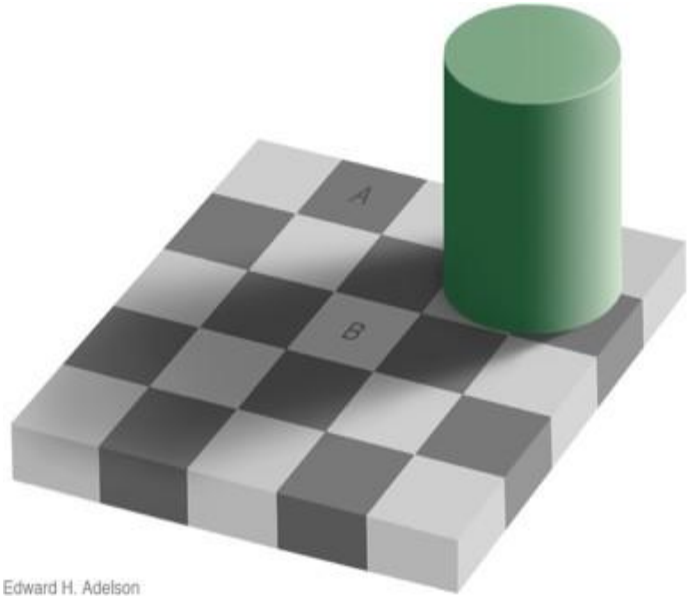
What does it mean to understand?

Early days of Computer Vision

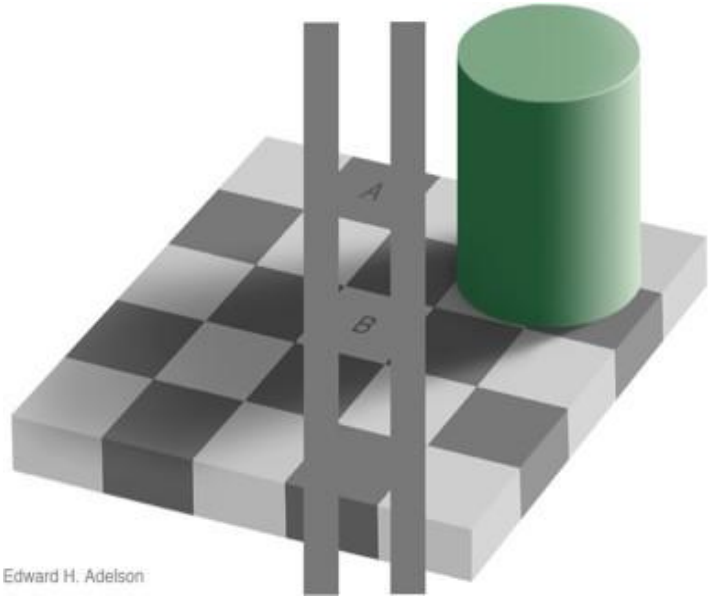
Answer #1: pixel of brightness 43 at position (124,54) …and depth .7 meters

Machine Vision: deals with how cameras work, how scenes create 2D images

Perception != Measurement



Edward H. Adelson



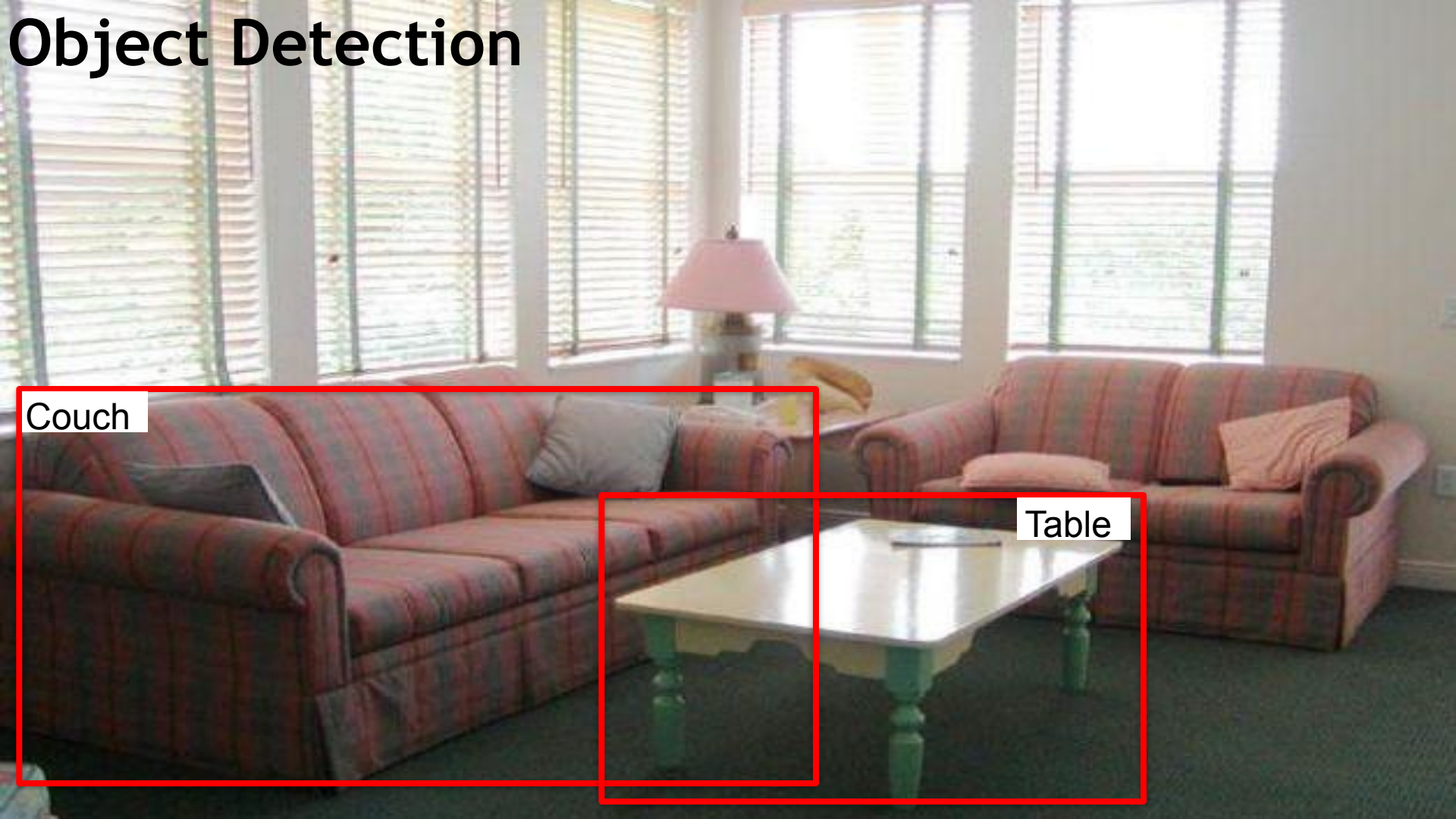
Edward H. Adelson

Computer Vision: Goal

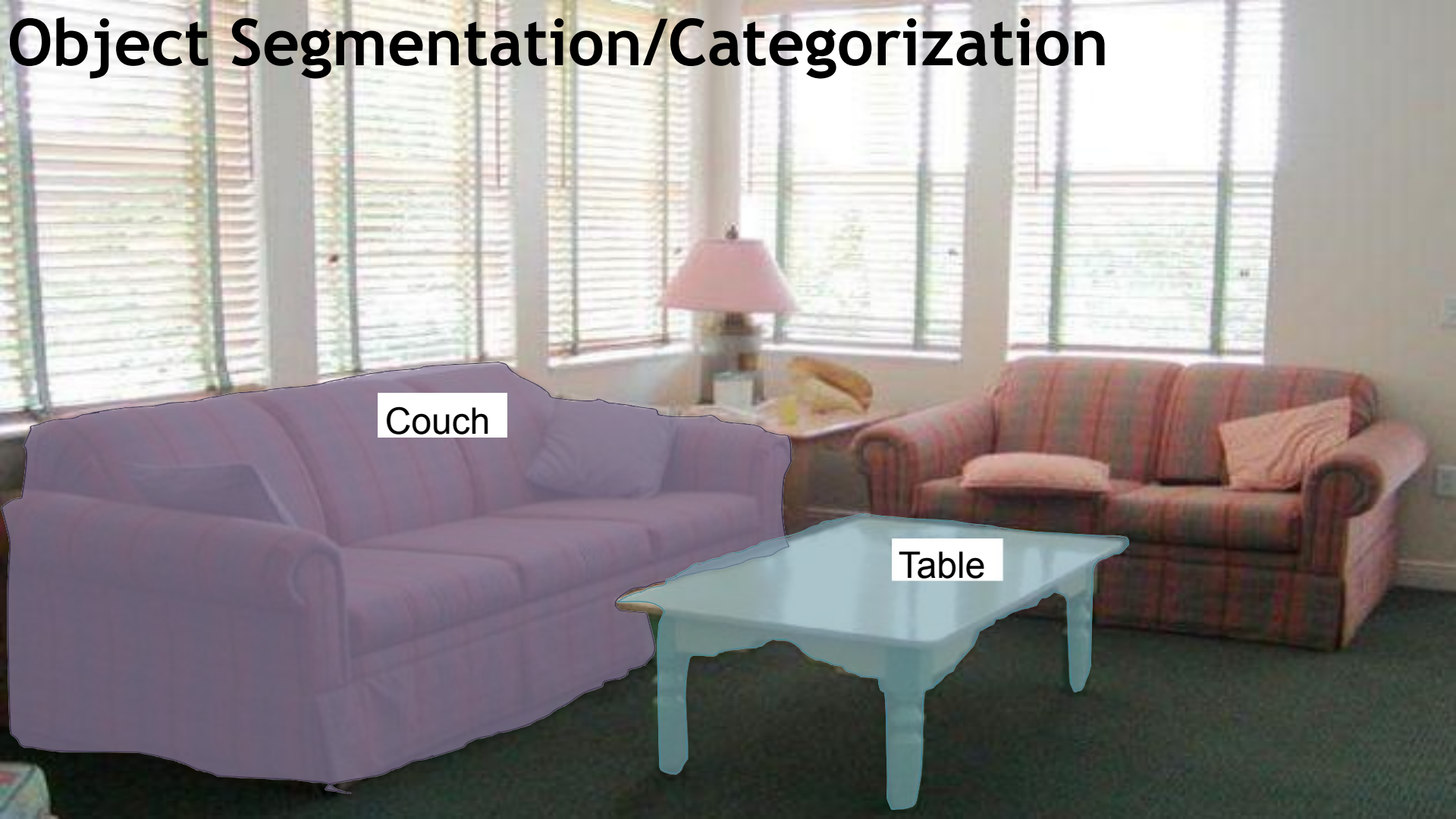
The goals of computer vision (what + where) are in terms of what humans care about

Image Classification/ Scene Recognition

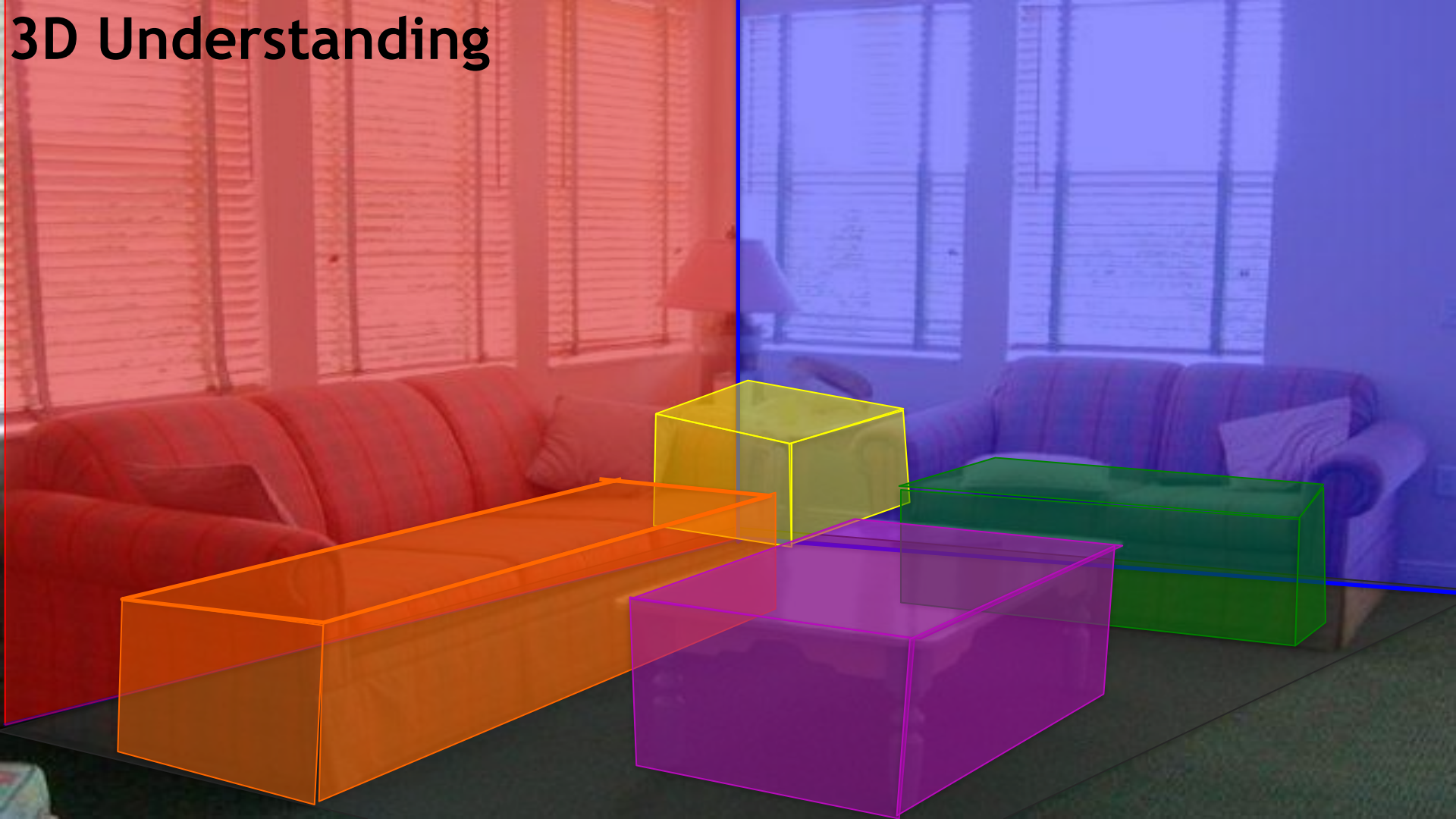
Object Detection



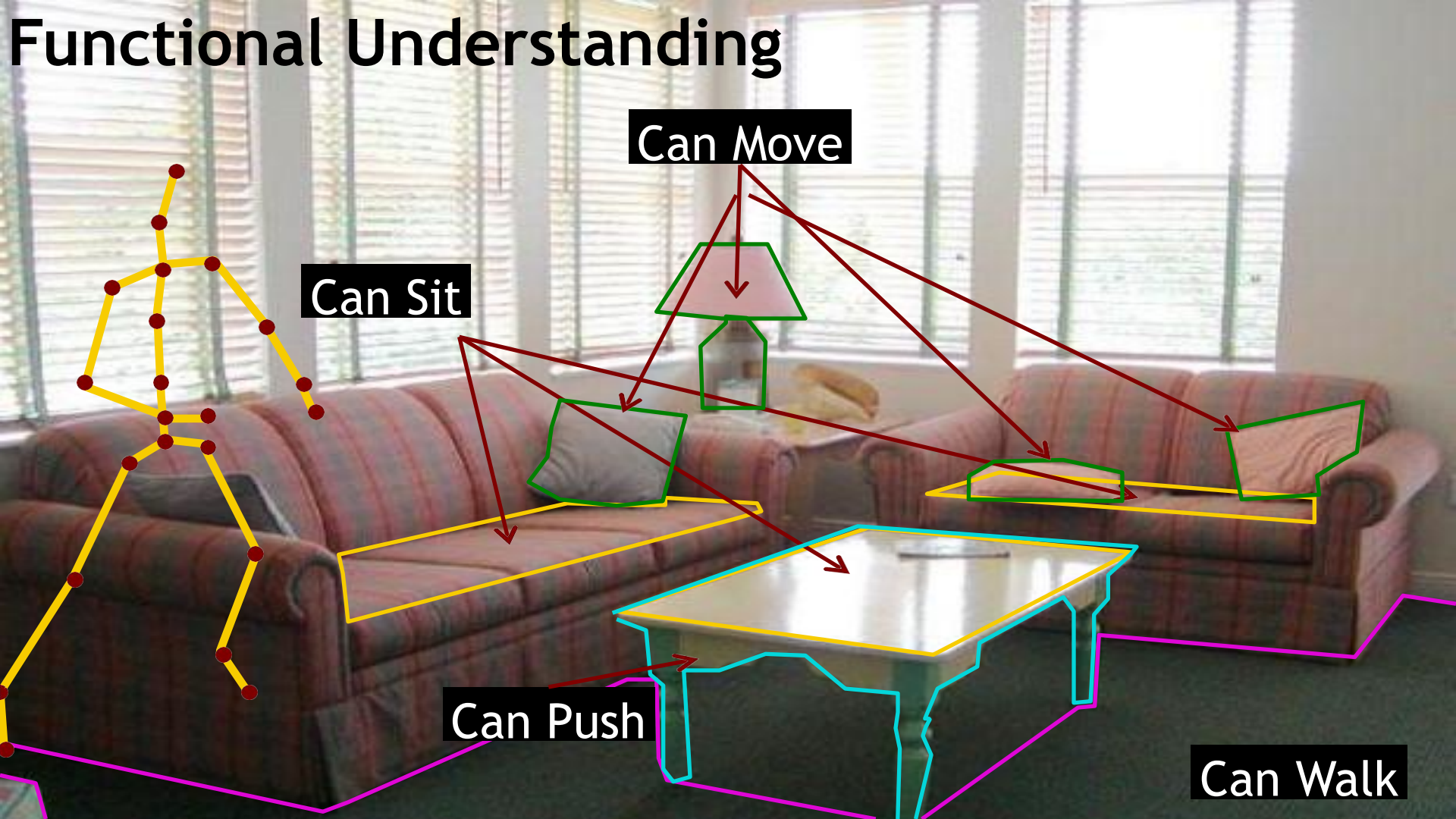
Object Segmentation/Categorization



3D Understanding



Functional Understanding



Computer Vision

- Is it possible to create models which can know all of this information?
- VLMs: Yes*

Computer Vision

- Is it possible to create models which can know all of this information?
- VLMs: Yes*

* VLMs have not solved all vision problems - have plenty of blind spots

Vision Language Models

Object Localization

Is one cat behind another?

Segmentation

Segment: striped cat

Visual QA

What is the breed of these cats?

Learning w/ Instructions

Striped cats are called tabby cats. What's the breed of the cats in the image?

Vision Language Model

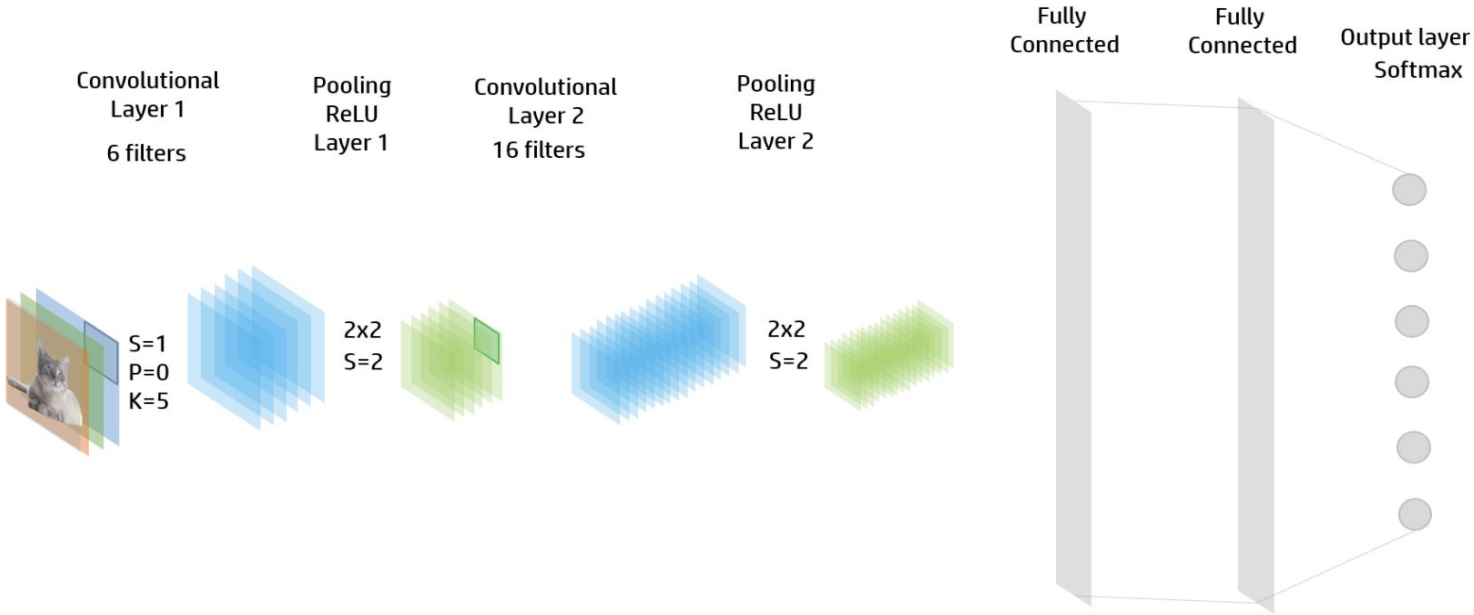
Yes, one cat is behind the other in the image. The cat in the back us facing the camera, while the cat in front it is facing away from the camera.

The cats in the image appear to be domestic shorthair

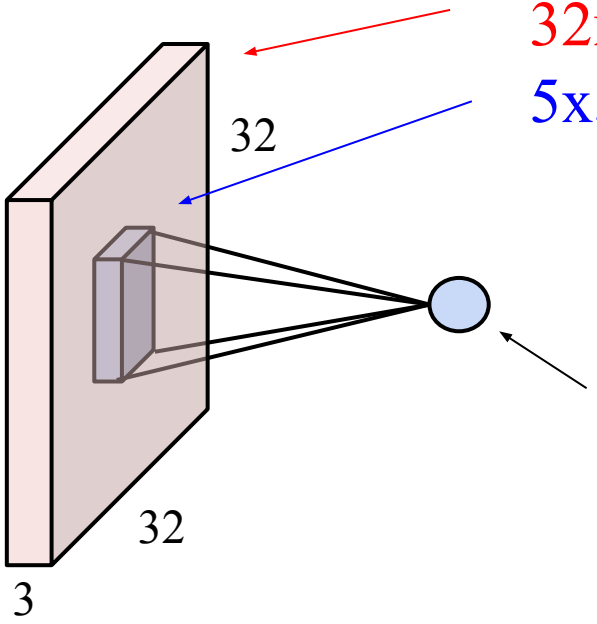
The cats in the image are tabby cats. Tabby cats are a common domestic cat bread and are characterized by their distinctive coat pattern, stripes on the body and a ringed tail.

Slide credit: XuDong Wang

Convolutional Neural Networks



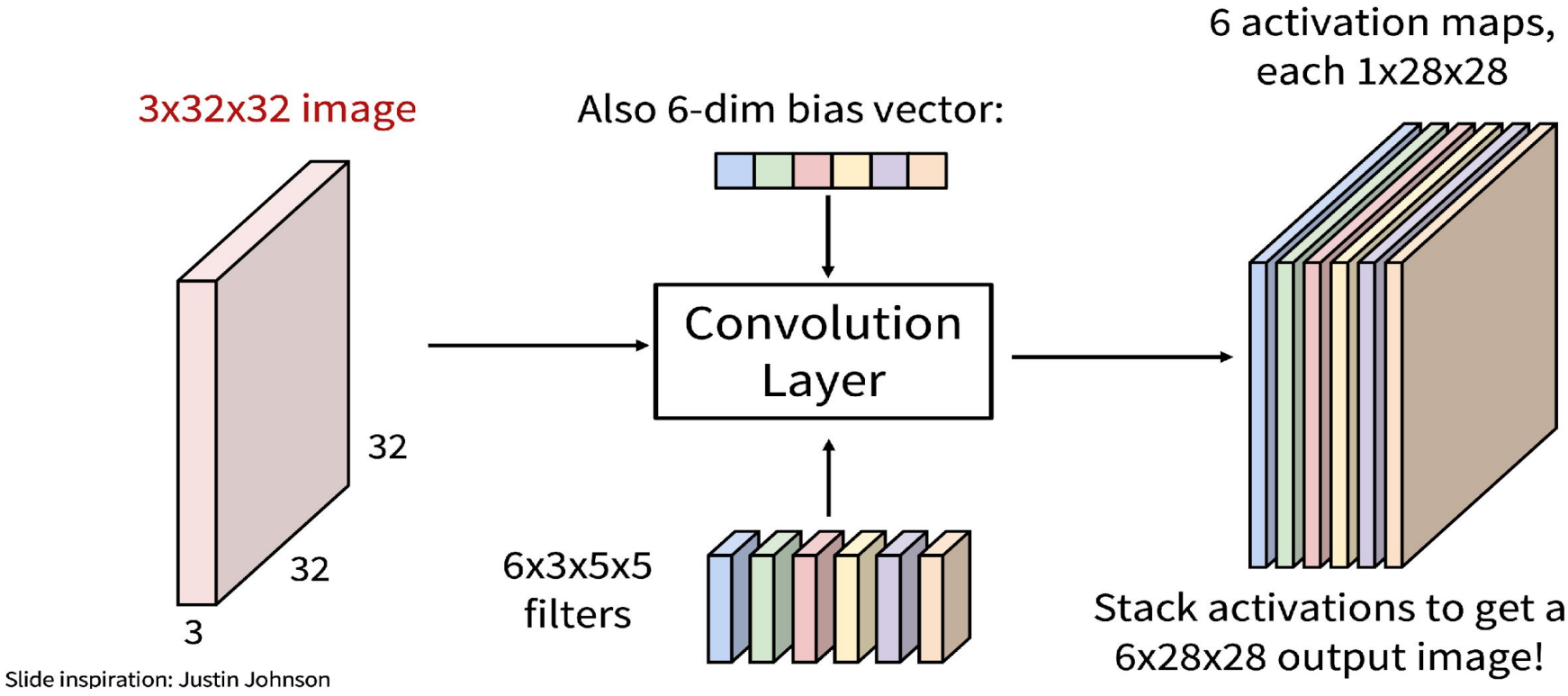
Main Component: Convolutional Layers



1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

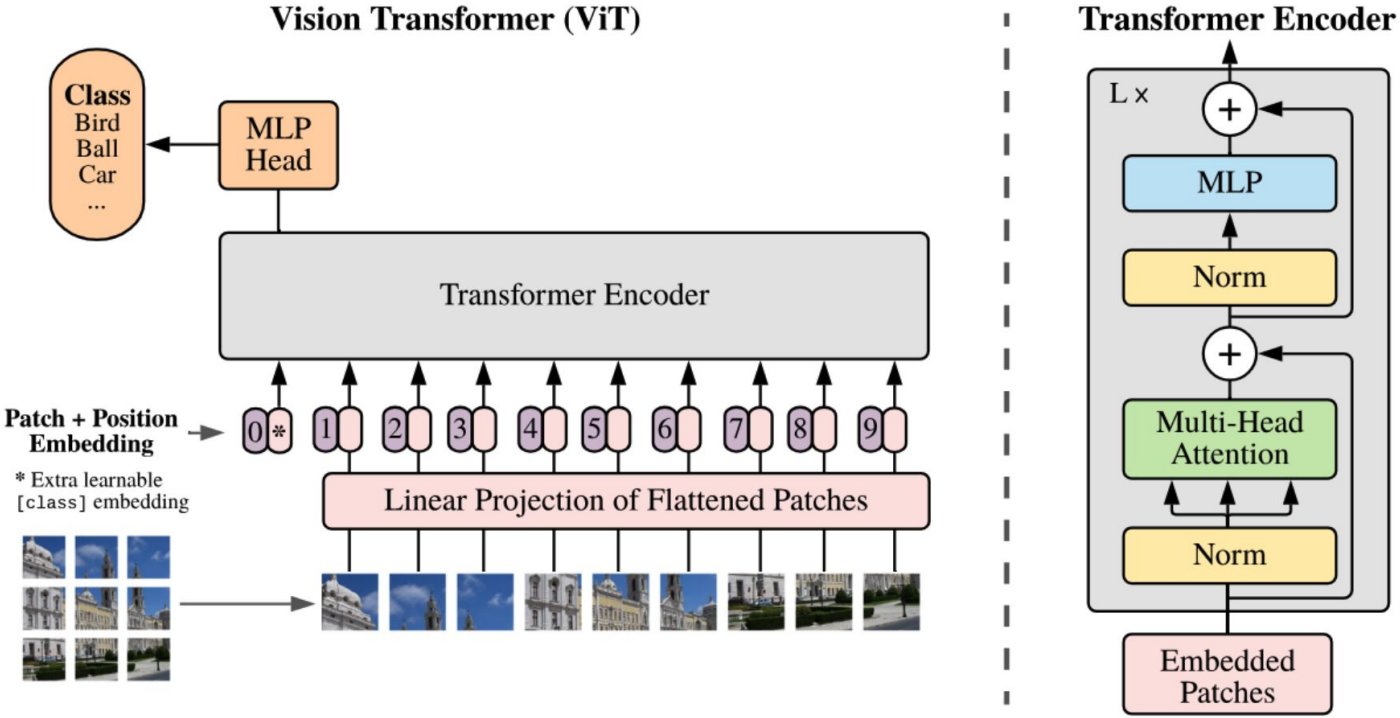
$$w^T x + b$$

Main Component: Convolutional Layers



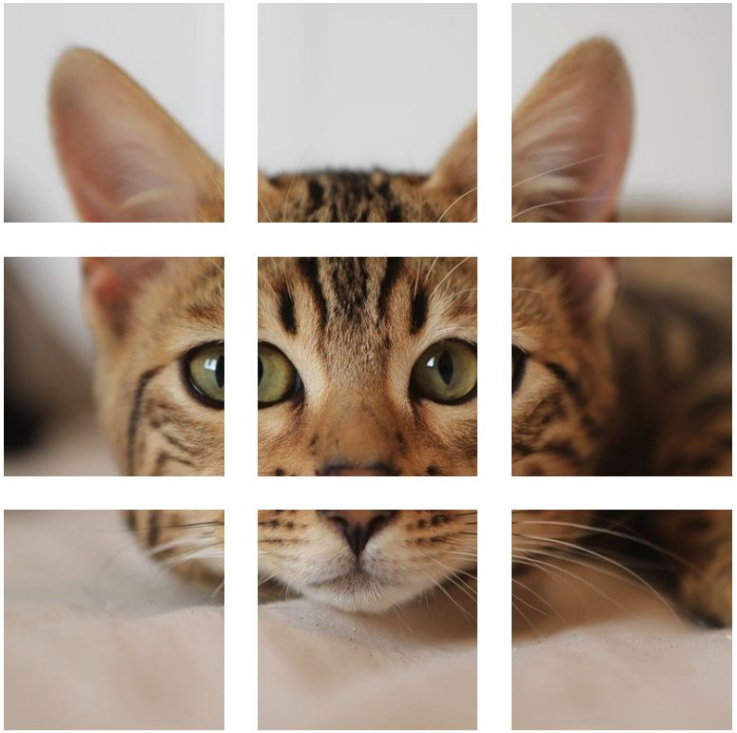
Slide inspiration: Justin Johnson

Alternative: Vision Transformers (ViTs)



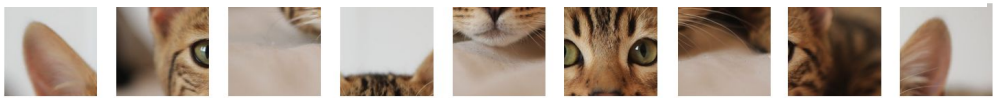
Vision Transformers (ViTs) – Patches

Vision Transformers (ViTs) – Patches



Vision Transformers (ViTs) – Patches

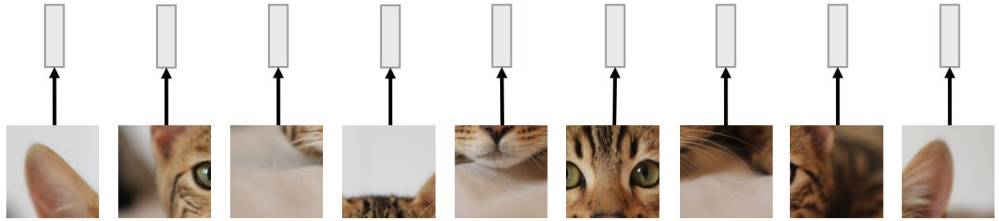
N input patches, each of shape 3x16x16



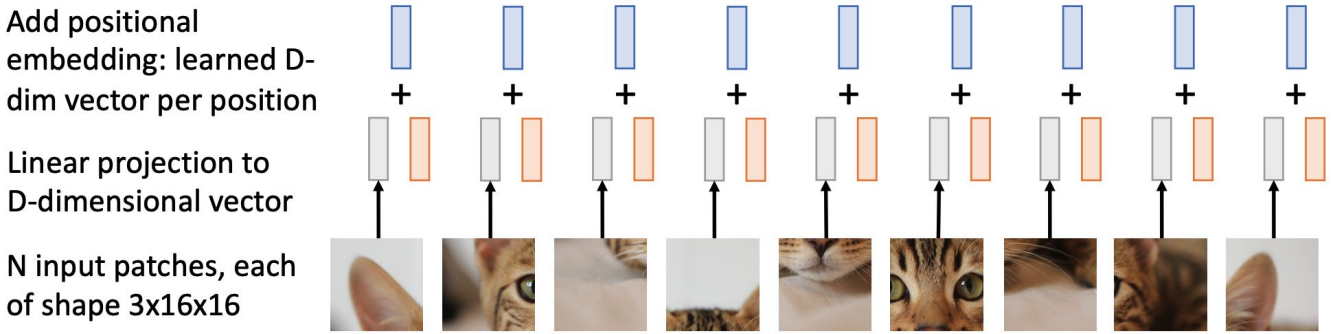
Vision Transformers (ViTs) – Patches

Linear projection to D-dimensional vector

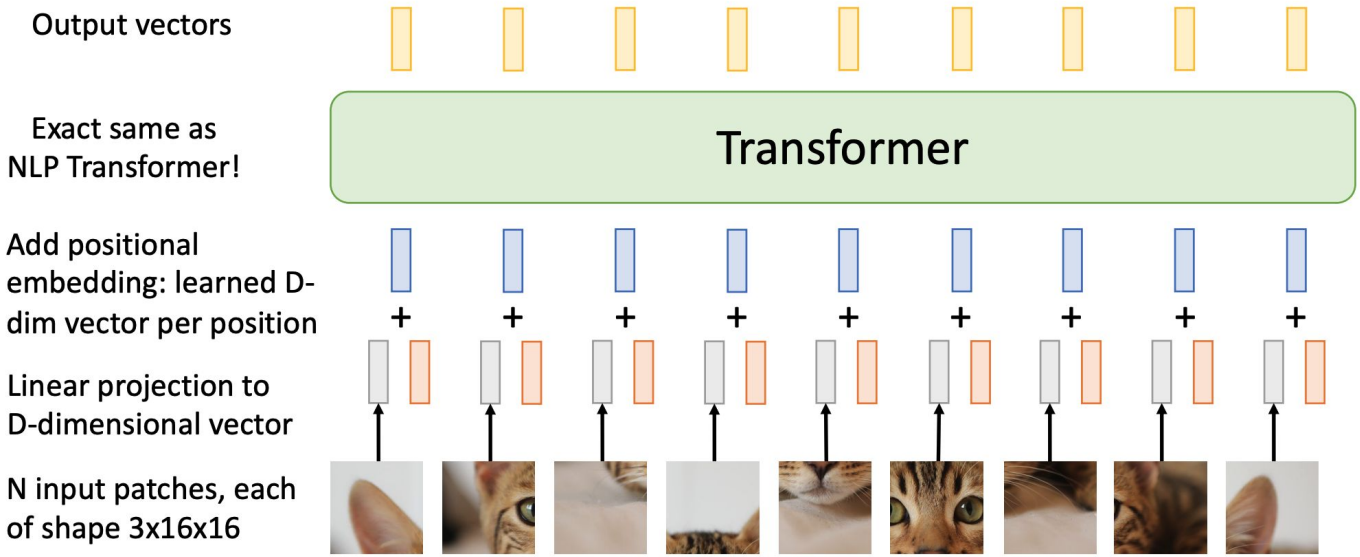
N input patches, each of shape 3x16x16



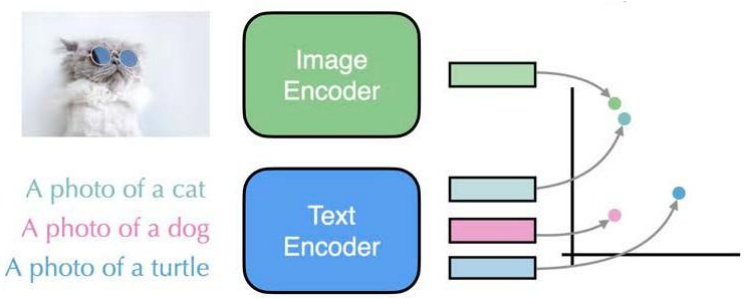
Vision Transformers (ViTs) – Patches



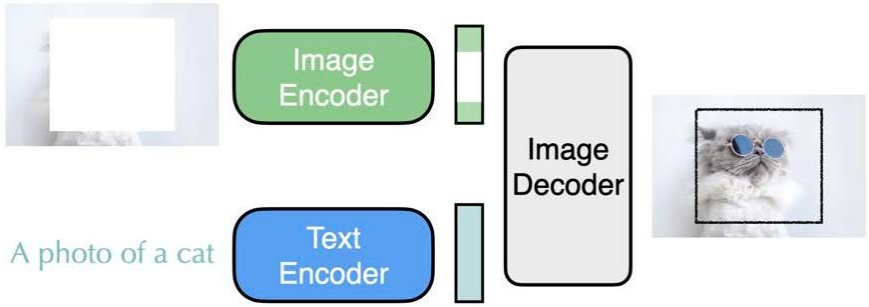
Vision Transformers (ViTs) – Patches



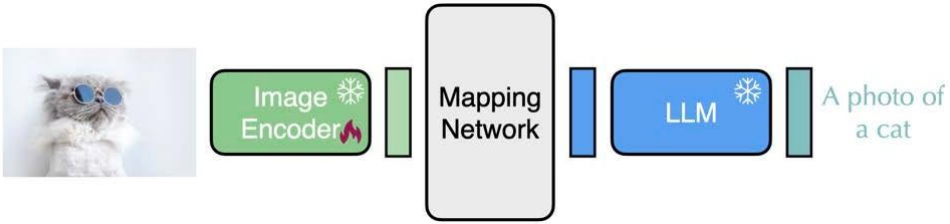
The Families of VLMs



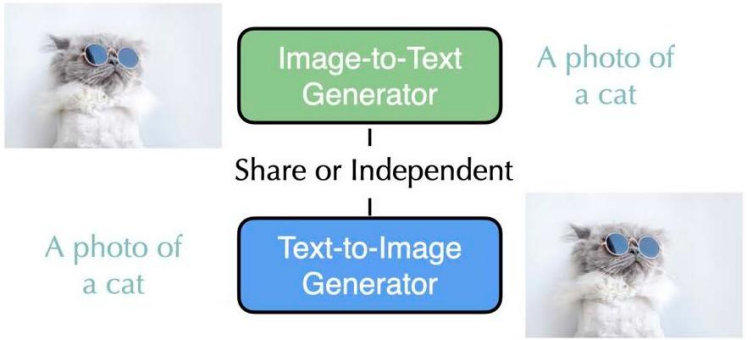
Contrastive-Based



Masking Objective

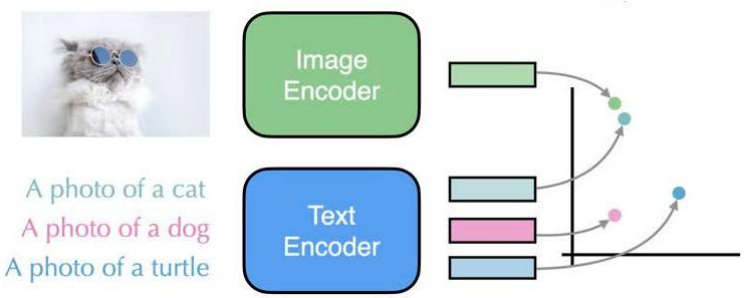


VLMs from Pretrained Backbones



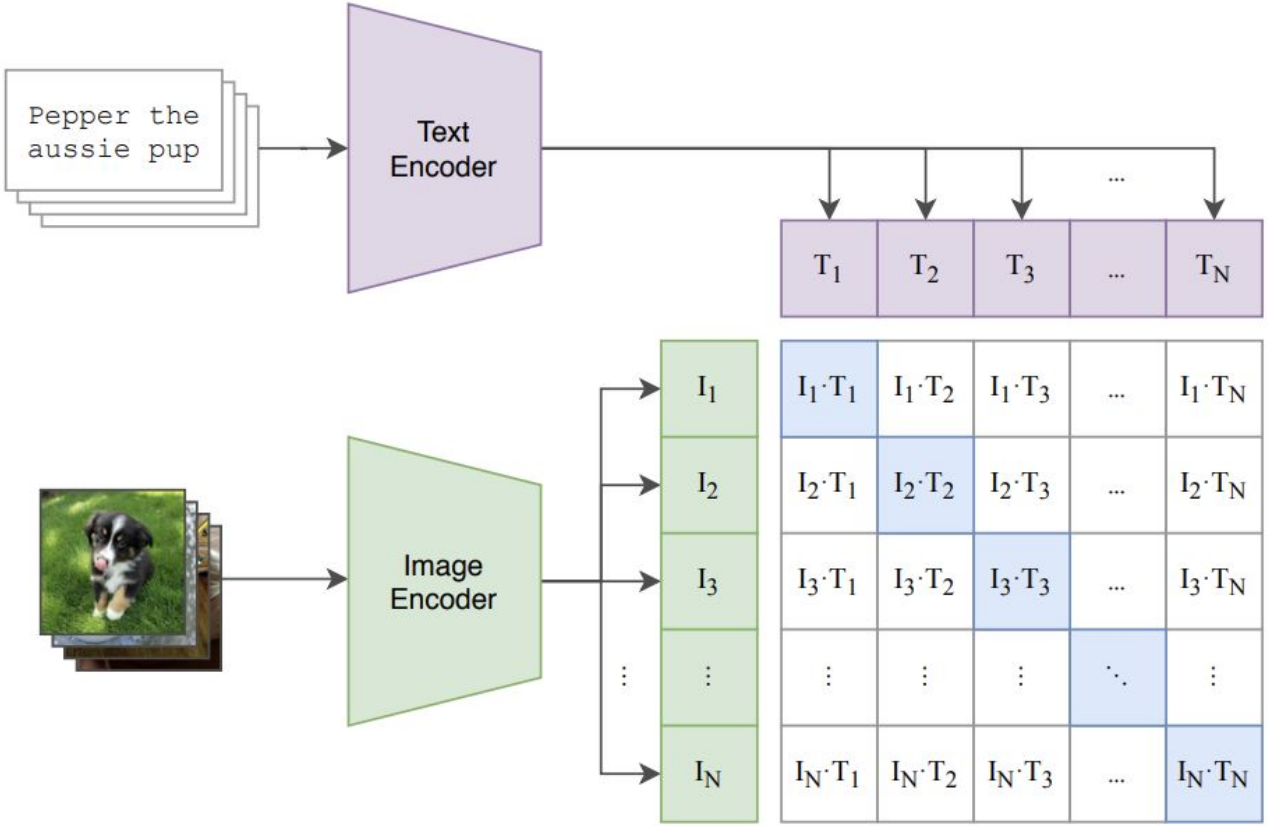
Generative-Based

Contrastive VLMs (CLIP)



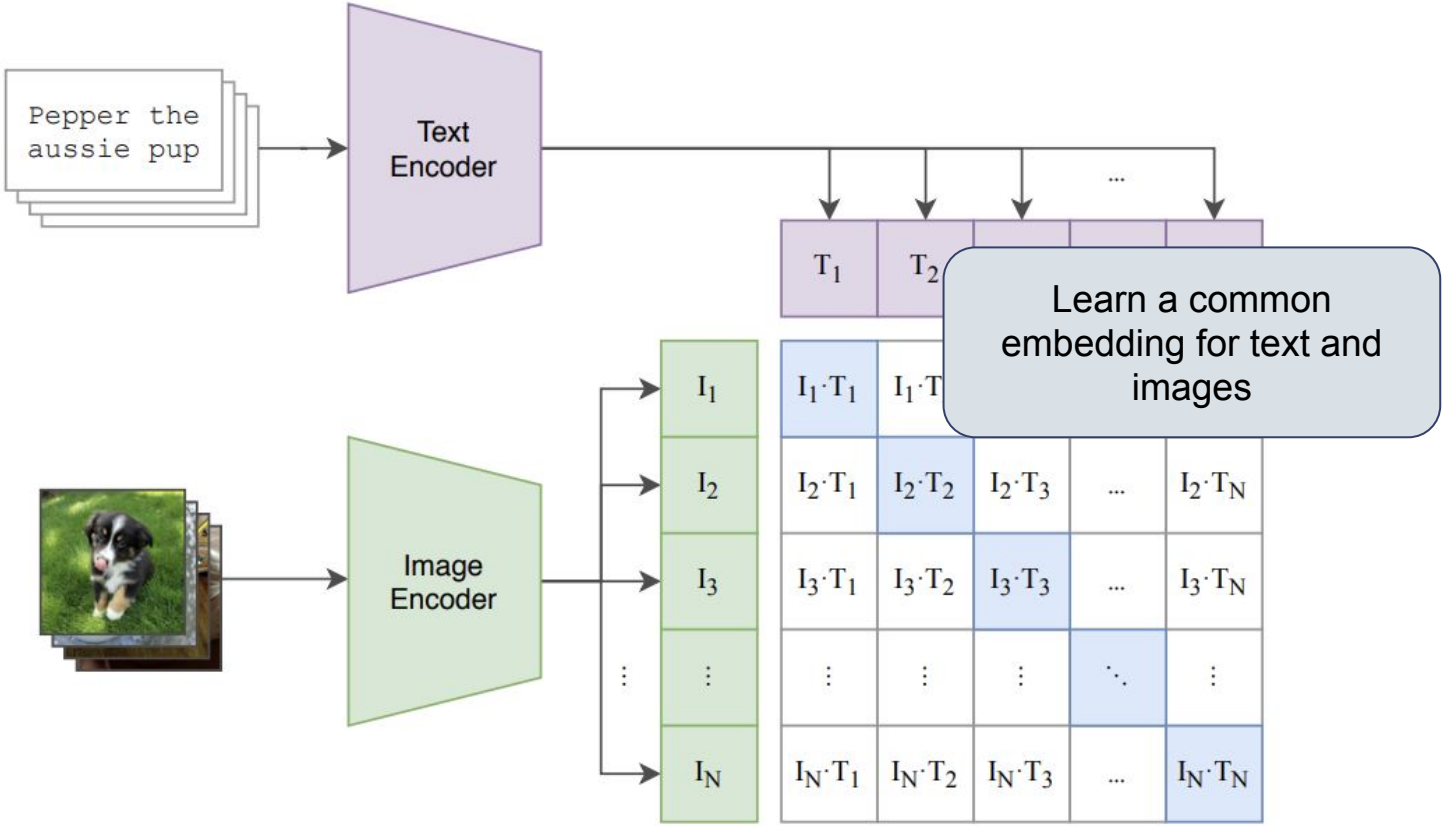
Contrastive-Based

Contrastive VLMs (CLIP)



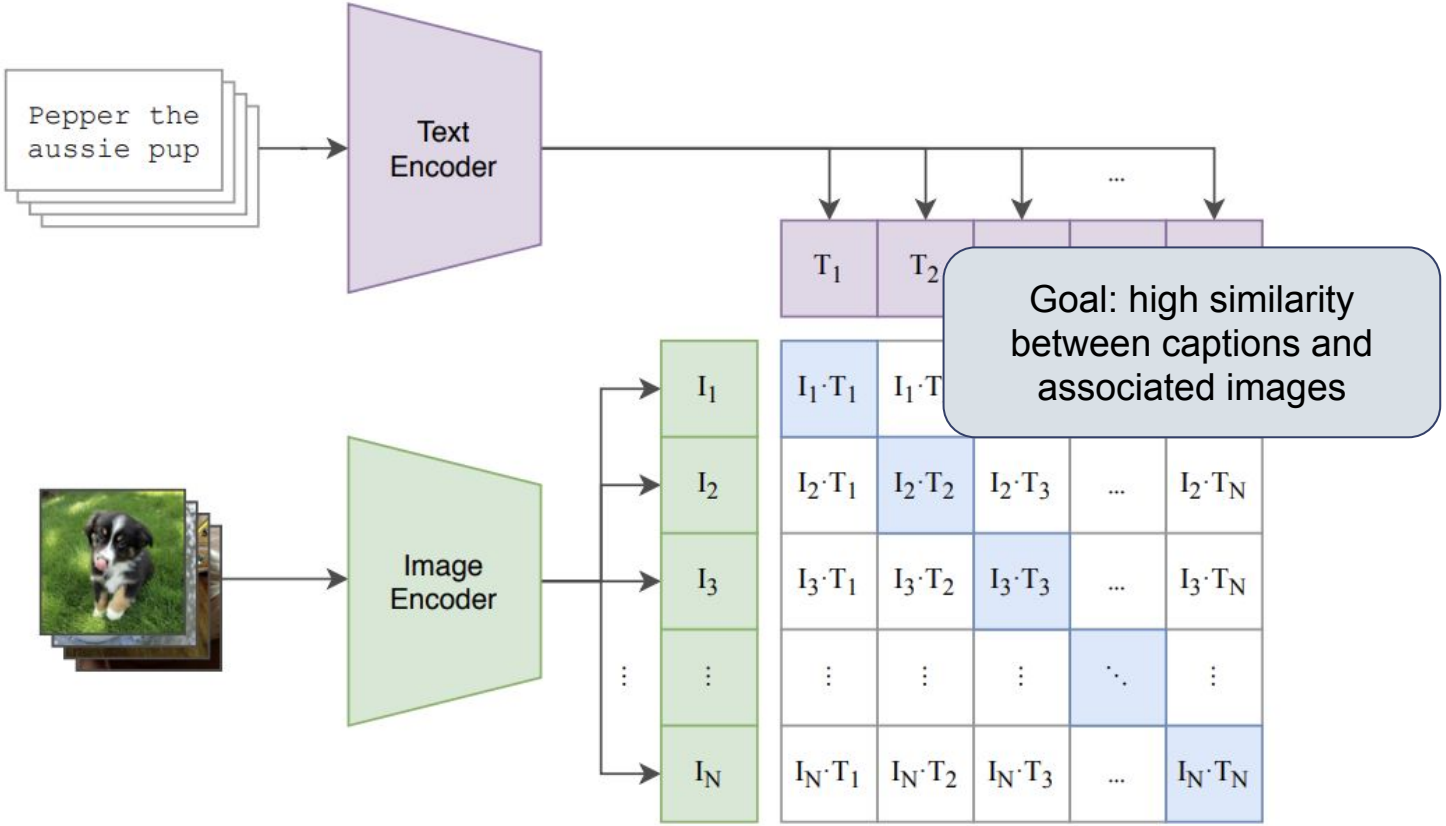
https://arxiv.org/pdf/2103.00020

Contrastive VLMs (CLIP)



https://arxiv.org/pdf/2103.00020

Contrastive VLMs (CLIP)



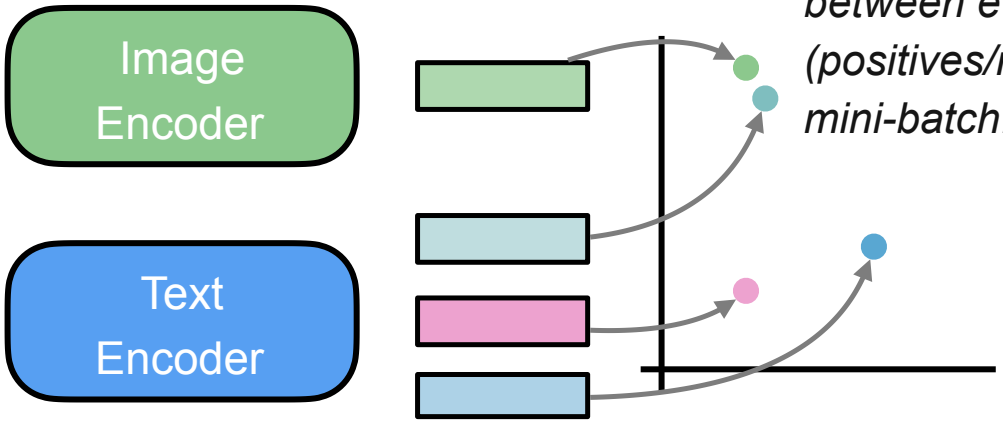
https://arxiv.org/pdf/2103.00020

Contrastive Losses

A photo of a cat

A photo of a dog

A photo of a turtle



Computes the similarity between every pair (positives/negatives) within a mini-batch.

Contrastive Losses

CLIP [ICML 2021]: InfoNCE Loss

$$\mathcal{L} = -\frac{1}{2N}\sum_{i=1}^{N}\left(\overbrace{\log\frac{e^{t\mathbf{x}_i\cdot\mathbf{y}_i}}{\sum_{j=1}^{N}e^{t\mathbf{x}_i\cdot\mathbf{y}_j}}}^{\text{image}\to\text{text softmax}} + \overbrace{\log\frac{e^{t\mathbf{x}_i\cdot\mathbf{y}_i}}{\sum_{j=1}^{N}e^{t\mathbf{x}_j\cdot\mathbf{y}_i}}}^{\text{text}\to\text{image softmax}}\right)$$

Every positive pair is normalized by **all** negative pairs

SigLIP [ICCV 2023]: Sigmoid Loss

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\log\frac{1}{1+e^{z_{ij}(t\mathbf{x}_i\cdot\mathbf{y}_j+b)}} \quad \text{s.t.} \quad z_{ij} = \begin{cases} 1, & \text{for positive pairs.} \\ -1, & \text{for negative pairs.} \end{cases}$$

Every pair (positive/negative) is independent of other pairs

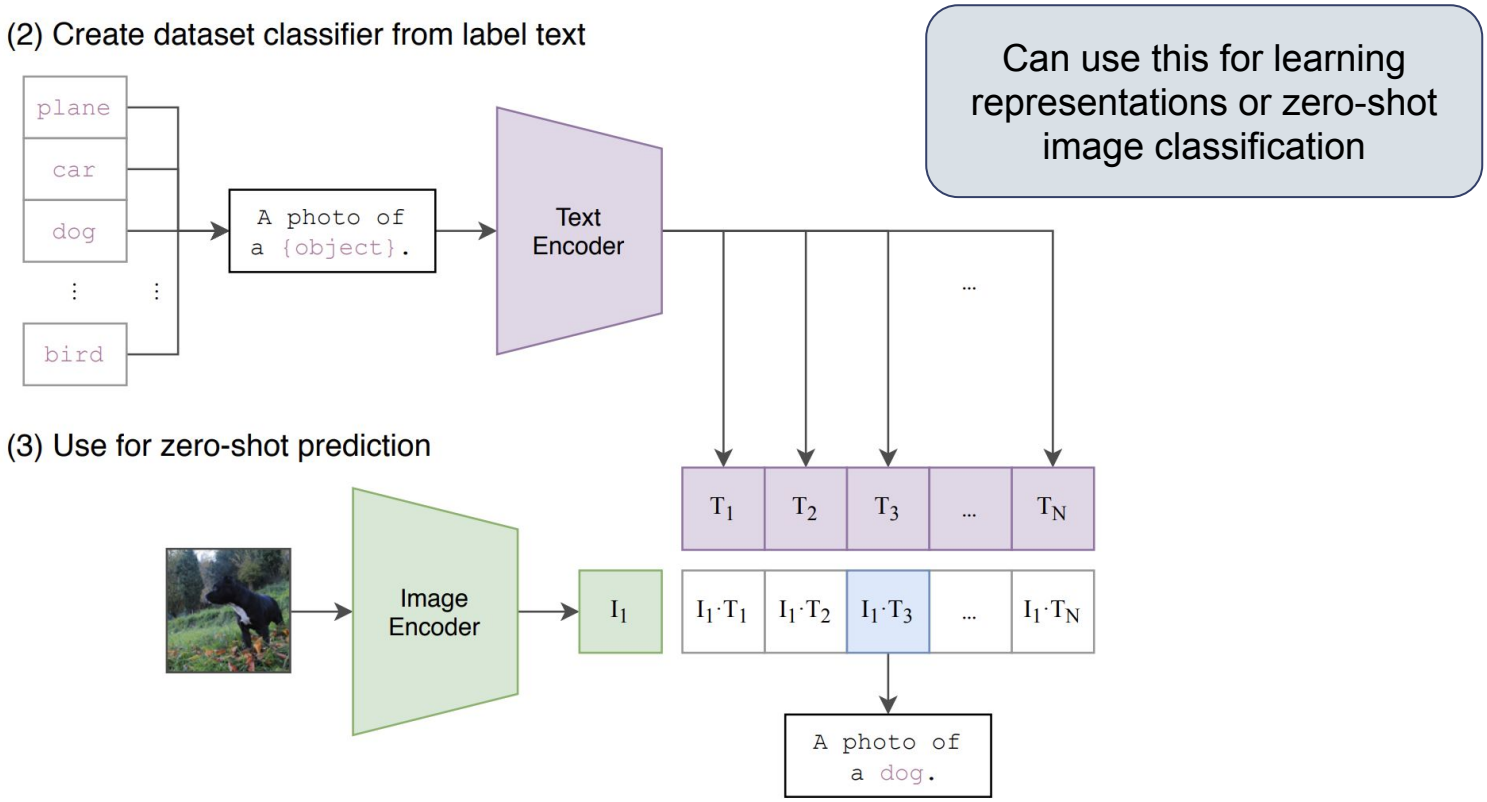
Why Contrastive Models

For own sake (0-shot recognition)

Can use as an encoder into another model (e.g. CLIPPort)

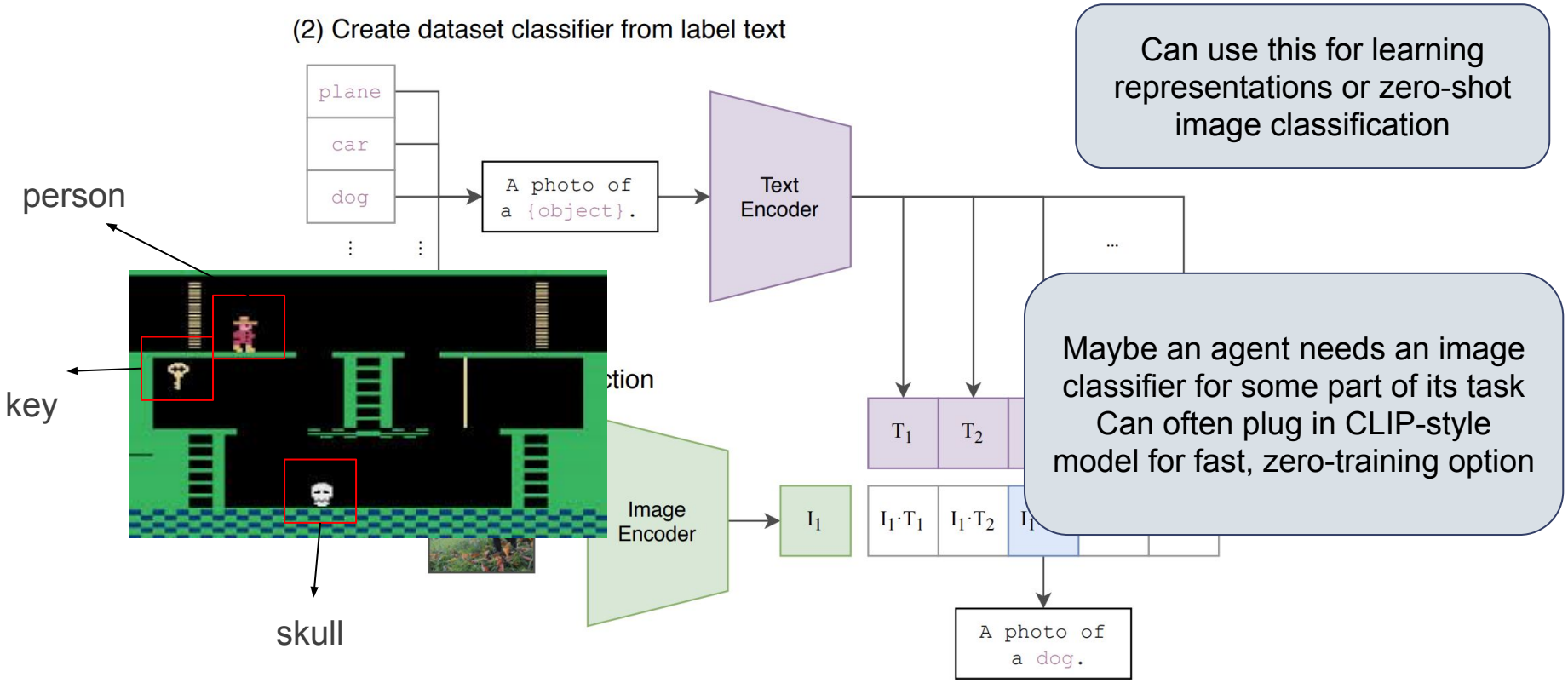
Can fuse with LLM (e.g. Flamingo) - we'll discuss alternatives to this later

Use as zero-shot classifier



Can use this for learning representations or zero-shot image classification

Use as zero-shot classifier

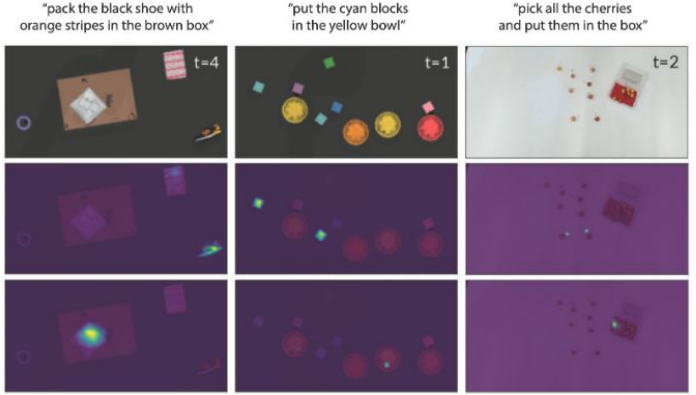


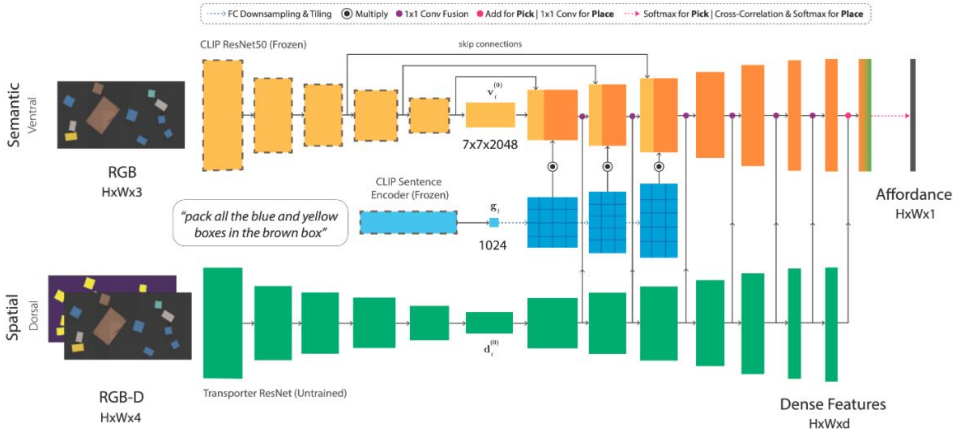
As part of larger models

CLIPort: What and Where Pathways for Robotic Manipulation

CoRL 2021

Mohit Shridhar[1], Lucas Manuelli[2], Dieter Fox[1, 2]
[1]University of Washington, [2]NVIDIA





As part of larger models

Contrastive encoder

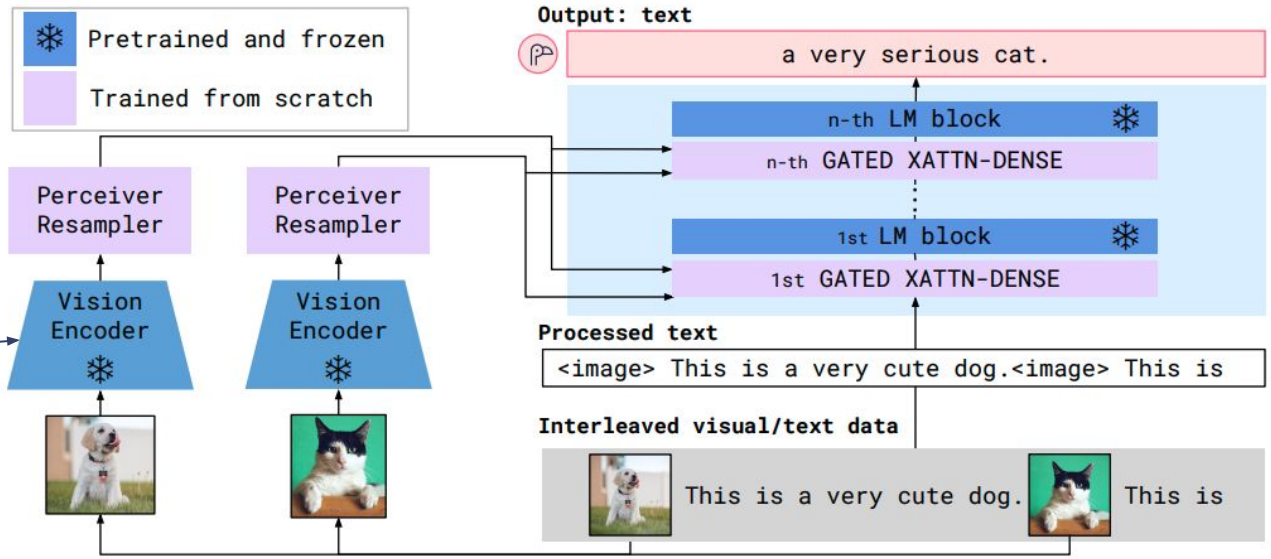
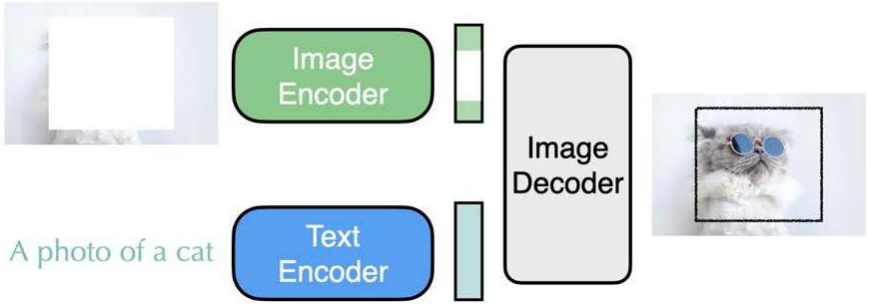


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Masking Objective VLMs



Masking Objective

Masking Objective VLMs

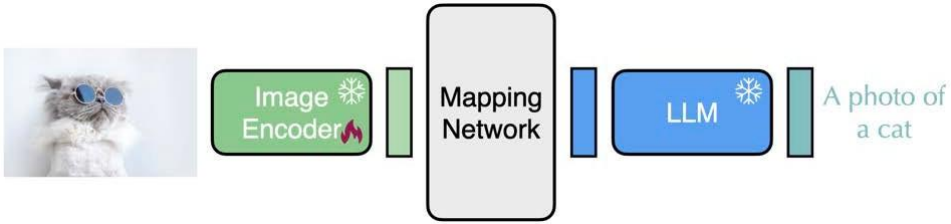
- Mask part of images / text
- BERT for multimodal
- Different versions predict text and/or images

GT: “people are fixing the roof of a house”
Masked: “people are [MASK] [MASK] [MASK] of a [MASK]”
Recon (mask): “people are on **the wing of a tree**”
Recon (org): “people are **working the roof of a house**”

GT: “a young woman is giving a baby a ride on her shoulders”
Masked: “a young [MASK] is giving [MASK] [MASK] [MASK] ride on her [MASK]”
Recon (mask): “a young girl is giving **theons** a ride on her **horse**”
Recon (org): “a young mother is giving **her baby** a ride on her **shoulders**”

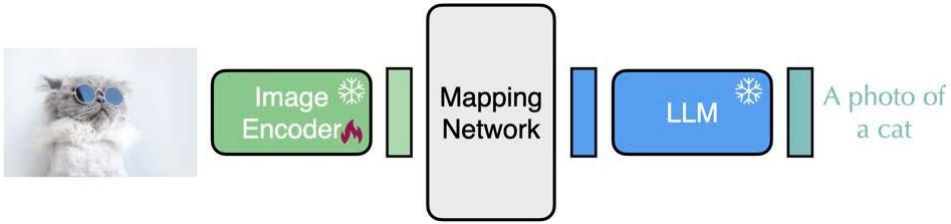
GT: “a girl in a jean dress is walking along a raise balance beam”
Masked: “a girl in a jean [MASK] is [MASK] along a raise [MASK] [MASK]”
Recon (mask): “a girl in a jean **house** is **mirrored** along a raise **pink boat**”
Recon (org): “a girl in a jean **dress** is **walking** along a raise **wooden beam**”

VLMs from LLM Backbones



VLMs from Pretrained Backbones

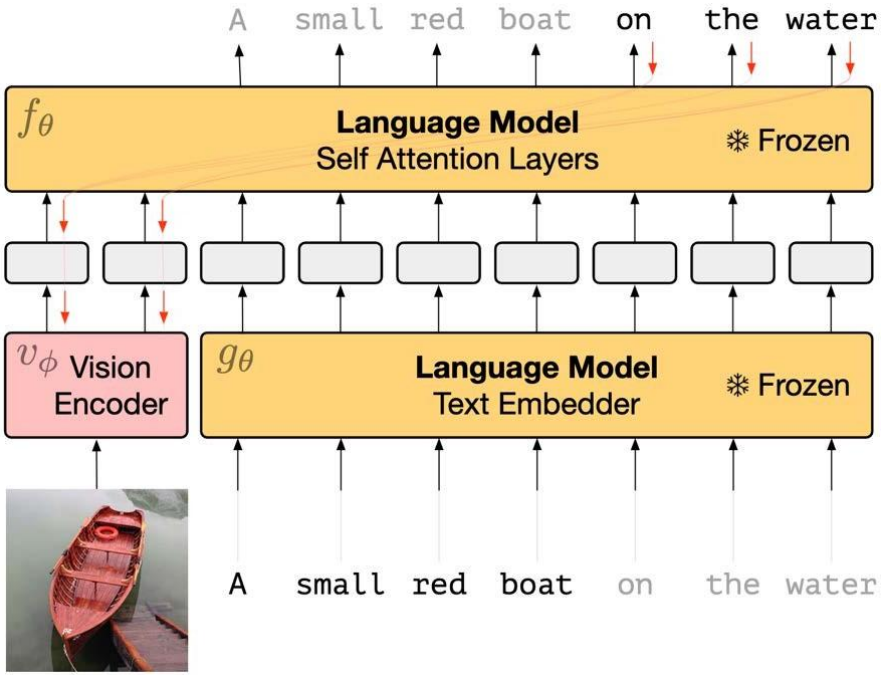
VLMs from LLM Backbones



VLMs from Pretrained Backbones

Key Questions:
What Image / LLM backbones
How to combine
Where to freeze

Early Example



Freeze LLM
Train image encoder from scratch

Tsimpoukelli, Maria, et al. "Multimodal few-shot learning with frozen language models." *NeurIPS 2021*.

Example: Flamingo

Contrastive encoder

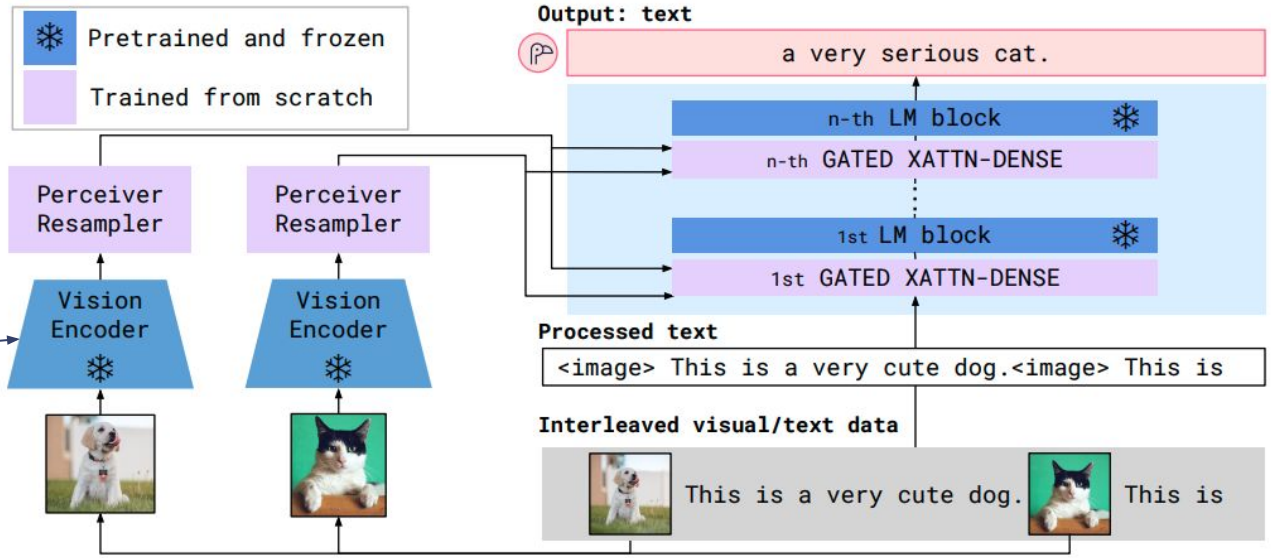


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Example: Flamingo

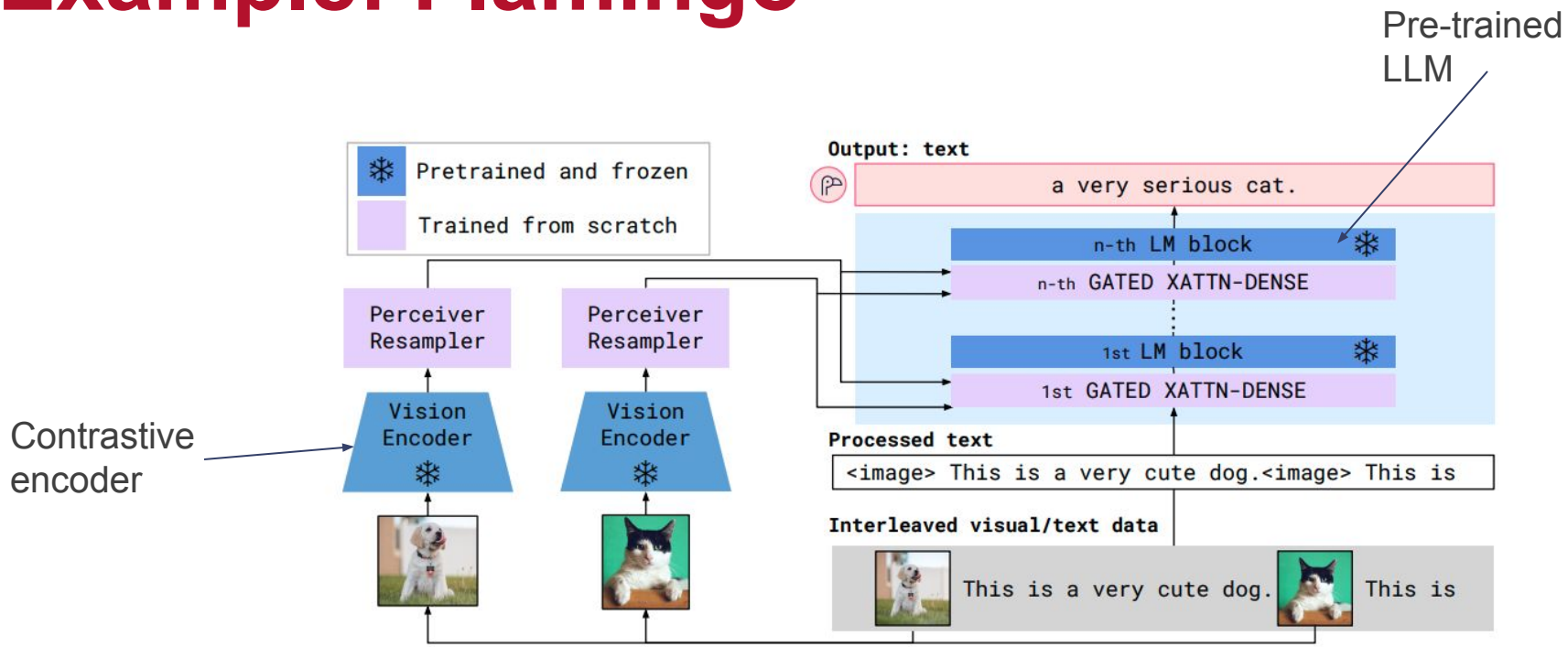


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Example: Flamingo

Frozen VM and LM

Perceiver Resampler

- Outputs fixed number of visual tokens given variable number of visual features

Gated cross-attention block

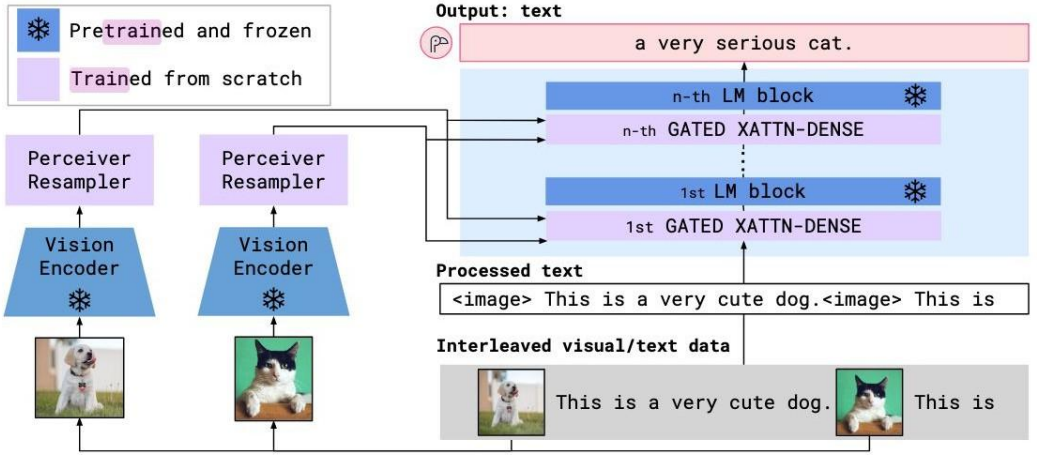
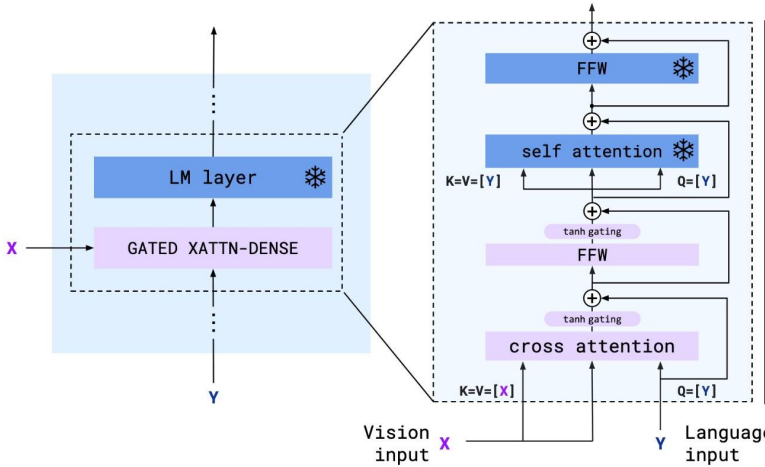


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Example: Flamingo

- Inserted every 4th LM block (efficiency/accuracy tradeoff)
- Tanh gating to maintain LM performance at initialization



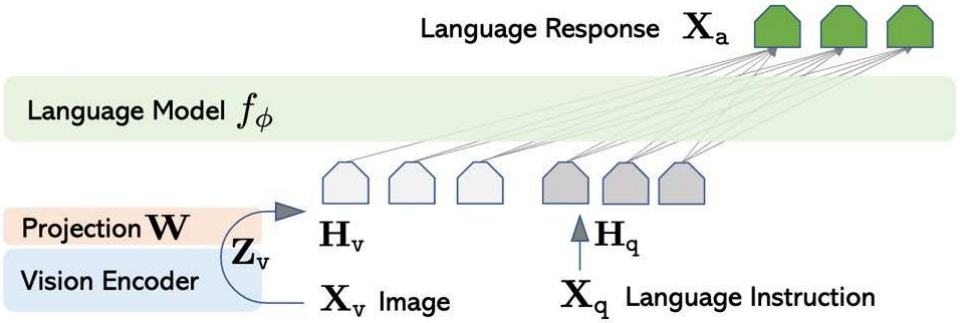

```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter – init at 0.
    alpha_dense, # ffw gating parameter – init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 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
```

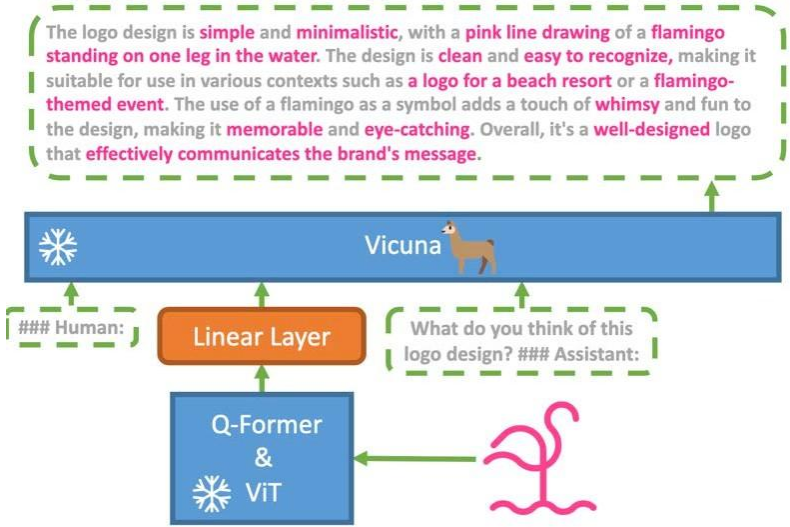
Figure 4: **GATED XATTN-DENSE layers.** To condition the LM on visual inputs, we insert new cross-attention layers between existing pretrained and frozen LM layers. The keys and values in these layers are obtained from the vision features while the queries are derived from the language inputs. They are followed by dense feed-forward layers. These layers are *gated* so that the LM is kept intact at initialization for improved stability and performance.

Slide credit: Ryan Tabrizi

LLaVA / Mini GPT-4

MiniGPT-4/LLaVA require only training the project layer: the visual encoder and LLM are already pretrained and used as off-the-shelf from prior work, such as CLIP and Vicuna



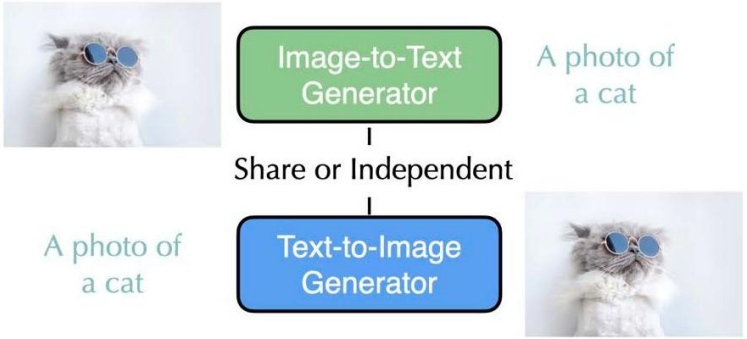


1 Liu, Haotian, et al. "Visual instruction tuning." *NeurIPS 2024*.

2 Zhu, Deyao, et al. "MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models." *ICLR 2023*

Slide credit: XuDong Wang

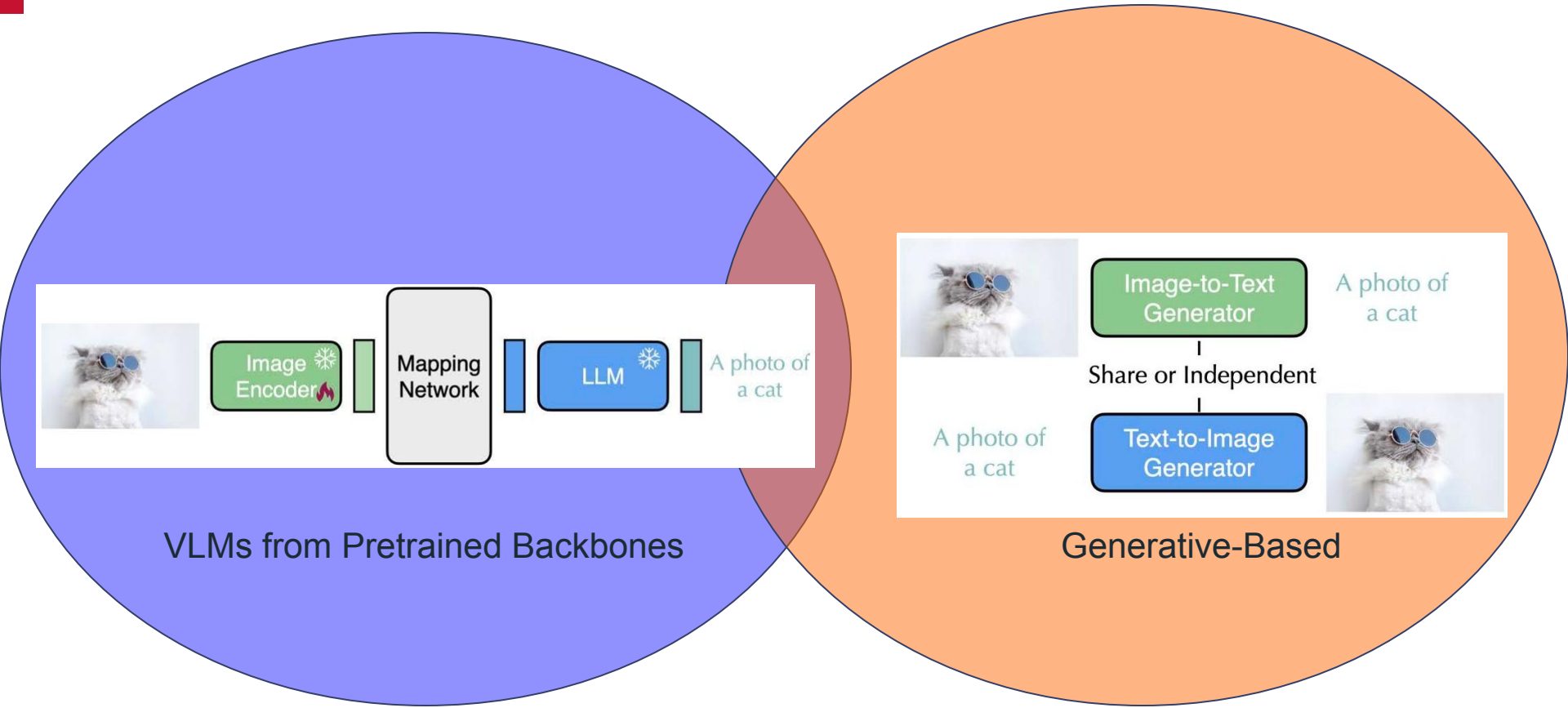
Generative VLMs



Generative-Based

Generative VLMs

Obviously some overlap here



VLMs from Pretrained Backbones

Generative-Based

Generative VLMs

VLMs are trained in such a way they can generate entire images or very long captions (or both)

Image-to-Text Generator

A photo of a cat

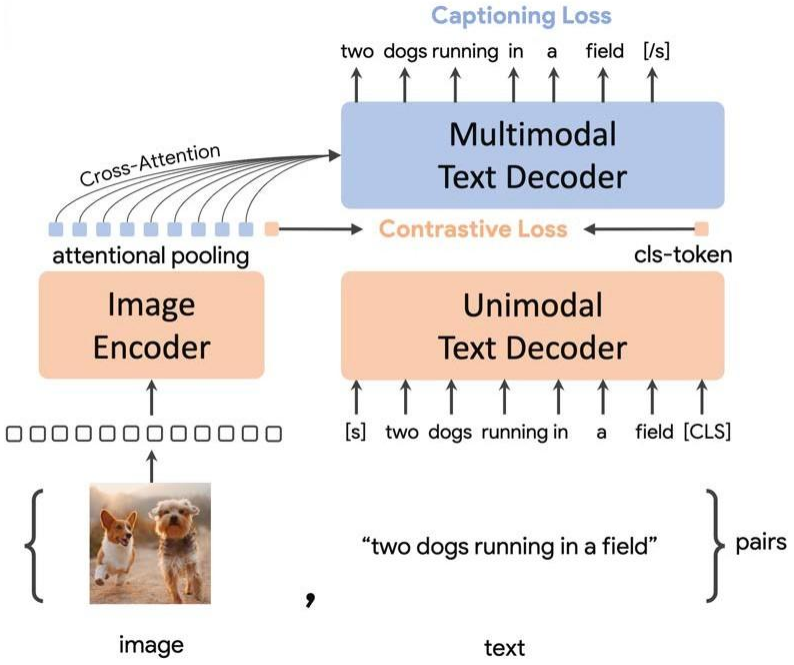
Shared or Independent

A photo of a cat

Text-to-Image Generator

Generative VLMs - CoCa

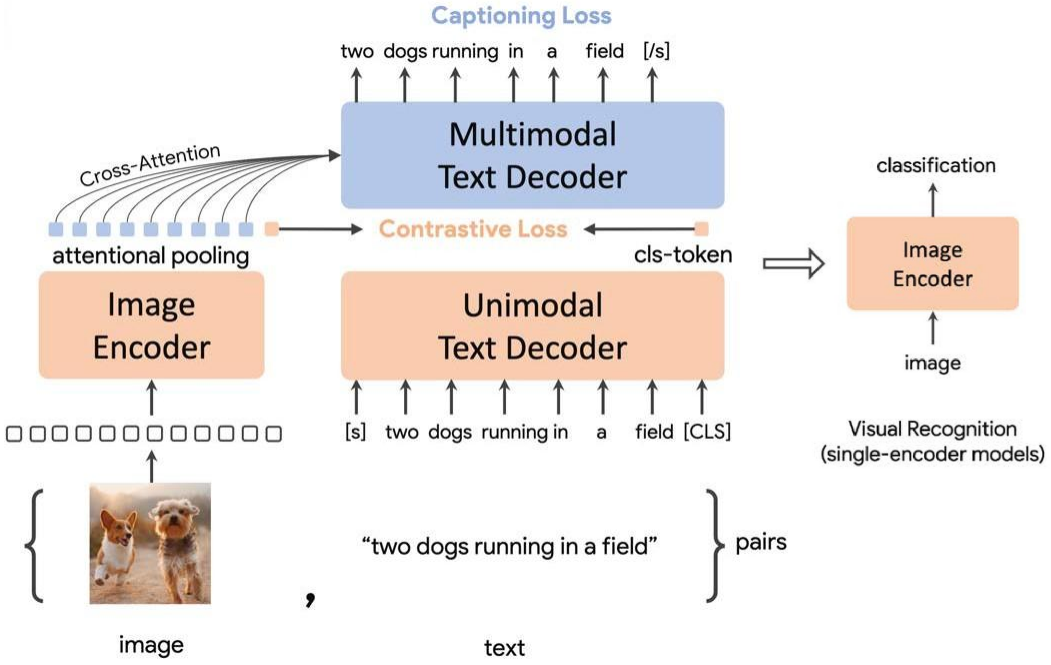
Pretrain an image-text encoder-decoder model with contrastive and captioning loss.



Yu, Jiahui, et al. "CoCa: Contrastive Captioners are Image-Text Foundation Models." *TMLR 2022*

Generative VLMs - CoCa

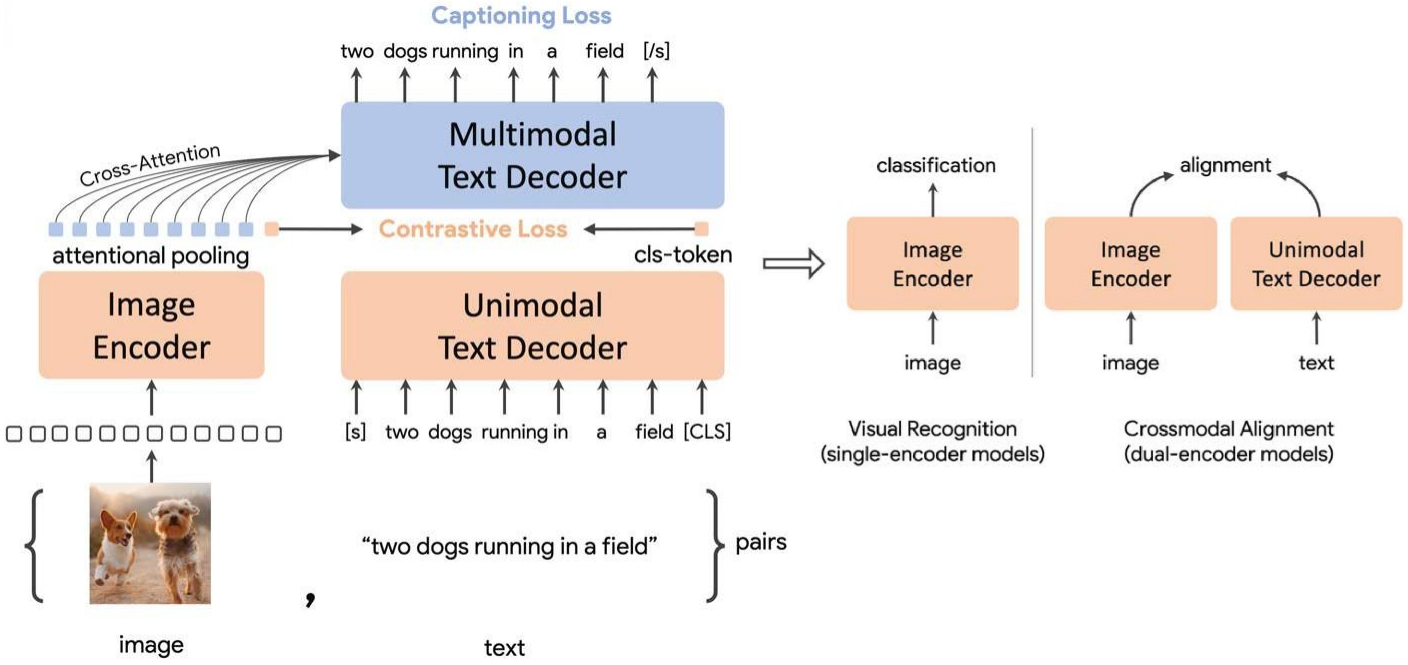
The pretrained CoCa can be used for visual recognition, vision-language alignment, image captioning and multimodal understanding with zero-shot transfer, frozen-feature evaluation or end-to-end finetuning.



Yu, Jiahui, et al. "CoCa: Contrastive Captioners are Image-Text Foundation Models." *TMLR 2022*

Generative VLMs - CoCa

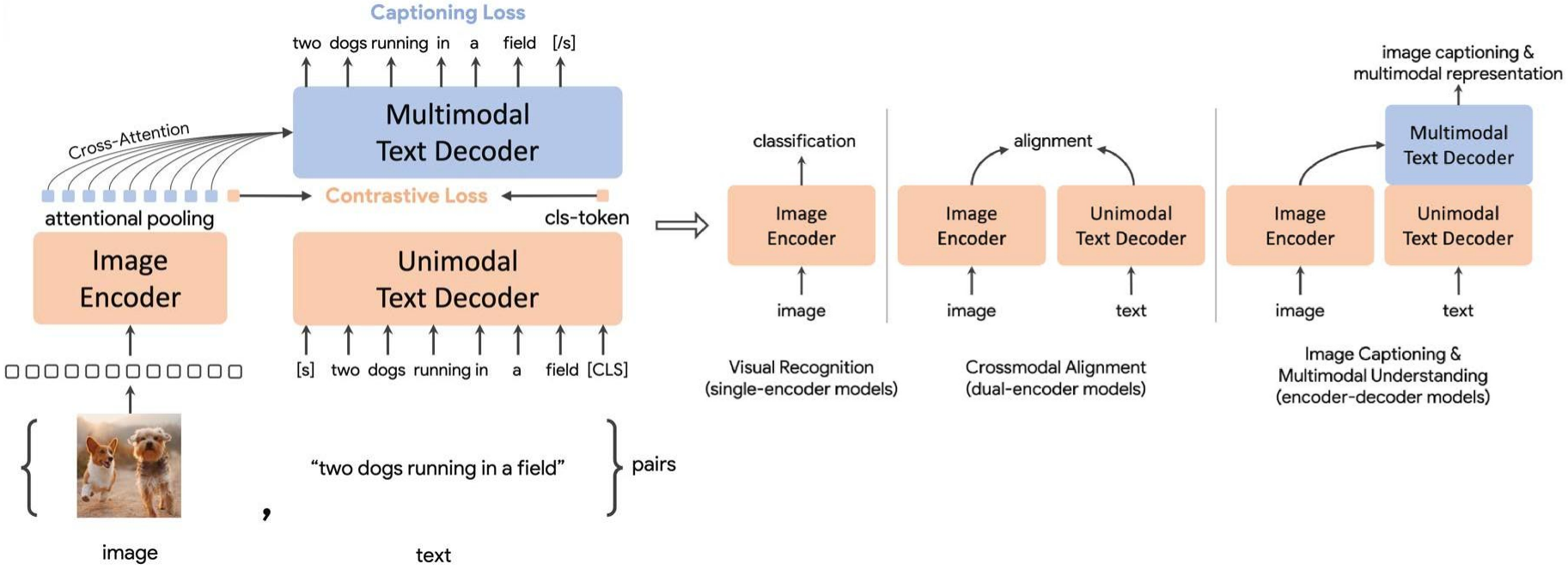
The pretrained CoCa can be used for visual recognition, vision-language alignment, image captioning and multimodal understanding with zero-shot transfer, frozen-feature evaluation or end-to-end finetuning.



Yu, Jiahui, et al. "CoCa: Contrastive Captioners are Image-Text Foundation Models." *TMLR 2022*

Generative VLMs - CoCa

The pretrained CoCa can be used for visual recognition, vision-language alignment, image captioning and multimodal understanding with zero-shot transfer, frozen-feature evaluation or end-to-end finetuning.

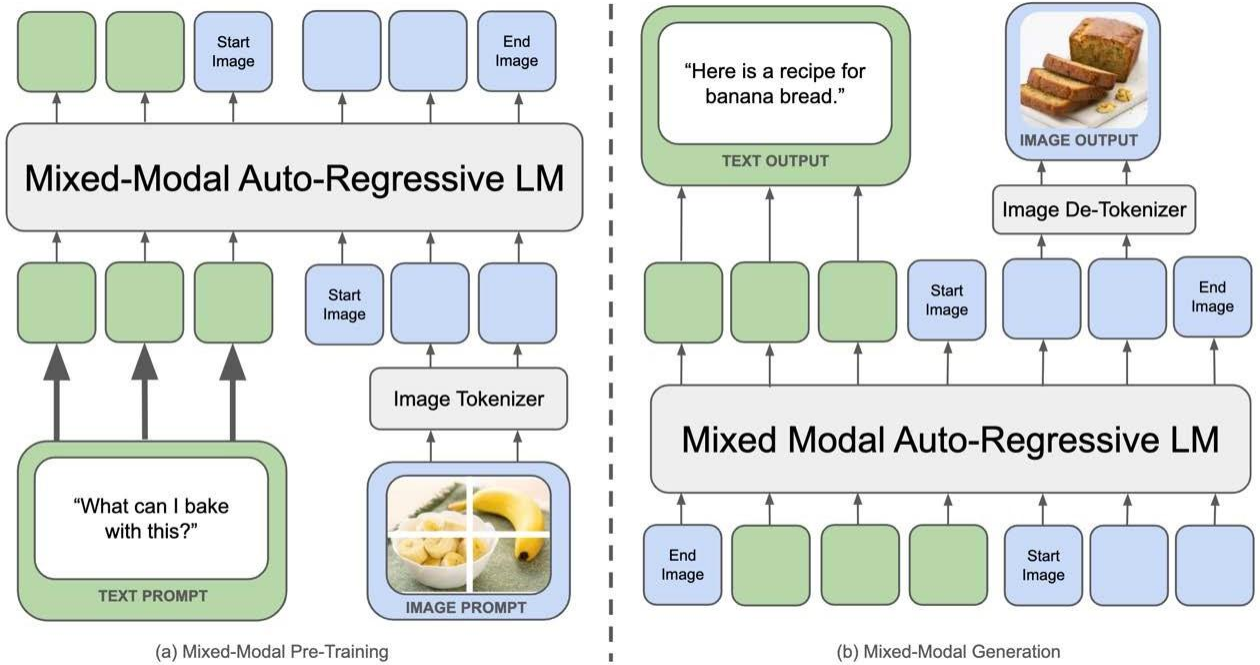


Yu, Jiahui, et al. "CoCa: Contrastive Captioners are Image-Text Foundation Models." *TMLR 2022*

Slide credit: XuDong Wang

Generative VLMs - Chameleon

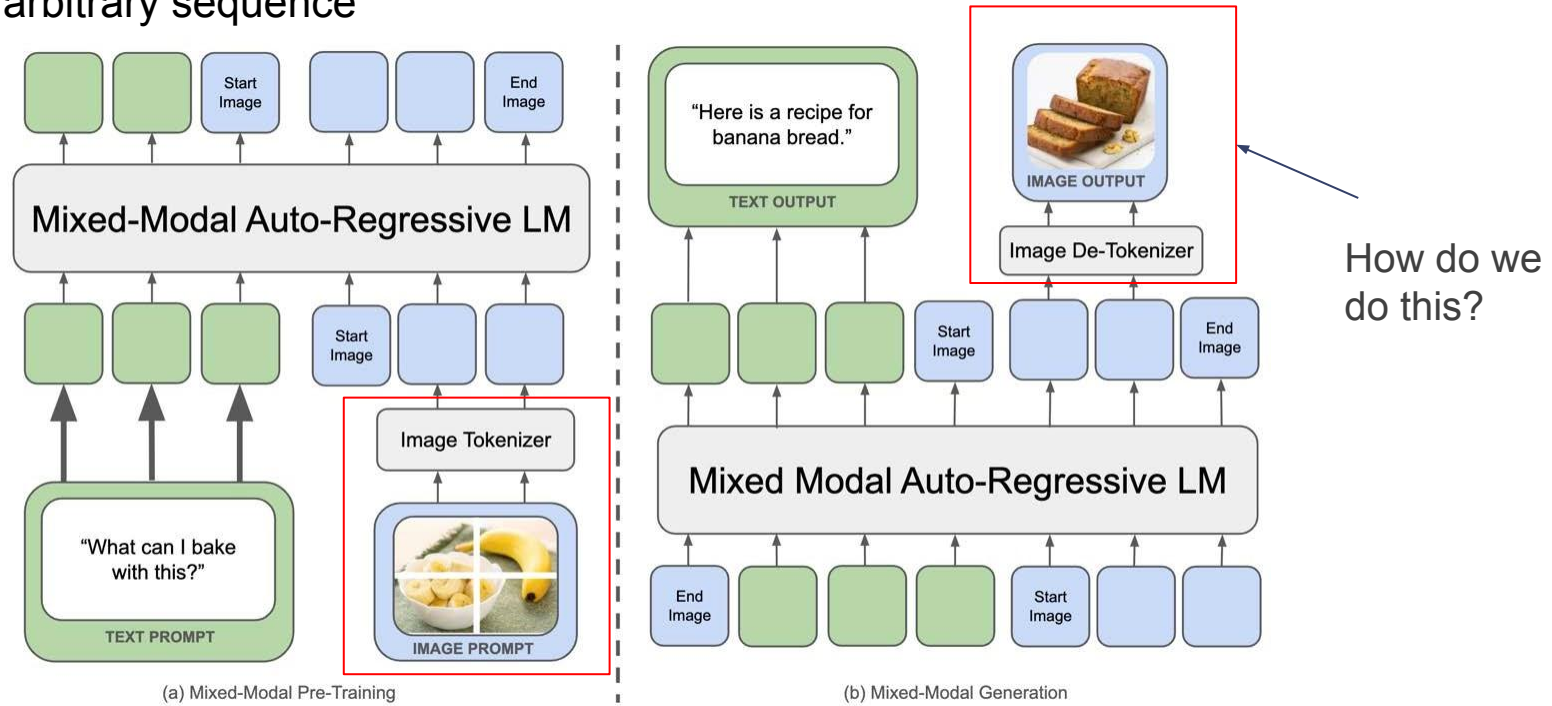
Early-fusion token-based mixed-modal models capable of understanding and generating images and text in any arbitrary sequence



Chameleon Team. "Chameleon: Mixed-modal early-fusion foundation models." *arXiv preprint arXiv:2405.09818* (2024).

Generative VLMs - Chameleon

Early-fusion token-based mixed-modal models capable of understanding and generating images and text in any arbitrary sequence

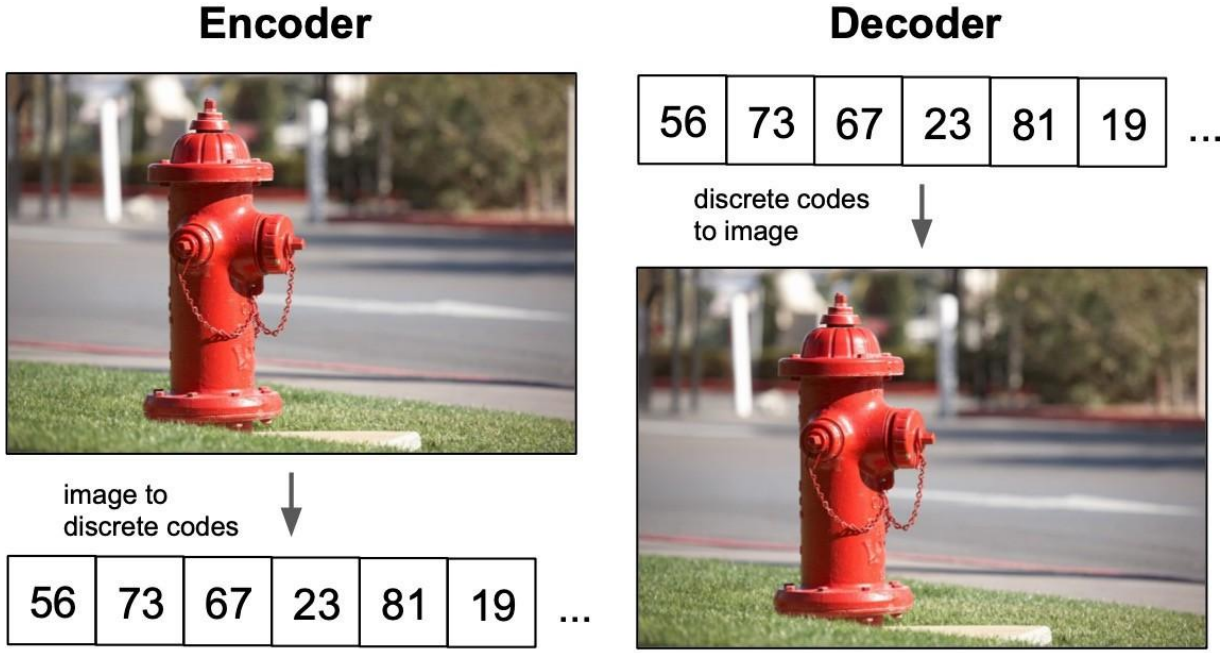


Chameleon Team. "Chameleon: Mixed-modal early-fusion foundation models." *arXiv preprint arXiv:2405.09818* (2024).

VQ-VAE based VLM encoder

- VQ-VAE encodes a image into a sequence of tokens (integers)
  - VQ-VAE-Encoder(image) = a list of integers, each integer is in [image_vocab_size] (something like 8096).
  - AE: Auto-Encoder
- Roughly speaking, we want to train two models:
  - VQ-VAE-Encoder
  - VQ-VAE-Decoder
- For every image:
  - VQ-VAE-Decoder(VQ-VAE-Encoder(image)) = image
  - We can recover the original image from the list of integers output by VQ-VAE-Encoder

VQ-VAE based VLM encoder



Training a VQ-VAE

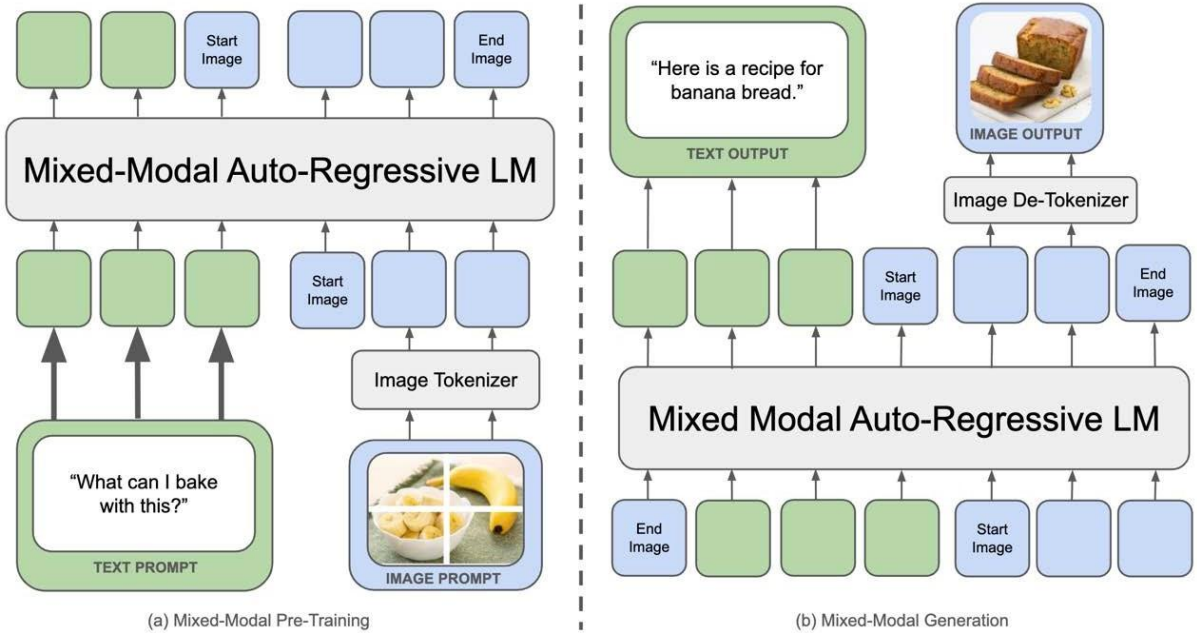
- Quantizing the output of standard VAE.
- VAE takes an input image, and output (a sequence of) vectors.
  - VAE(image) = vector1, vector2, …, vectork. Those vectors can take any value.
- We want to map each vector to a vector from a finite set (e1, e2, …, en).
  - Training objective:
  - Maintain a set of vectors e1, e2, …, en.
  - For each vector vectori, map it to the argmin of $\left|\left|e_j - vector_i\right|\right|^2$ (for all j in [n]), let's call the argmin R(i)
  - Add to the loss function: sum of $\left|\left|e_{R(i)} - vector_i\right|\right|^2$ for all i.
  - Argmin is not differentiable, but we just treat the gradient as 0.

See https://rohitbandaru.github.io/blog/VAEs/ for more detailed explanations

Slide credit: Matt Gormley

Using VQ-VAE in VLM

- Training objective: Next token prediction (loss on both image and text tokens):



- Input to the transformer
  - [text_token1, text_token2, …., text_tokenk, image_token1, image_token2, …, image_tokenm, text_token{k+1}, ….]

Slide credit: Matt Gormley

VQ-VAE vs CLIP encoding

- VQ-VAE
  - Can be used to generate images (loss on image tokens), support arbitrary resolution/aspect-ratio (different images will be mapped to different length of image tokens).
- CLIP
  - Vectors encode more information than discrete tokens, so CLIP preserves more details of the original image.

Generative VLMs - Chameleon

Prompt: <img> While chameleons are masters of camouflage in the natural world, let's see how they might fare in a man-made setting.
1. Analyze the colors, patterns, and textures in this image and identify the most challenging spot for a chameleon to blend in.
2. Then generate an image of a chameleon with the optimal pattern to camouflage itself.
3. Then, rate and explain how successful each attempt is on a scale from 1-10.

Response: The image is of a stop sign in a forested area, with some sunlight filtering through the trees. The sign has a red octagonal shape with a white border and the word "STOP" written in white letters. In the background, there are green leaves and trees.

Challenging Spot for a Chameleon: The stop sign would be the most challenging spot for a chameleon to blend in due to its bright red color and bold white letters, which would stand out in the forested environment.

Optimal Pattern for Camouflage: To camouflage itself on the stop sign, a chameleon would need to have a pattern that would break up its outline and make it less conspicuous. A pattern with shades of brown and red would help the chameleon blend in more effectively. Additionally, a mottled or dappled pattern would help to scatter light and further reduce the chameleon's visibility.

Rating and Explanation: A rating of 2/10 would be given for the chameleon's ability to blend in on the stop sign. While the chameleon's green and brown pattern helps to break up its outline, the bright red color of the stop sign will still make it stand out in the forested environment.

Prompt: Text + Image

Output: Text + Image

Understanding

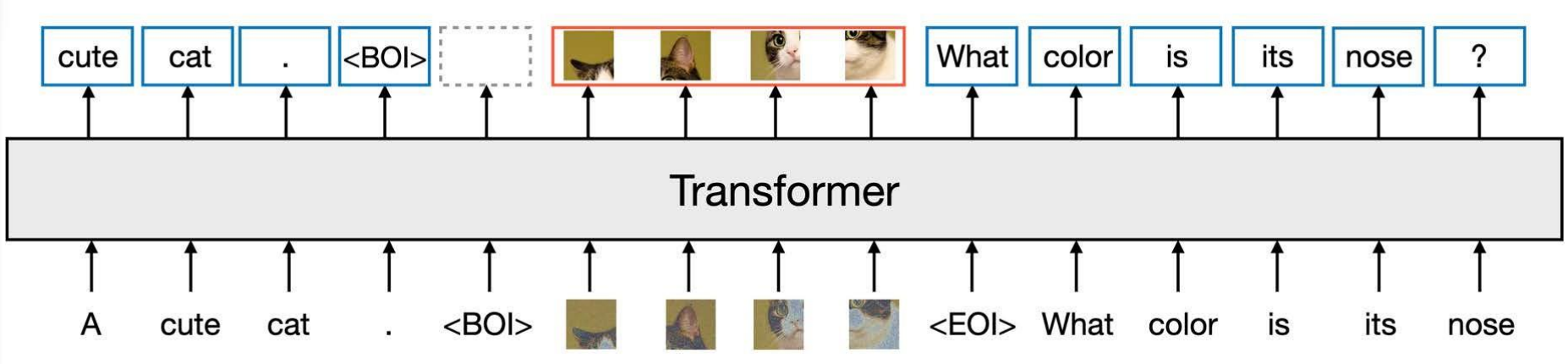
Reasoning

Image Generation

Slide credit: XuDong Wang

Generative VLMs - Transfusion

vs. Chameleon: uses continuous image vectors and trains on the diffusion objective. The image generation results can be significantly improved.



Zhou, Chunting, et al. "Transfusion: Predict the Next Token and Diffuse Images with One Multi-Modal Model." arXiv preprint arXiv:2408.11039 (2024).

Generative VLMs - Transfusion

a monarch butterfly.

A rowboat on a lake with a bike on it.

An expressive oil painting of a chocolate chip cookie being dipped in a glass of milk, depicted as an explosion of flavors.

An angry duck doing heavy weightlifting at the gym.

Downtown Seattle at sunrise. detailed ink wash.

A car made out of vegetables.

An emoji of a baby panda wearing a red hat, green gloves, red shirt, and green pants.

A tranquil, anime-style koi pond in a serene Japanese garden, featuring blossoming cherry trees.

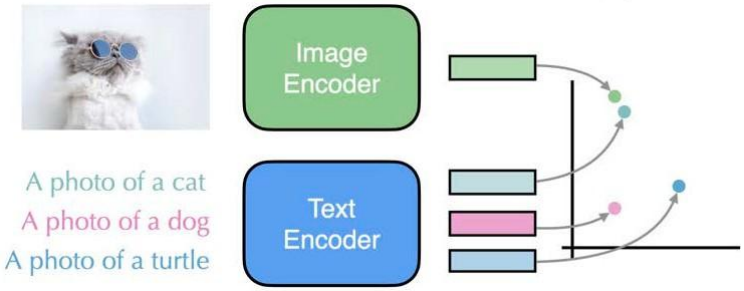
a massive alien space ship that is shaped like a pretzel.

graffiti of a funny dog on a street wall.

A sign that says “Diffusion".

A black basketball shoe with a lightning bolt on it.

When to Use Contrastive Models?



Contrastive-Based

Pros:

1. associate text with visual concepts while keeping a simple training paradigm
2. a good base for building more complex model
3. retrieve the images (captions) via prompting the CLIP text (image) encoder with words (images)

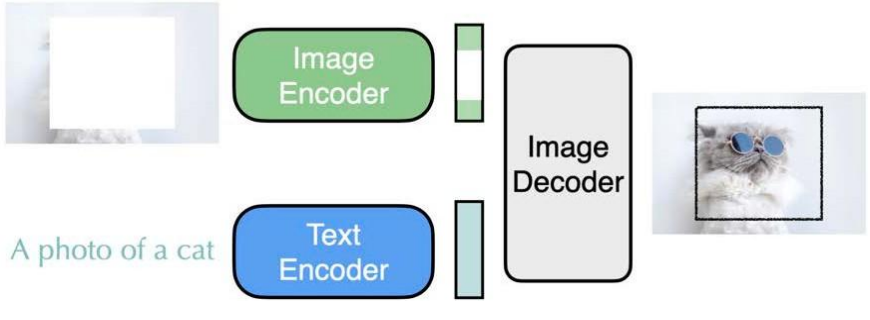
Cons:

1. Is not a generative model, thus it is not possible to generate a caption
2. current CLIP models cannot be used to provide high-level descriptions of a given image
3. usually needs a very large dataset as well as large batch sizes to offer decent performances

When to Use Masking-based Models?

Pros:

1. By learning to reconstruct data from both masked images and text, it jointly models their distributions
2. Removing negative examples can enable the use of smaller mini-batches without the need to finetune additional hyper-parameters



Masking Objective

Cons:

1. Need to leverage a decoder to map back the representation to the input space
2. An additional decoder might add an additional bottleneck which might make these methods less efficient than a purely contrastive one.

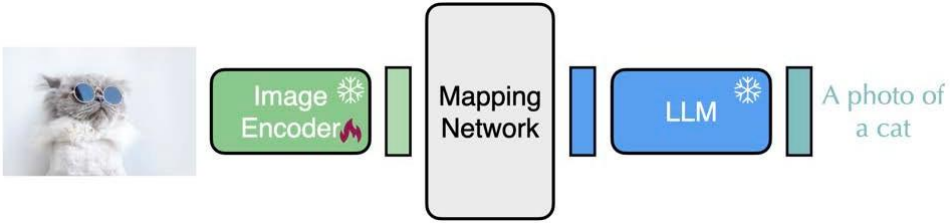
When to Use Pretrained Backbones?

Pros:

1. Can work with limited resource
2. Simple pipeline and framework

Cons:

1. VLMs will be impacted by the potential hallucination of the LLM.
2. VLMs could also be impacted by any bias coming from the pretrained models.
3. There might be an additional overhead in trying to correct the defect of the vision model or of the LLM.



VLMs from Pretrained Backbones

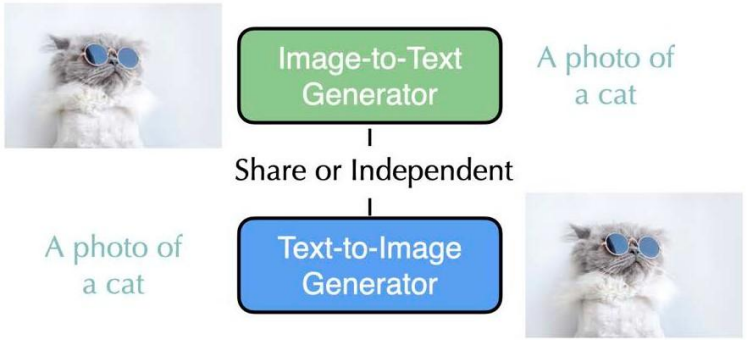
When to Use Generative Objectives?

Pros:

1. It might be easier to understand and assess what the model has learned when it is able to decode abstract representations in the input data space
2. Can learn an implicit joint distribution between text and images, which might be more suited for learning good representations than leveraging pretrained unimodal encoders.

Cons

1. They are more computationally expensive to train than their contrastive learning counterparts.
2. Not easy to train, especially when having two generative tasks (T2I and I2T)



Generative-Based

Disclaimer (Again)

- Seems very unsettled
  - Nobody seems to agree what the best way to combine images/text
  - Many kinds of architectures
  - Many ways to encode images
  - Many different training objectives
- I definitely didn’t cover everything
  - Unclear how to unify all of these different ideas
  - Feels very much like a bunch of different papers
- Learn what *you need and choose a model that works for your problem

Any Questions

Questions