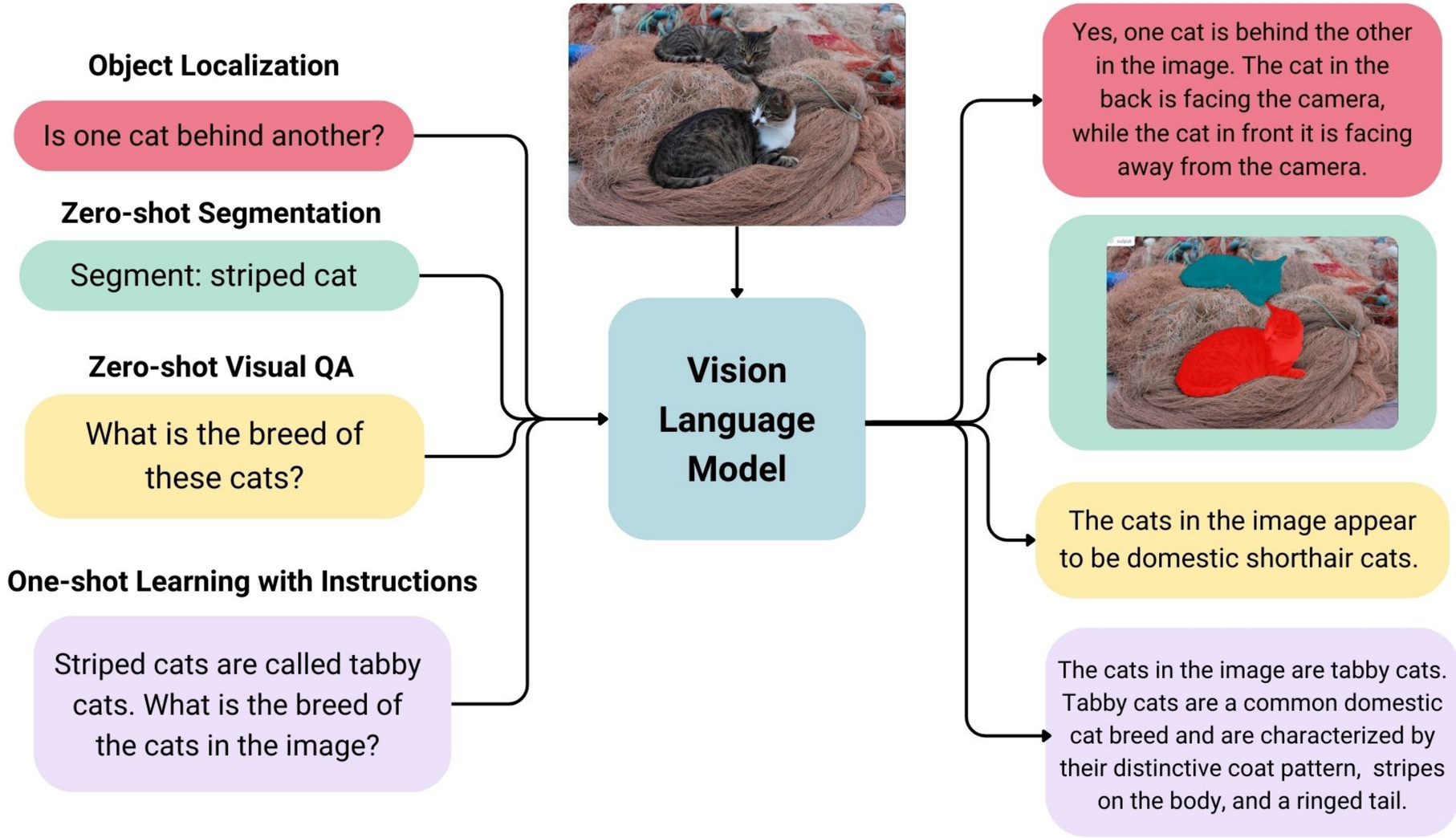
Multimodal Learning

EECE454 Intro. to Machine Learning Systems

Fall 2024

Overview

- **Today.** Multimodal Learning — the case of vision + language

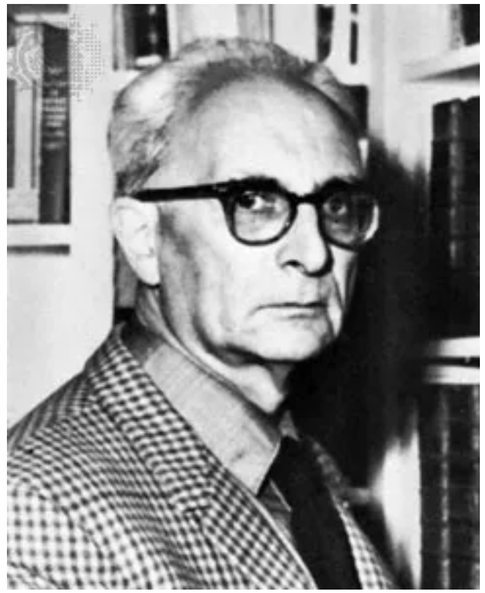


Overview

- **Popular approach.** Let LLMs be our central interface for thinking & reasoning
  - Why? Language shapes how we think (or at least we believe so)

George Boole

"That language is an instrument of human reason,
and not merely a medium for the expression of thought,
is a truth generally admitted."

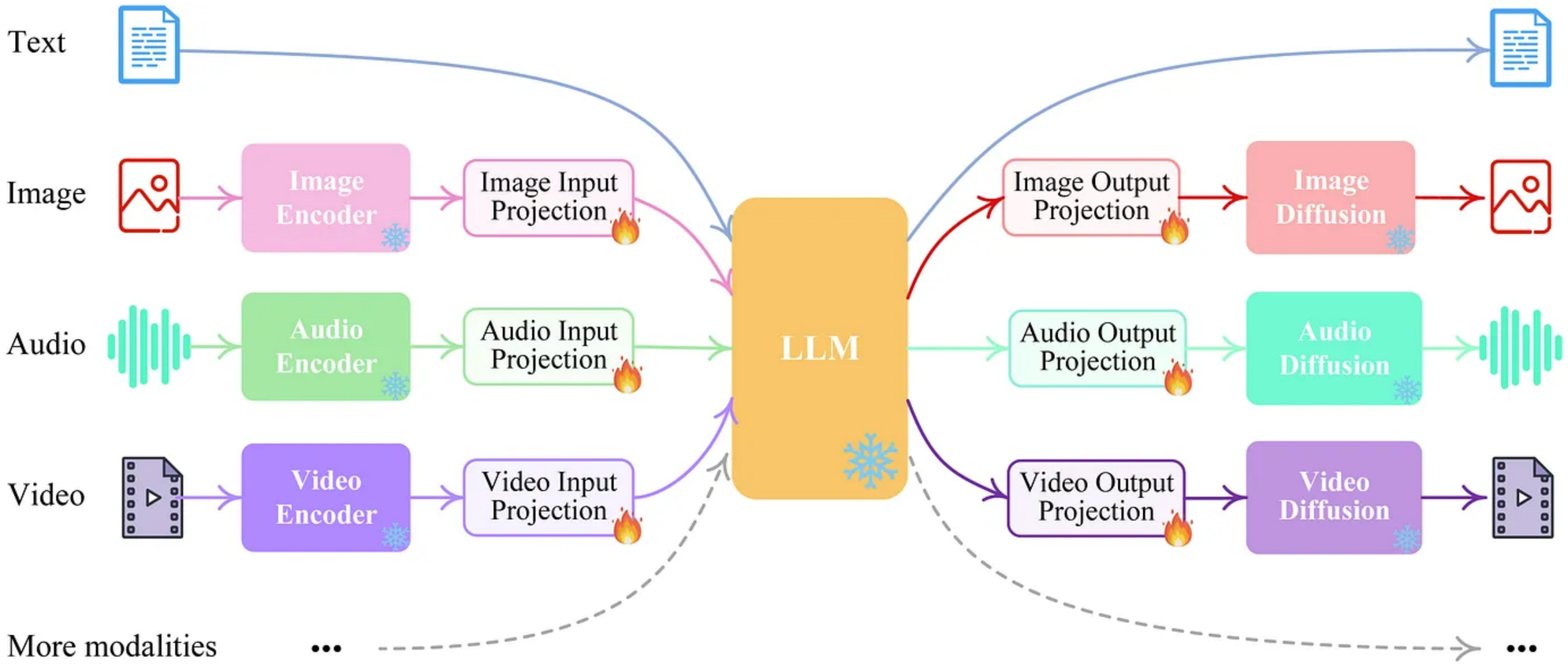


Claude Lévi-Strauss

"Language is a form of human reason,
which has its internal logic
of which man knows nothing."

Overview

- To let LLMs process multimodal information
  - Input. Various modalities encoded into a form that LLM can understand
  - Output. Acquire LLMs with tools that can be queried with text



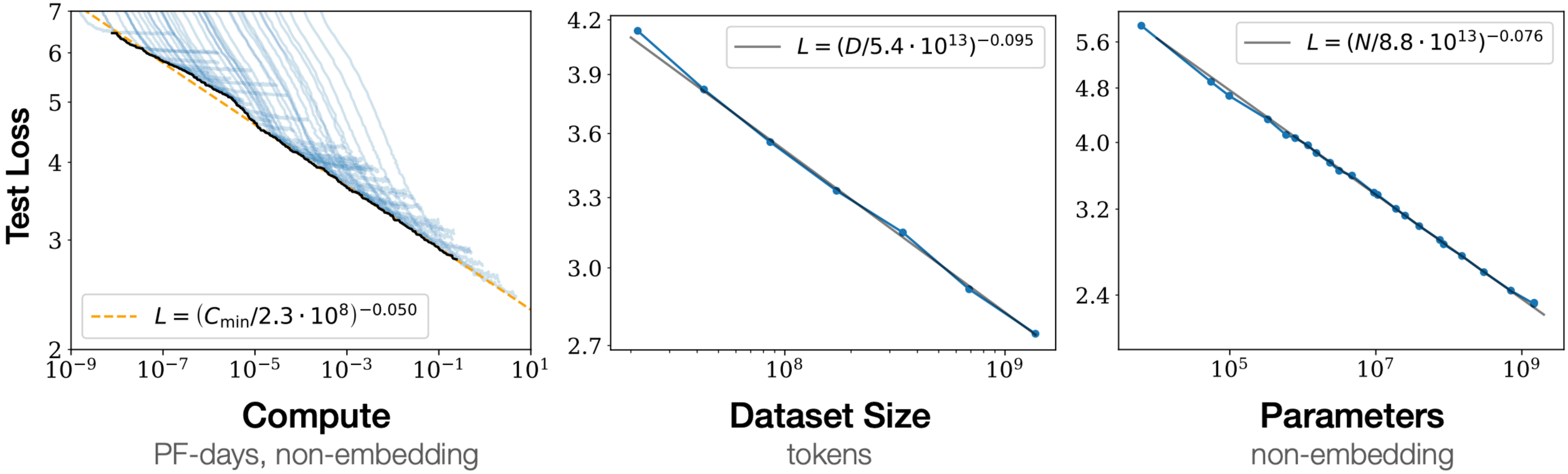
Overview

- **Scope.** How to make LLMs process visual inputs
  - Architecture. Vision Transformer
  - Training encoders. CLIP
  - Overall pipeline. LLaVA

Vision Transformers

Vision Transformers

- **Question.** Can we use transformers to process visual inputs?
  - Hope#1. Transformers are scalable
    - performance gets better, seemingly without limit, with more data & parameter & compute
  - Hope#2. Handling text and image training within a unified architecture

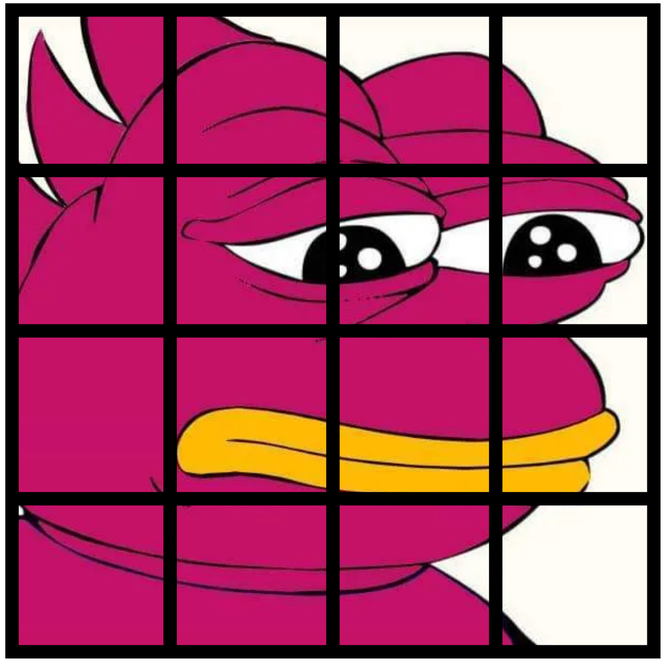


Idea

- Break image down into a sequence of low-resolution patches (tokens)
  - Typically 14x14 or 16x16

Image

$\mathbf{x} \in \mathbb{R}^{H \times W \times C}$



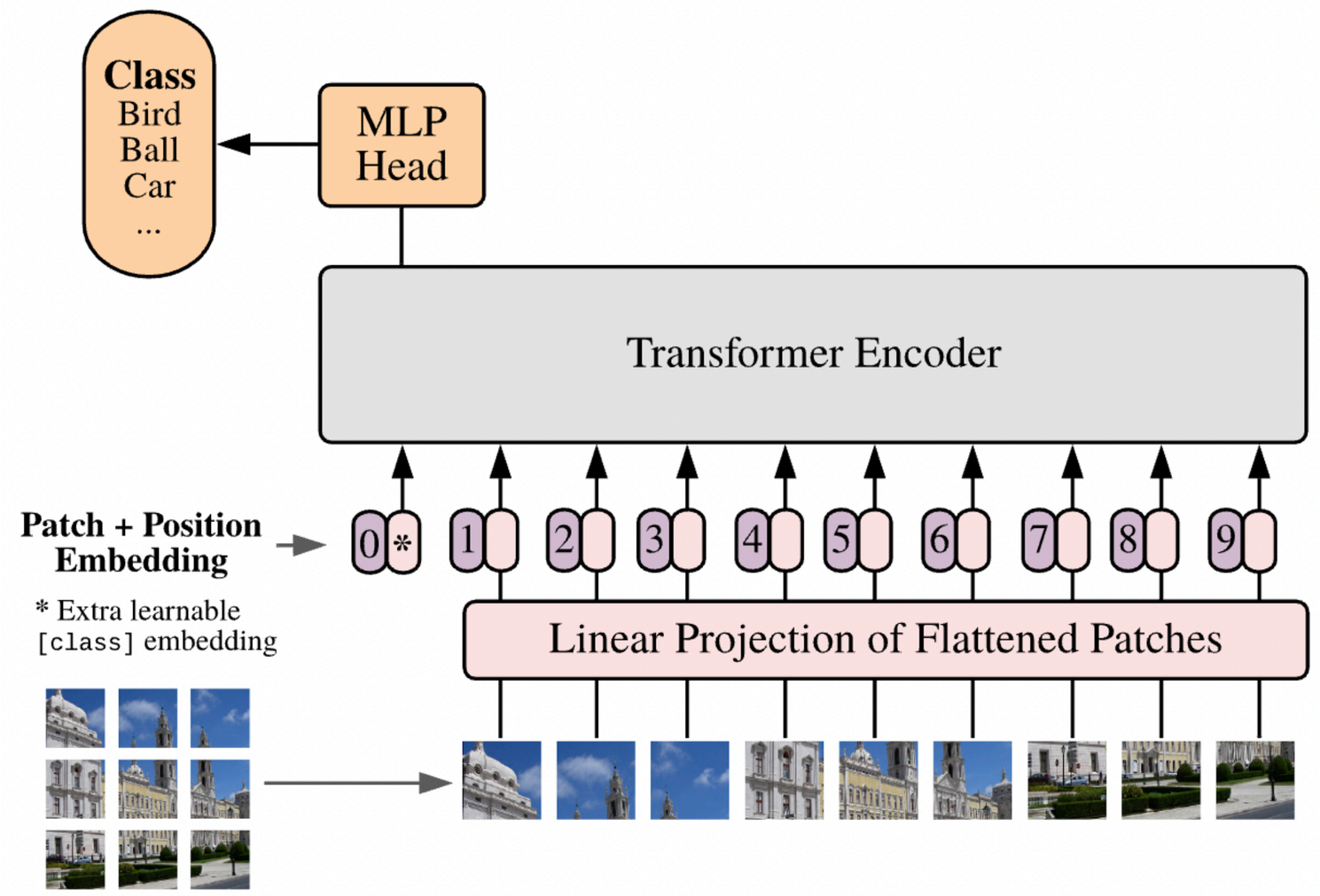
Sequence of Patches

$\mathbf{x}_i \in \mathbb{R}^{P \times P \times C}$

(total HW/P^2 patches)

Idea

- Then, of course, (1) embed these tokens
  (2) process with transformer <— Typically trains a **class token**, jointly

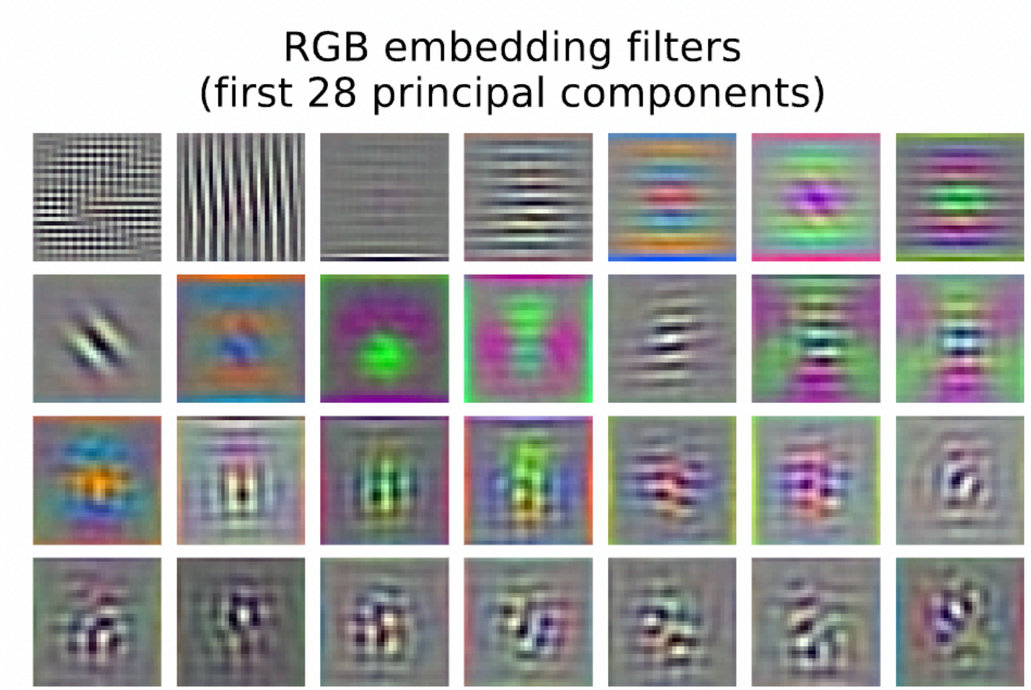


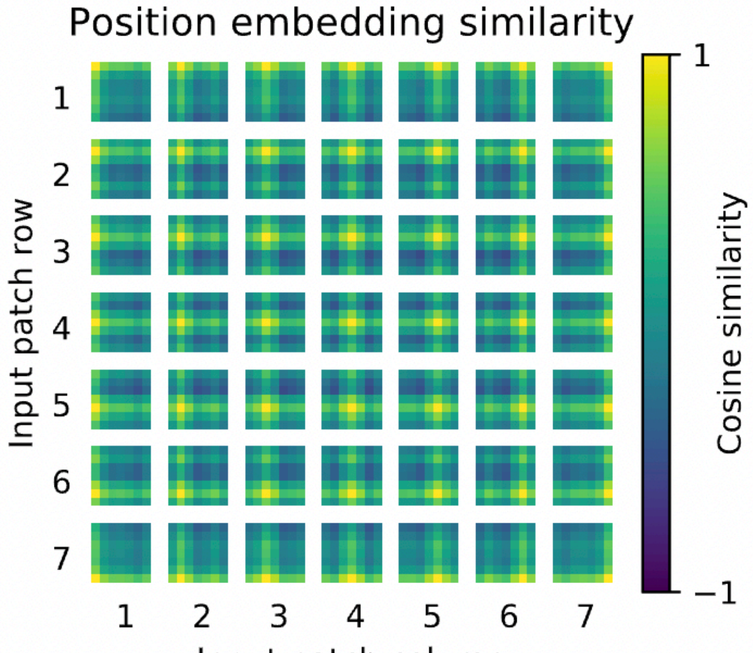
Embedding

- Learned linear embedding + Learned 1D positional encoding

$$\mathbf{z}_i = \mathbf{W}_{\text{emb}}\, \text{flatten}(\mathbf{x}_i) + \mathbf{e}_i$$

  - Linear embedding. Depends only on how patch looks; $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{d \times (P^2 C)}$
  - Positional encoding. Depends only on the location of the patch; $\mathbf{e}_1, \ldots, \mathbf{e}_{HW/P^2} \in \mathbb{R}^d$





Drawback

- **Sample Complexity.** Comparing with ConvNet, we are imposing less handcrafted bias
  (syn. architectural bias, inductive bias)
  - ConvNet. We architecturally constrain the learned function to have strong locality
  - ViTs. No such constraint, but we expect the model to learn from data
- Thus, commonly believed that ViTs need more samples to perform better than ConvNets

Drawback

- **Sample Complexity.** Comparing with ConvNet, we are imposing less handcrafted bias (syn. architectural bias, inductive bias)
  - ConvNet. We architecturally constrain the learned function to have strong locality
  - ViTs. No such constraint, but we expect the model to learn from data
- Thus, commonly believed that ViTs need more samples to perform better than ConvNets
- **Solutions.**
  - Hybrid architecture
    - e.g., ConvViT
  - Distilling convolutional priors
    - e.g., DeiT
  - Self-supervised pre-training
    - e.g., MAE

Drawback

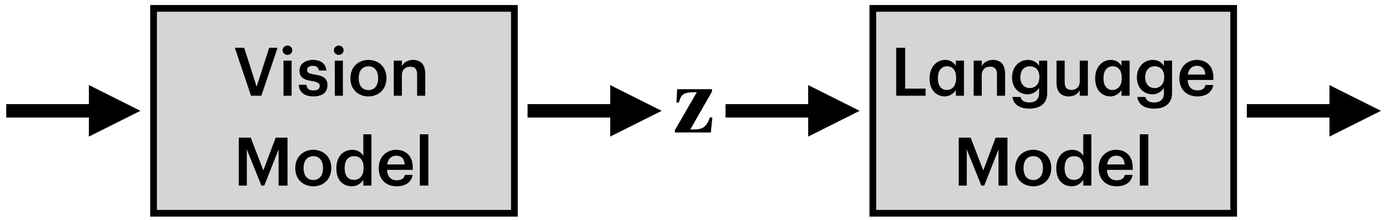
- **Computation.** Typically requires more parameters to work well...

- **Solution.** Model compression & Lightweight versions

CLIP

CLIP

- **Question.** How can we let language models utilize features from visual inputs?



"Lion is an animal which usually lives in ..."

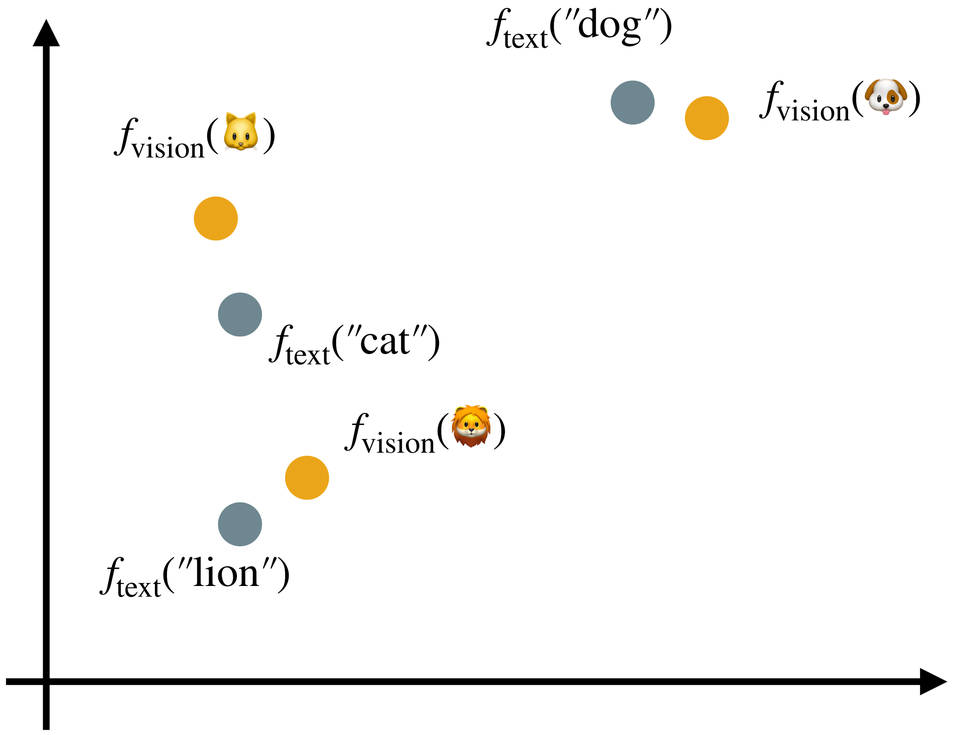
Rough overview

- Construct a shared feature space for both text and image.
  - Training. Encourage the “lion” and 🦁 to be close in this joint feature space, i.e.,

$$f_{\text{text}}(\text{"lion"}) \approx f_{\text{vision}}(🦁)$$

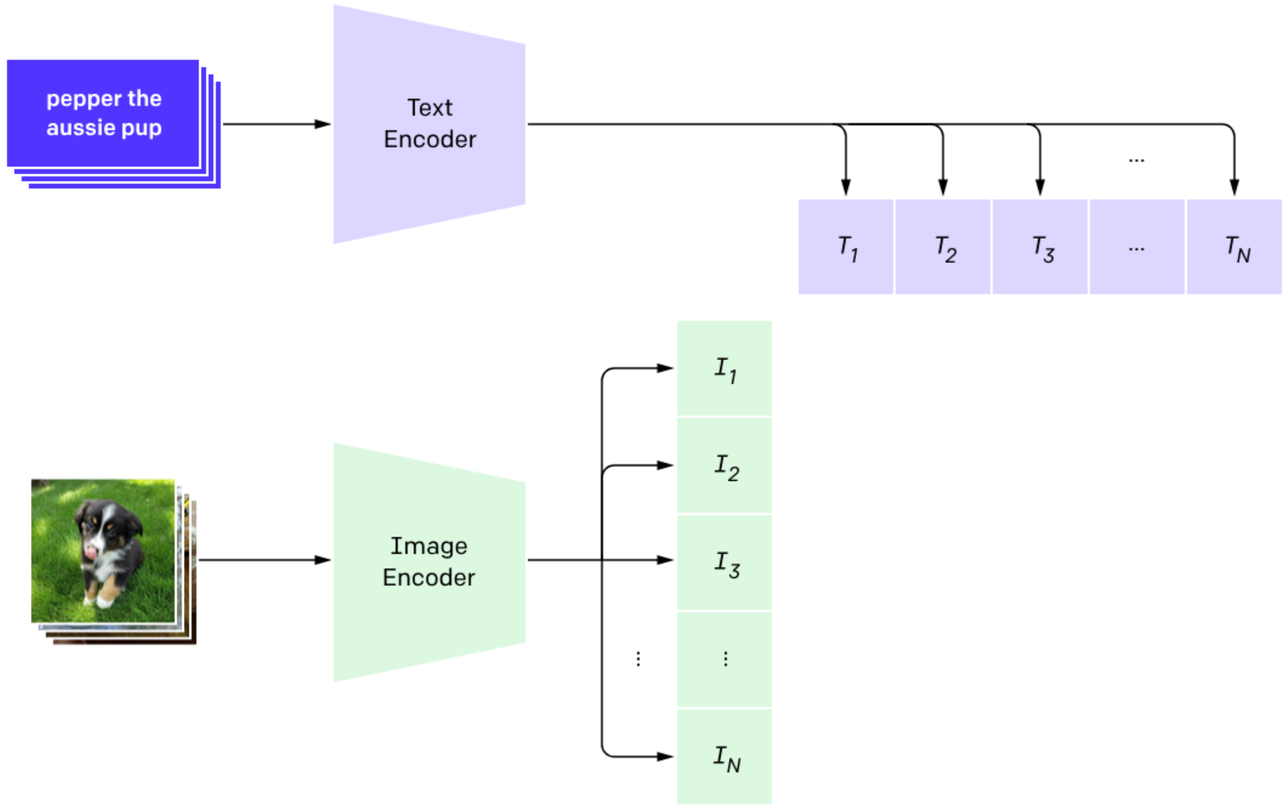
  - Using. Given an image, find a text whose embedding is the closest to the image feature

$$\min_{\mathbf{x}} \quad \text{distance}\left(f_{\text{text}}(\mathbf{x}), f_{\text{image}}(🐯)\right)$$



Training

- Done by the contrastive pre-training
  - Draw a batch of image-text pairs with N samples in it
  - Generate image and text embeddings $(I_1, T_1), \ldots, (I_N, T_N)$
  - Text. Transformer, with <EOS> token (end-of-sentence) being the feature
  - Image. ViT, with <cls> token (class) being the feature



Training

- Train with the InfoNCE loss

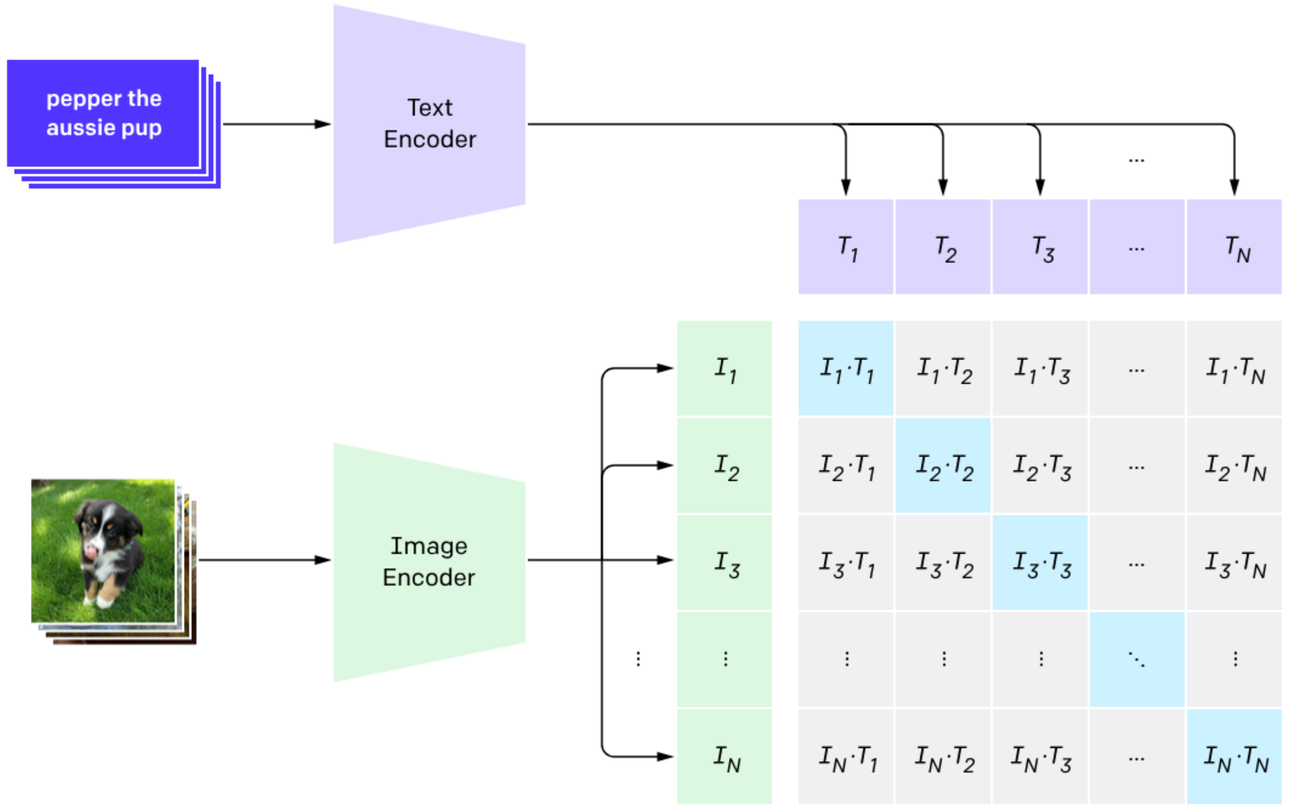
$$L = \frac{1}{N}\sum_{i=1}^{N}\frac{\ell(I_i) + \ell(T_i)}{2}$$

  - Here, the loss are

$$\ell(I_i) = -\log\frac{\exp(I_i^\top T_i/\tau)}{\sum_j \exp(I_i^\top T_j/\tau)}$$

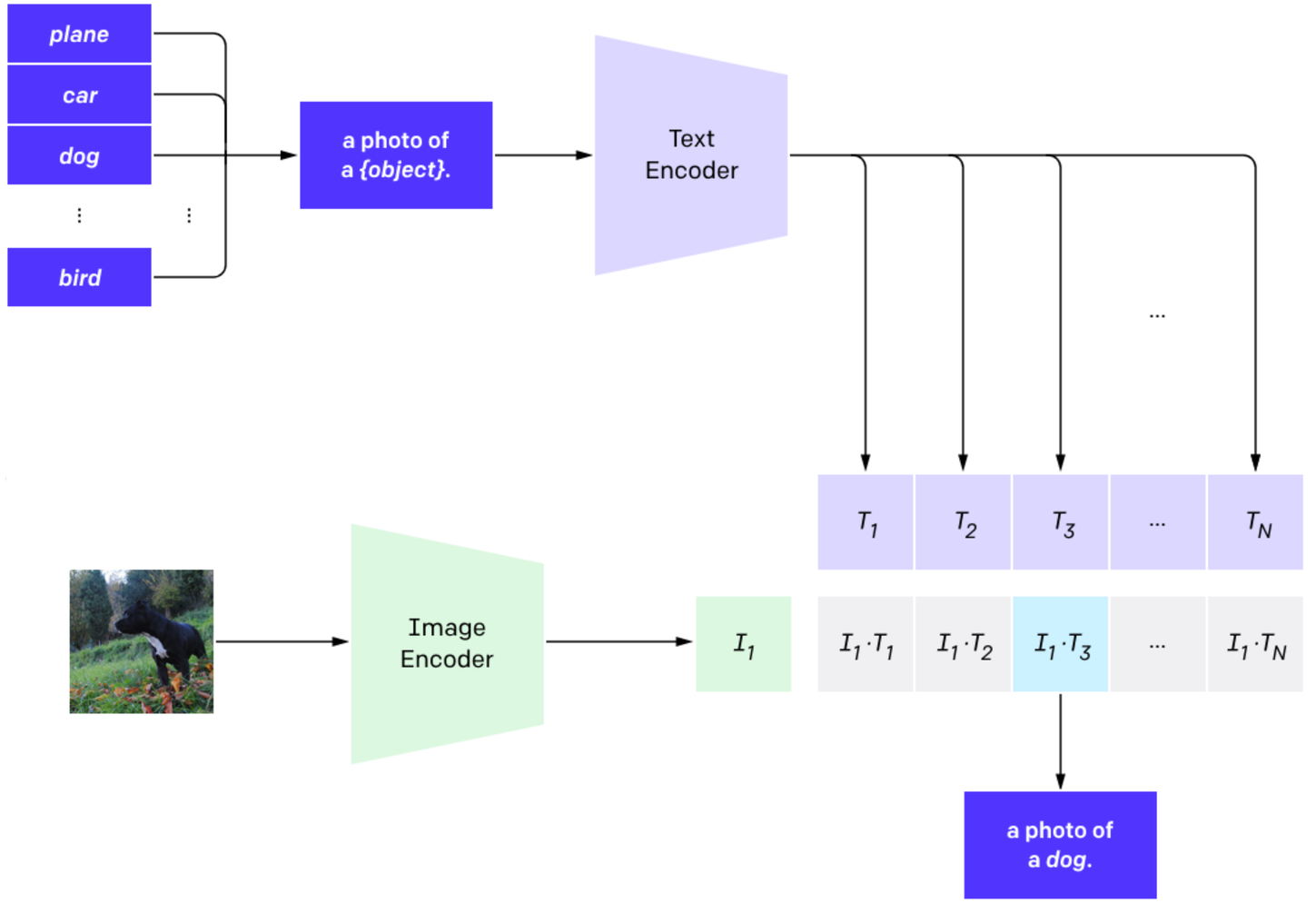
$$\ell(T_i) = -\log\frac{\exp(I_i^\top T_i/\tau)}{\sum_j \exp(I_j^\top T_i/\tau)}$$

(here, τ is the "temperature" hyperparameter)



Inference

- Combine candidate words with a nice prompt, then compare with the target image
  - Allows an "open-set classification" i.e., can classify image with unlimited set of target classes configured by natural language classes
  - Note. The quality of prompt matters; the prompt can be fine-tuned, too.



Inference

SUN397

television studio (90.2%) Ranked 1 out of 397 labels

✓ a photo of a **television studio**.

× a photo of a **podium indoor**.

× a photo of a **conference room**.

× a photo of a **lecture room**.

× a photo of a **control room**.

Inference

Stanford Cars

2012 Honda Accord Coupe (63.3%) Ranked 1 out of 196 labels

✓ a photo of a **2012 honda accord coupe**.

× a photo of a **2012 honda accord sedan**.

× a photo of a **2012 acura tl sedan**.

× a photo of a **2012 acura tsx sedan**.

× a photo of a **2008 acura tl type-s**.

Inference

German Traffic Sign Recognition Benchmark (GTSRB)

red and white triangle with exclamation mark warning (45.7%) Ranked 1 out of 43 labels

✓ a zoomed in photo of a "**red and white triangle with exclamation mark warning**" traffic sign.

× a zoomed in photo of a "**red and white triangle with black right curve approaching warning**" traffic sign.

× a zoomed in photo of a "**red and white triangle car skidding / slipping warning**" traffic sign.

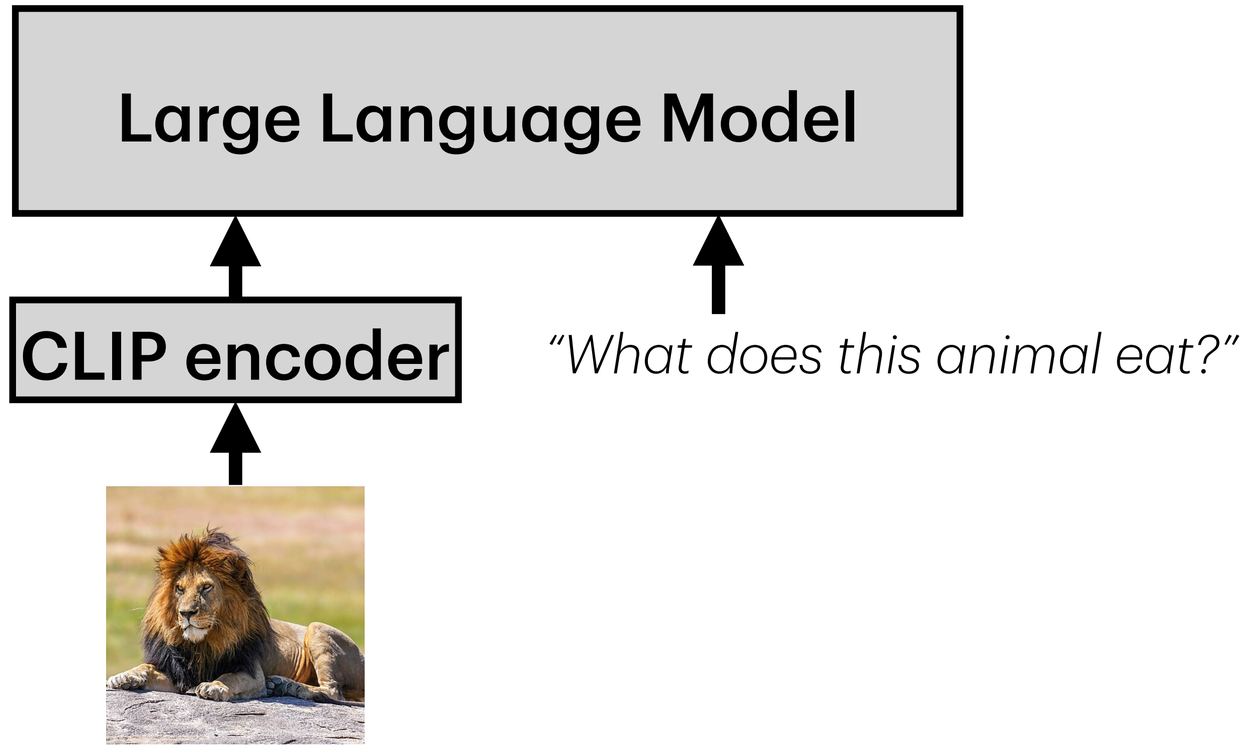
× a zoomed in photo of a "**red and white triangle rough / bumpy road warning**" traffic sign.

× a zoomed in photo of a "**red and white triangle with black left curve approaching warning**" traffic sign.

LLaVA

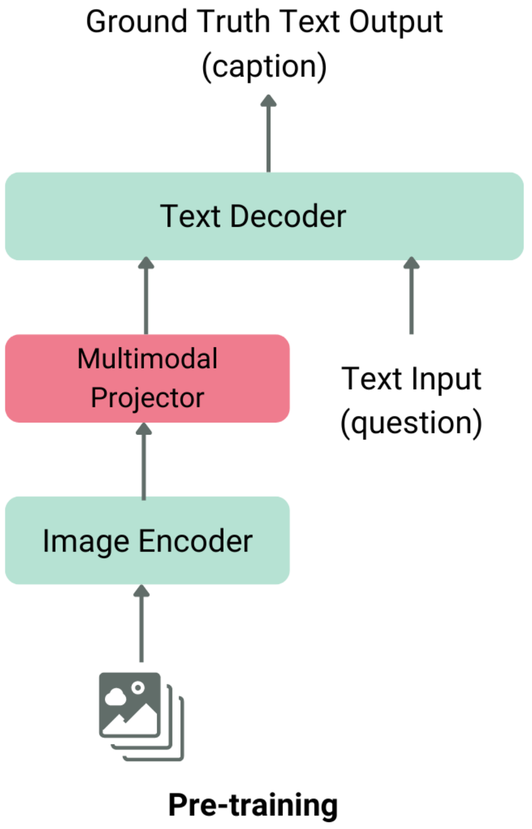
LLaVA

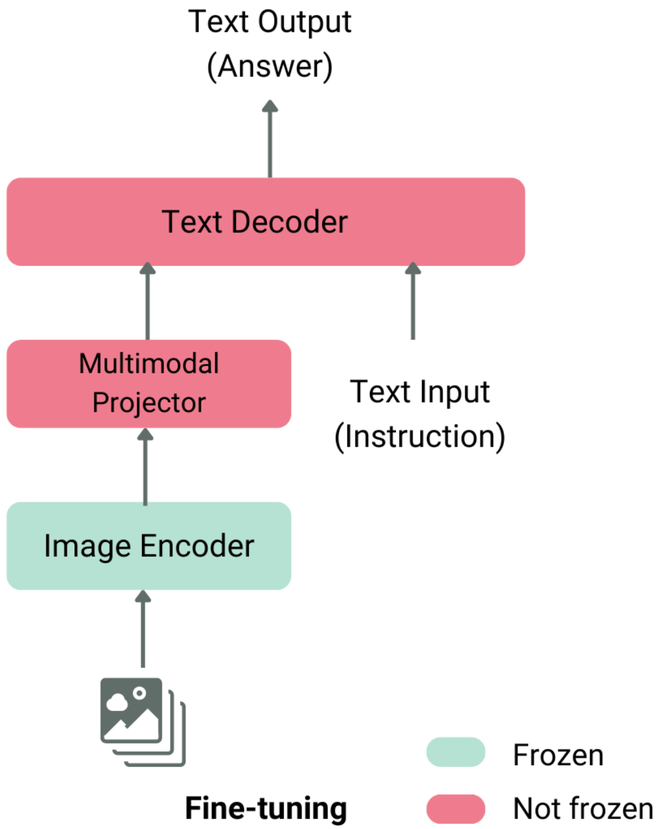
- **Idea.** Simply use the outputs of (pre-trained) vision encoder as a prompt for the (pre-trained) LLM
  - Use CLIP as our vision encoder
  - Problem. (1) LLM features are not "well-aligned" with with CLIP features
    (2) LLMs are not instructed to do visual Q&A



LLaVA

- **Solution.**
  - Aligning. Add a trainable projection between CLIP & LLM
    - train with image-caption pairs
    - freeze CLIP & LLM
  - Instruction tuning. Fine-tune using a visual instruction tuning dataset
    - Use GPT for annotation
    - freeze CLIP encoder



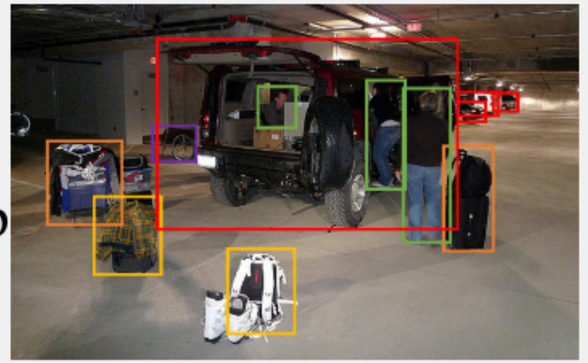


Visual Instruction Tuning

- Visual instruction tuning dataset is collected using text-only GPT
- **Prompting GPT.** GPT is provided with the textual description of an image (not the image itself)
  - Captions & bounding boxes

Context type 1: Captions

A group of people standing outside of a black vehicle with various luggage.
Luggage surrounds a vehicle in an underground parking area
People try to fit all of their luggage in an SUV.
The sport utility vehicle is parked in the public garage, being packed for a trip
Some people with luggage near a van that is transporting it.



Context type 2: Boxes

person: [0.681, 0.242, 0.774, 0.694], person: [0.63, 0.222, 0.686, 0.516], person: [0.444, 0.233, 0.487, 0.34], backpack: [0.384, 0.696, 0.485, 0.914], backpack: [0.755, 0.413, 0.846, 0.692], suitcase: [0.758, 0.413, 0.845, 0.69], suitcase: [0.1, 0.497, 0.173, 0.579], bicycle: [0.282, 0.363, 0.327, 0.442], car: [0.786, 0.25, 0.848, 0.322], car: [0.783, 0.27, 0.827, 0.335], car: [0.86, 0.254, 0.891, 0.3], car: [0.261, 0.101, 0.787, 0.626]

Visual Instruction Tuning

- Use GPT to generate three types of Q&As
- **Q&A conversations.** Given the prompt, GPT simulates both the person who asks, and also the person who answers.

Response type 1: conversation
Question: What type of vehicle is featured in the image?
Answer: The image features a black sport utility vehicle (SUV).
Question: Where is the vehicle parked?
Answer: The vehicle is parked in an underground parking area, likely in a public garage.
Question: What are the people in the image doing?
Answer: The people are trying to fit all of their luggage into the SUV, likely preparing for a trip.

Visual Instruction Tuning

- **Detailed descriptions.** GPT generates a very detailed description of the image
  - Uses bounding box information to fill in the details

Response type 2: detailed description

The image is an underground parking area with a black sport utility vehicle (SUV) parked. There are three people in the scene, with one person standing closer to the left side of the vehicle, another person in the middle, and the third person on the right side. They are all working together to pack their luggage into the SUV for a trip.

In the parking area, various luggage items are scattered around the vehicle. There are two backpacks, one located near the left rear wheel and the other closer to the right side of the vehicle. Additionally, there are two suitcases, one on the right side of the car and another further away near the center of the parking area. A bicycle can also be seen on the left side of the vehicle.

Other cars are parked around the main SUV, with one car positioned behind it and slightly to the left, another behind and slightly to the right, and the third car further behind on the right side.

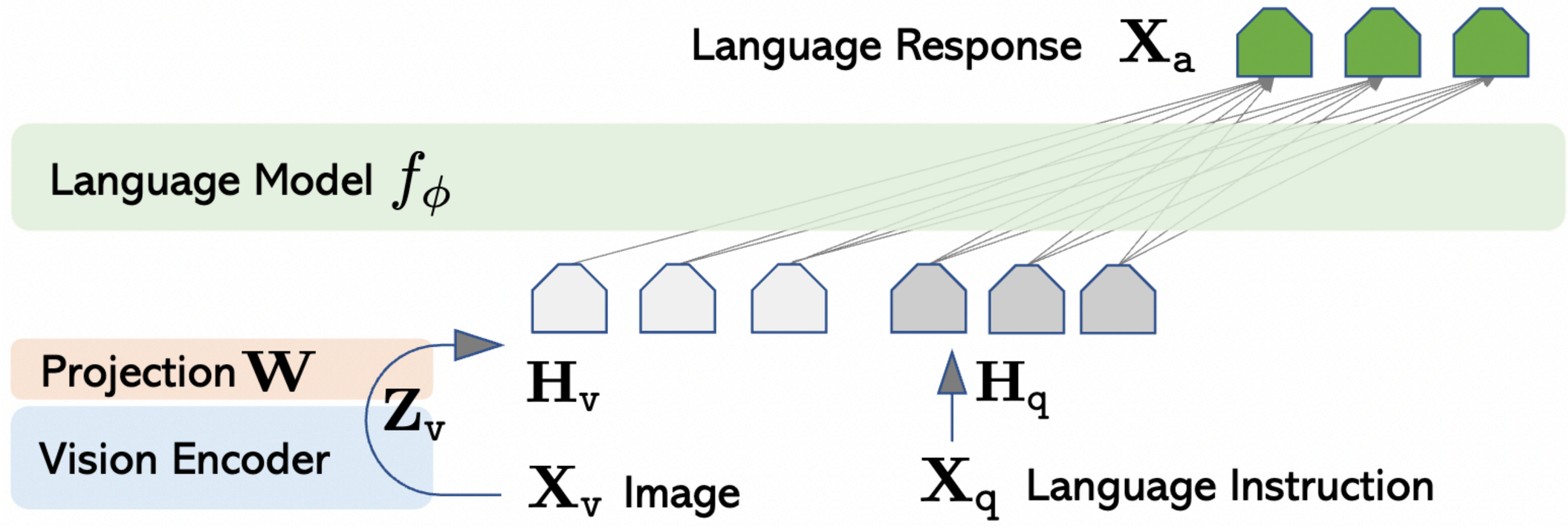
Visual Instruction Tuning

- **Complex reasoning.** GPT generates both the question and the answer that needs an in-depth understanding of the content of the image

Response type 3: complex reasoning
Question: What challenges do these people face?
Answer: In the image, a group of people is standing outside a black SUV in a parking area, surrounded by various pieces of luggage, including suitcases and backpacks. They are facing the challenge of fitting all their luggage into the black SUV. There are multiple suitcases and backpacks to be packed, which suggests that the group has a significant amount of belongings to accommodate. They might have to strategize and arrange the luggage efficiently to ensure that everything fits properly into the vehicle. Additionally, they need to consider the comfort of the passengers and visibility while driving, so the placement of the luggage must not obstruct the driver's view or make the passengers uncomfortable during the trip.

Fine-tuning

- Plug in the **[visual prompt, text instruction]** as an input.
  - Update the model parameters to generate output closers to **[text response]**



Other materials

- See the following materials for a more general overview:
  - **Beginner.** https://lilianweng.github.io/posts/2022-06-09-vlm/
  - **Advanced.** https://arxiv.org/abs/2405.17247

Cheers