



Large Vision Language Models

LLMs Series

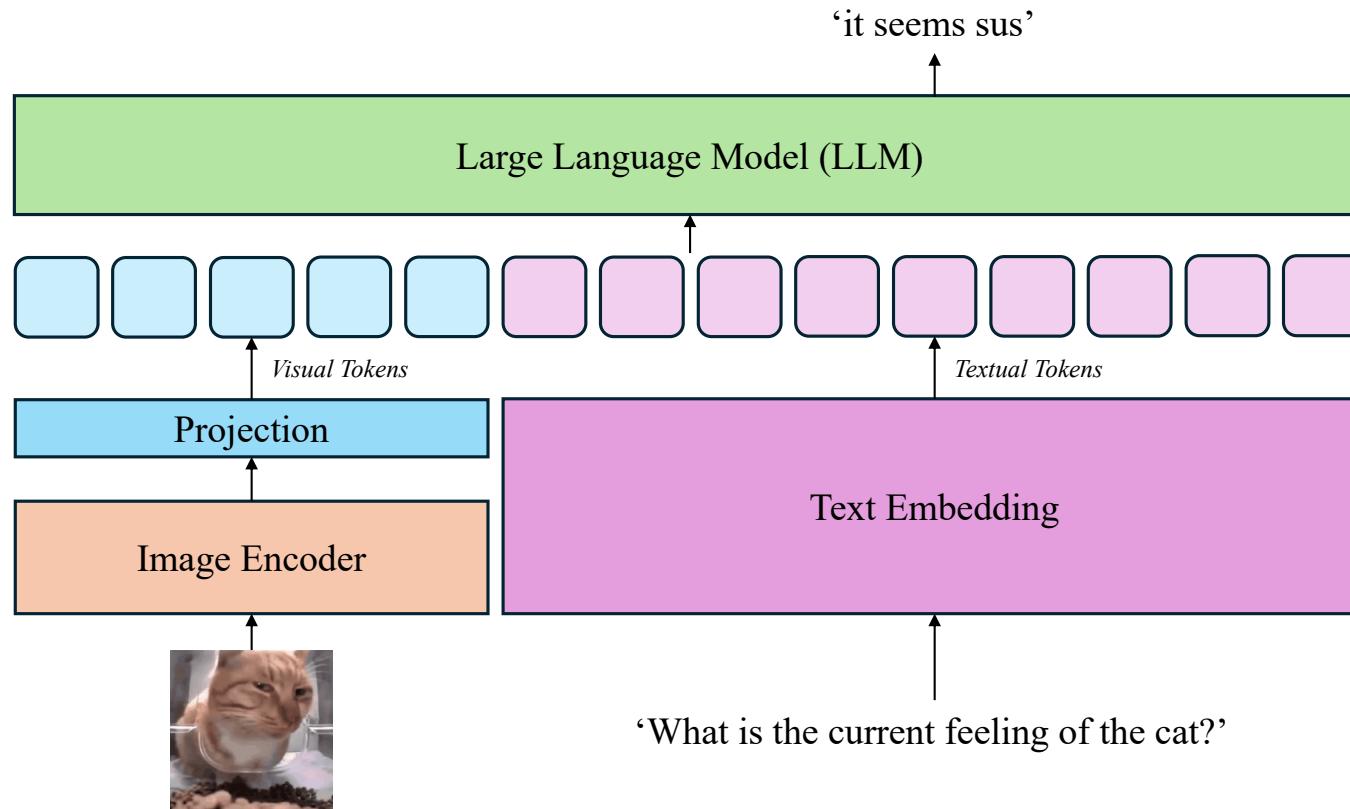
Dinh-Thang Duong – TA

Nguyen-Thuan Duong – TA

Objectives

❖ Content

Output:



In this study session, we will discuss about:

1. Introduction to Multimodal LLMs and Large Vision Language Models (LVLMs).
2. How to prompt LVLMs to solve vision and/or language tasks?
3. How to train LVLMs on a specific task?

Outline

- Introduction
- In-context Learning with LVLMs
- Training LVLMs
- Question

 AI

AI VIET NAM
@aivietnam.edu.vn

Introduction

Introduction

❖ Getting Started

Textual Description (Prompt):

Write a python function that receive an image and plot it using matplotlib library.

 You
Write a python function that receive an image and plot it using matplotlib library.

 ChatGPT
To create a Python function that receives an image and plots it using the Matplotlib library, you can use the following code as an example. You'll need to have Matplotlib and a library like PIL (Pillow) installed to work with images:

```
python
import matplotlib.pyplot as plt
from PIL import Image

def plot_image(image_path):
    try:
        # Open the image using PIL (Pillow)
        img = Image.open(image_path)

        # Create a Matplotlib figure and axis
        fig, ax = plt.subplots()
```

 Copy code



Introduction

❖ What are Large Language Models (LLMs)?



You

write a python code that print "hello world"



ChatGPT

Certainly! Here's a simple Python code that prints "Hello, World!" to the console:

python

Copy code

```
print("Hello, World!")
```

You can run this code in a Python interpreter, and it will display "Hello, World!" as the output.



ChatGPT App:

1. User inputs a prompt (greetings, task...).
2. ChatGPT (Chatbot) outputs a appropriate response.

Introduction

❖ What are Large Language Models (LLMs)?

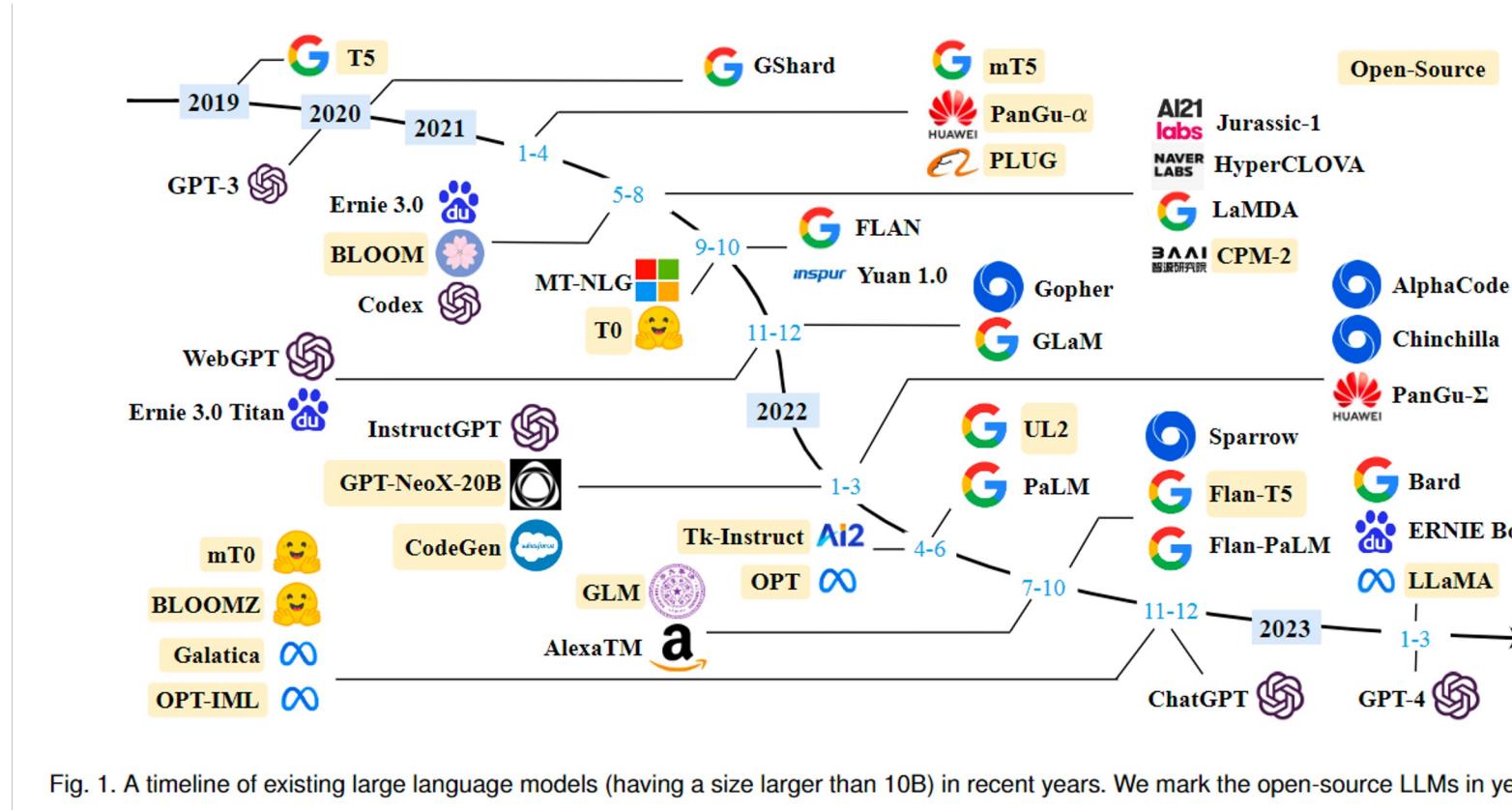


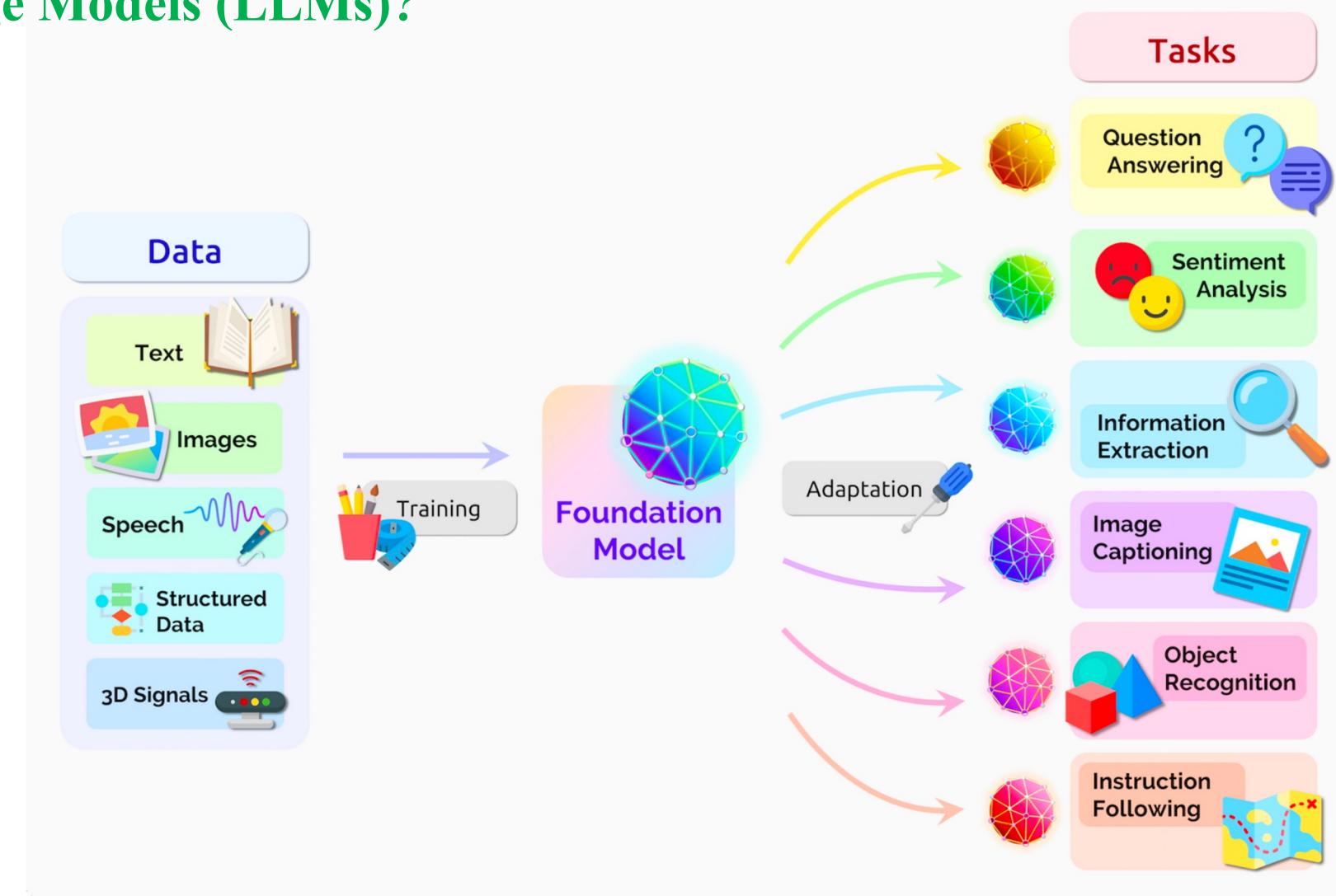
Fig. 1. A timeline of existing large language models (having a size larger than 10B) in recent years. We mark the open-source LLMs in yellow color.

LLMs (Large Language Models): AI models (language models) that were trained on a very large corpus of text. This made them capable of performing various NLP tasks with high precision.

Introduction

❖ What are Large Language Models (LLMs)?

LLMs are often pretrained on a vast majority of data and designed to be adaptable to a wide variety of tasks (Foundation models).

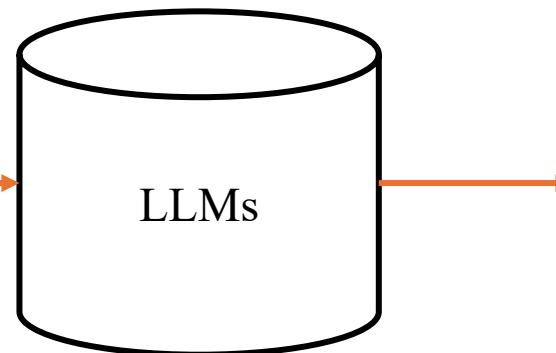


Introduction

❖ LLMs I/O

Input Text

Write a python function that receive an image and plot it using matplotlib library.



Given a “prompt”, LLMs can generate an appropriate response.

Output Text

AI You Write a python function that receive an image and plot it using matplotlib library.

ChatGPT To create a Python function that receives an image and plots it using the Matplotlib library, you can use the following code as an example. You'll need to have Matplotlib and a library like PIL (Pillow) installed to work with images:

```
python
import matplotlib.pyplot as plt
from PIL import Image

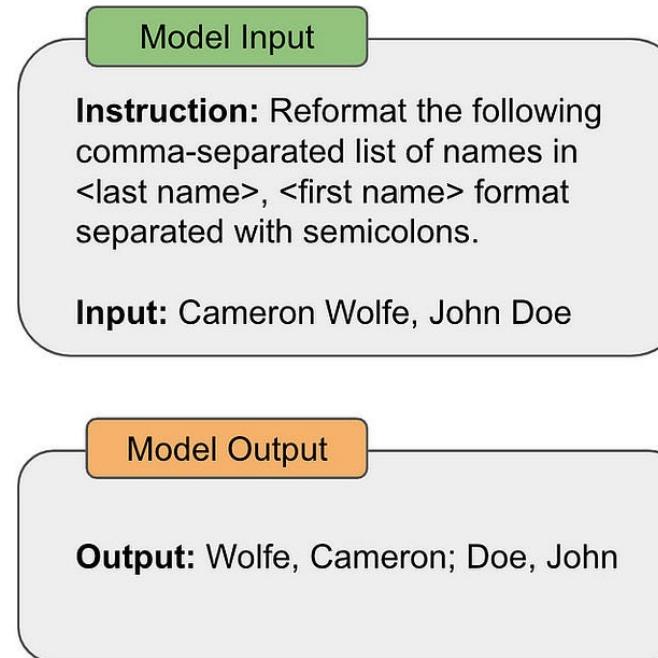
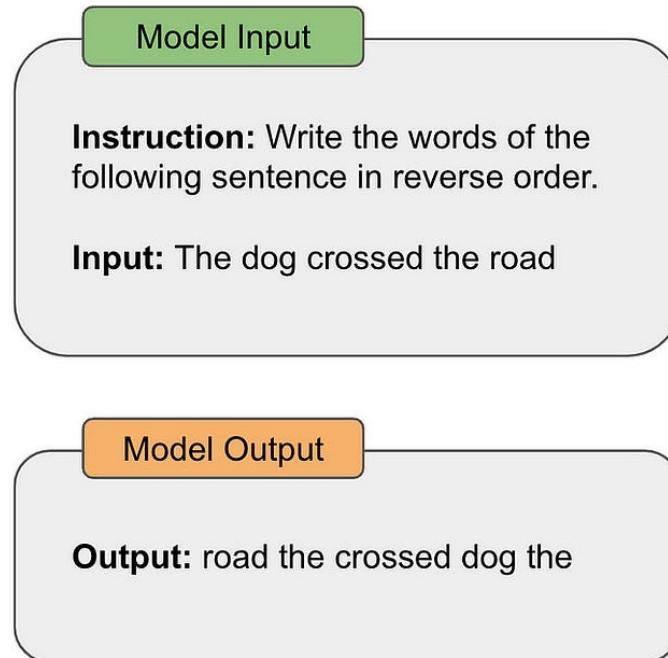
def plot_image(image_path):
    try:
        # Open the image using PIL (Pillow)
        img = Image.open(image_path)

        # Create a Matplotlib figure and axis
        fig, ax = plt.subplots()
```

Copy code

Introduction

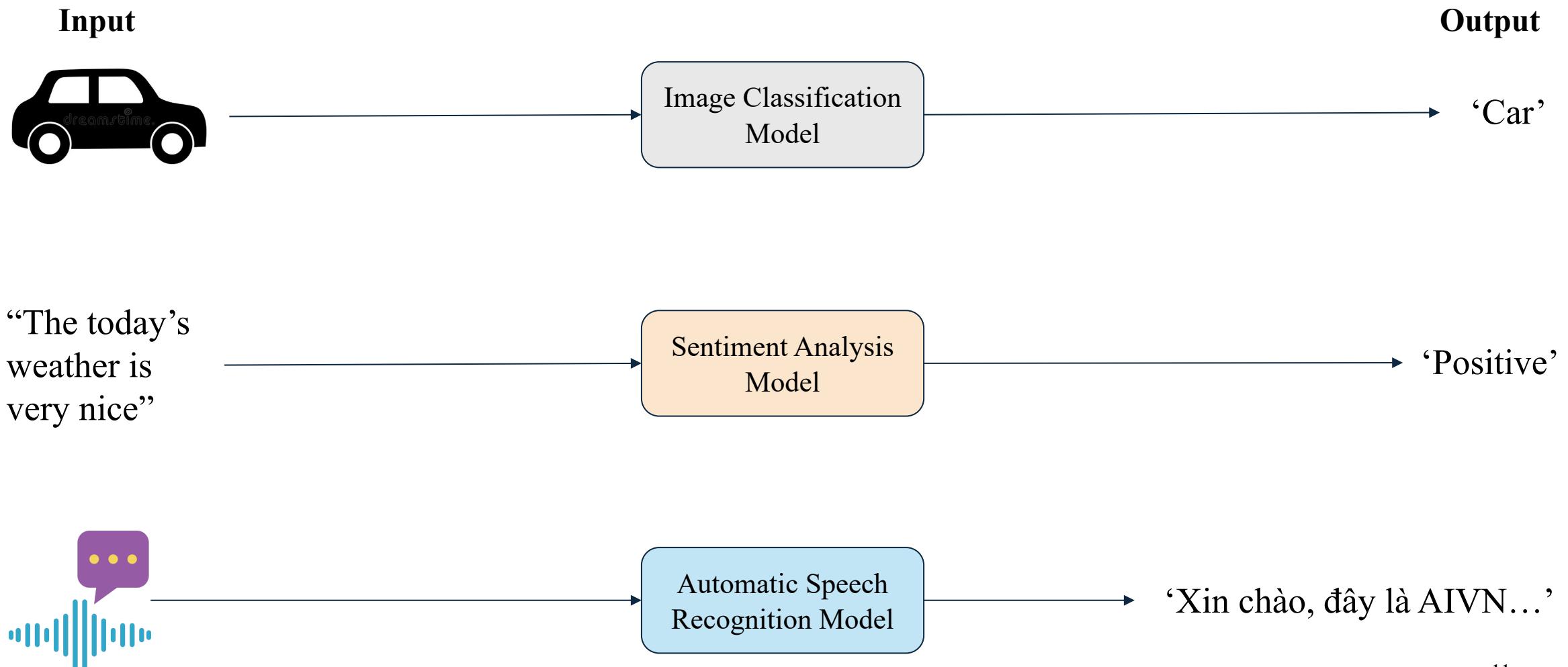
❖ Generative AI Prompting



Prompting: Refers to a process of providing an input, usually in the form of text data, to a generative AI model to generate a specific output.

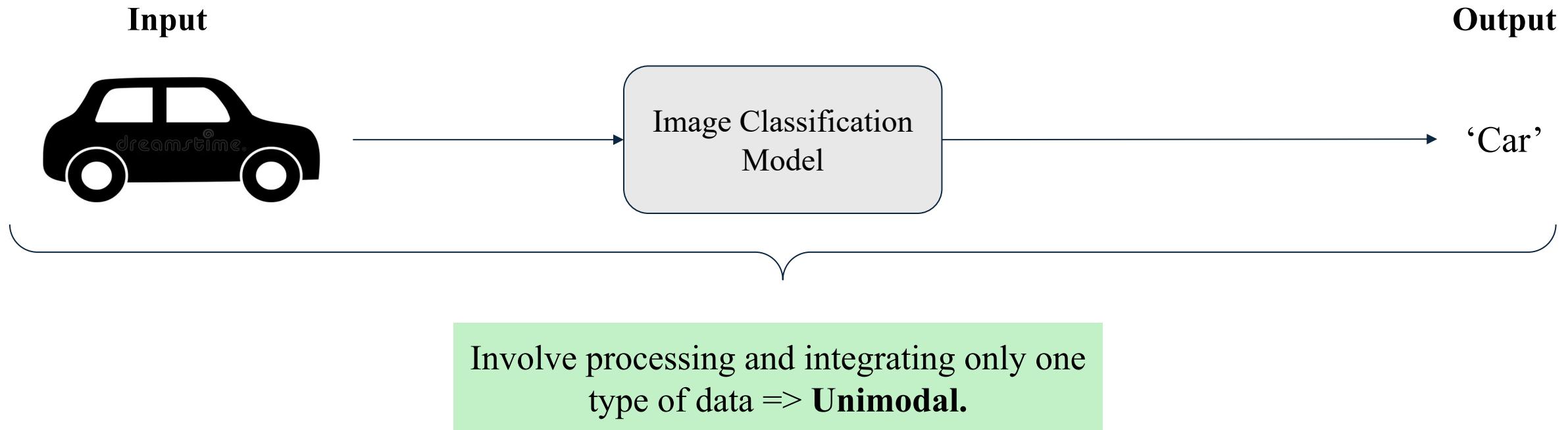
Introduction

❖ From unimodality to multimodality



Introduction

❖ From unimodality to multimodality



Introduction

❖ From unimodality to multimodality: VQA Task

Answer: No



Answer: Yes



complementary scenes

Tuple: <girl, walking, bike>

Question: Is the girl walking the bike?

Input:

- Image (Context).
- Text (Question).

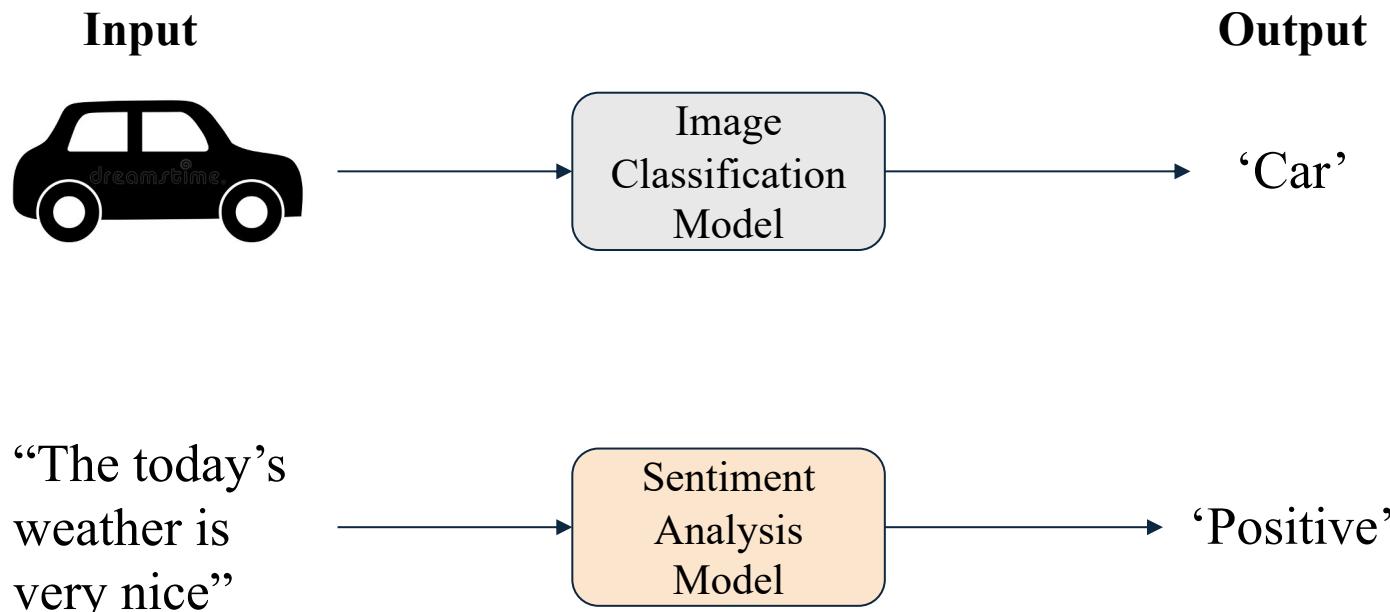
Output:

- Text (Answer).

Involve processing and integrating multi type of data => **Multimodal**.

Introduction

❖ What is multimodality?



Multimodality refers to the ability of a system to process and integrate information from multiple types of data, such as text, images, audio, or video. In AI, multimodal models are designed to understand and generate responses by combining insights from these diverse inputs, enabling more comprehensive reasoning and interaction.

Introduction

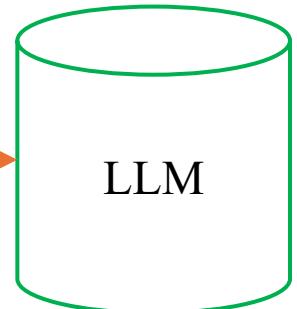
❖ Idea: How to make LLM to understand and perform vision tasks?

Image:



Question: What is the feeling of the cat in this image?

?

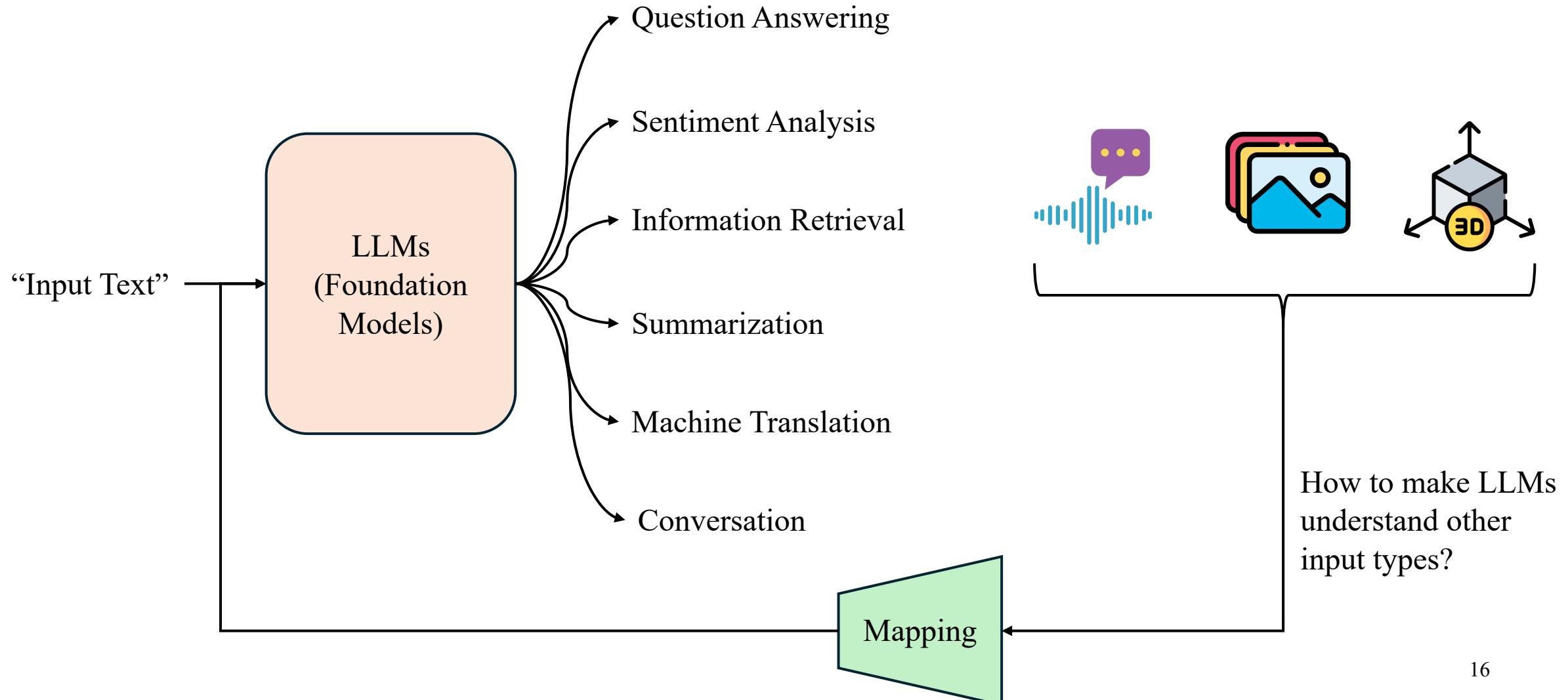


LLM only accepts text, how can an image be feeded into?

Figure: Illustration of a VQA input.

Introduction

❖ Activate multimodality to LLMs



Introduction

❖ Multimodal Large Language Models

AI You



How many people are there in this image?

ChatGPT

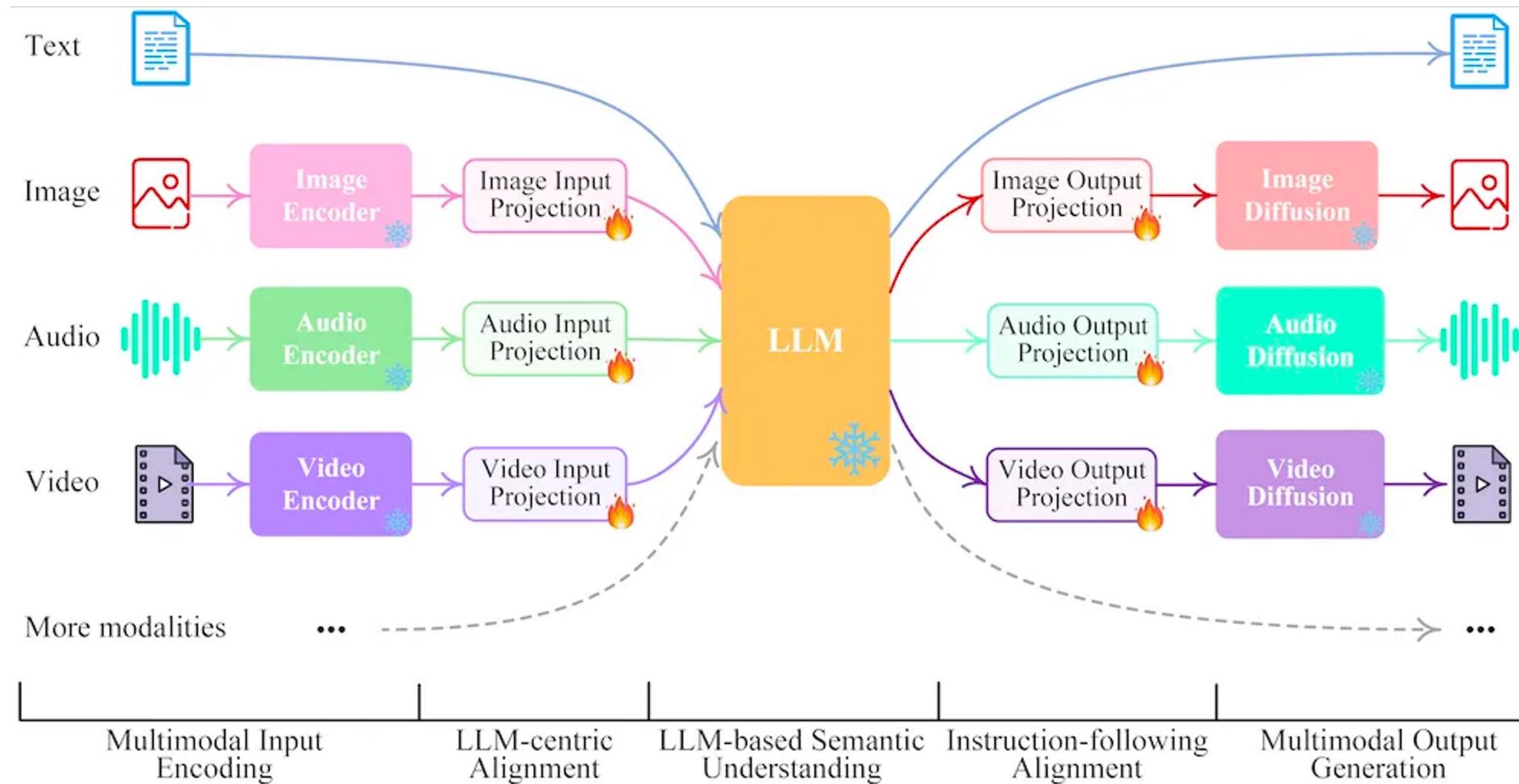
There are four people in this image.

⋮ ⏵ ⏴ ⏵ ⏴

Multimodal Large Language Models (M-LLMs) are AI systems that process and generate responses from multiple input types like text and images. They enable advanced tasks such as image captioning, visual question answering, and multimodal interaction.

Introduction

❖ What is MLLMs?



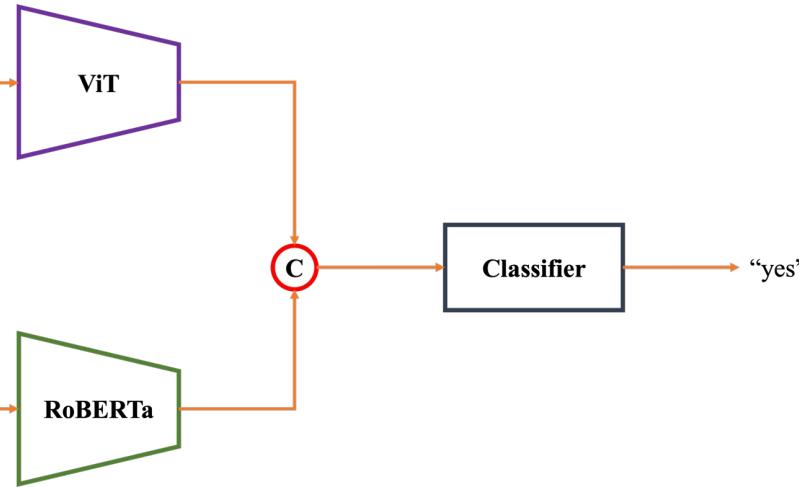
Introduction

❖ Vision Language Models?

Image:



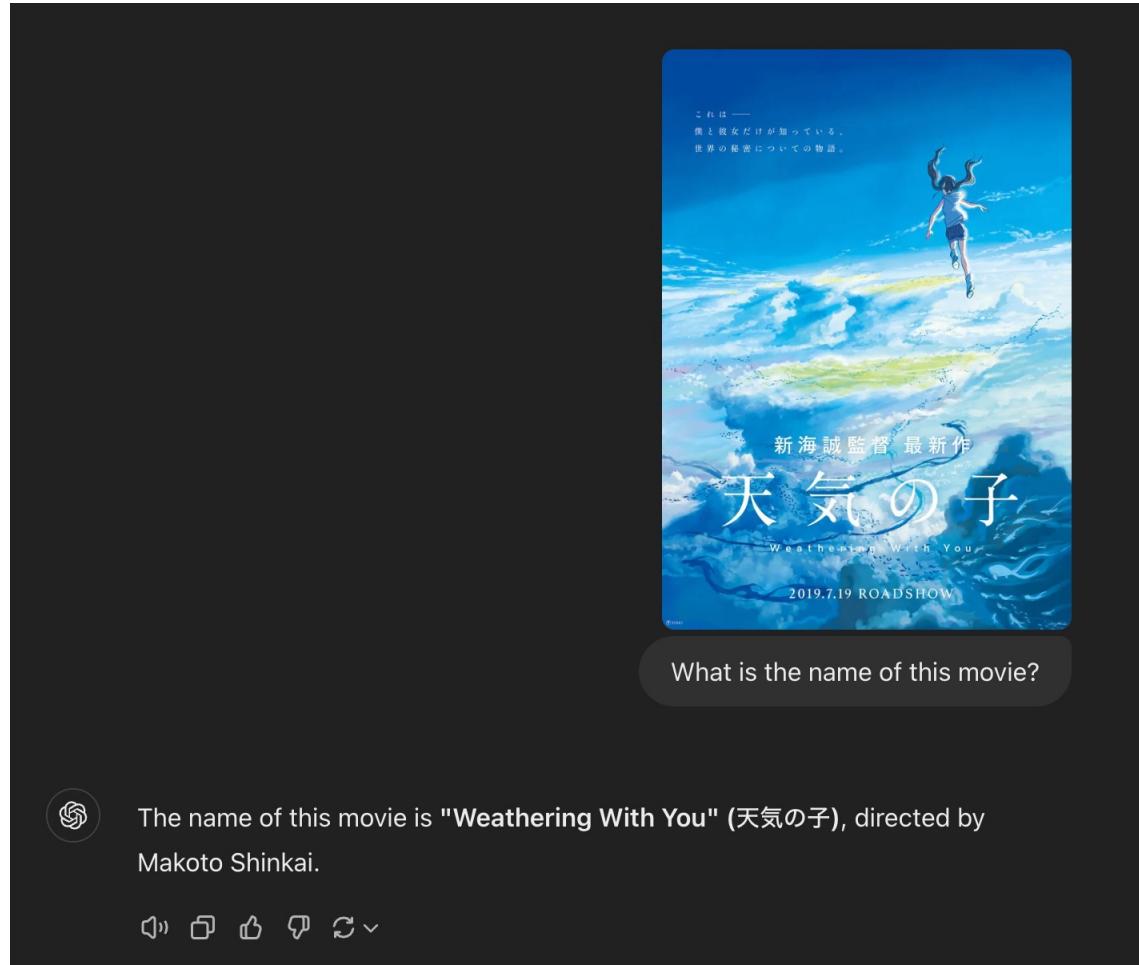
Question: Is this a high quality bottle of wine ?



Vision-Language Models (VLMs) are models that can process and integrate visual and textual information, enabling tasks like image captioning, visual question answering, and multimodal reasoning. In the case of VQA, it can also be considered as a VLM.

Introduction

❖ Large Vision Language Models



Large Vision-Language Models (LVLMs) are models that integrate large language models (LLMs) with pre-trained vision encoders, enabling them to process image inputs, understand various queries, and perform reasoning tasks.

Figure: ChatGPT, one of the most famous application that utilizing LVLMs.

Introduction

❖ Capabilities of LVLMs

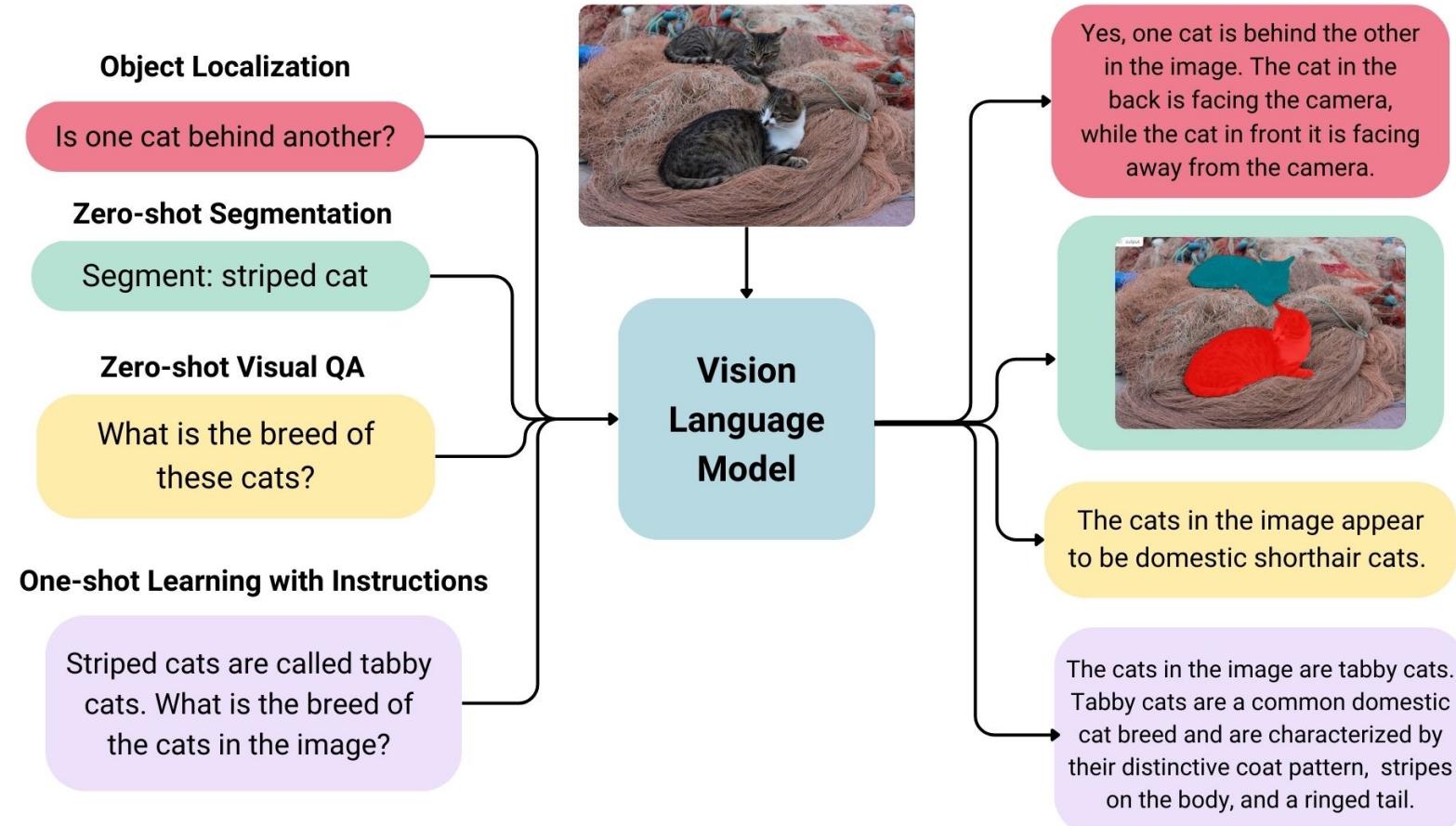


Figure: Illustration of what a Large Vision-Language Model (LVLM) can do nowadays.

Introduction

❖ LVLMs Applications

Prompt:

Localize each person in the image using bounding box. What is the image size of the input image?



GPT-4V:

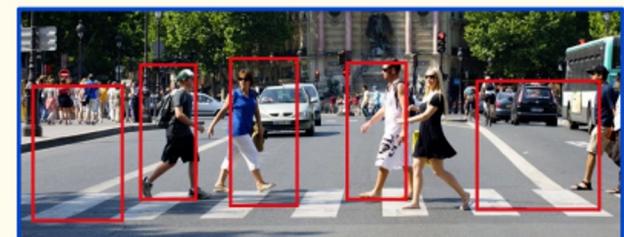
The image size of the input image is 1000 x 333 pixels.

Bounding boxes for each person:

- Person 1 (leftmost): $(x_1, y_1) = (22, 108), (x_2, y_2) = (170, 300)$
- Person 2: $(x_1, y_1) = (200, 80), (x_2, y_2) = (290, 270)$
- Person 3: $(x_1, y_1) = (350, 70), (x_2, y_2) = (460, 280)$
- Person 4: $(x_1, y_1) = (540, 80), (x_2, y_2) = (640, 270)$
- Person 5 (rightmost): $(x_1, y_1) = (760, 100), (x_2, y_2) = (920, 290)$

Note: These bounding box coordinates are approximate and may not be exact.

Visualization of Detected Bbox:



Object Localization

Prompt:

Localize each person in the image using bounding box. The image size is (width, height) = (998,786)



GPT-4V:

There is only one person in the image, so the bounding box for this person would be:

- Top left: (254, 118)
- Bottom right: (731, 693)

Explanation: The top left point of the bounding box is at $(x, y) = (254, 118)$, which is the top left corner of the person in the image. The bottom right point of the bounding box is at $(x, y) = (731, 693)$, which is the bottom right corner of the person in the image. This bounding box encompasses the entire person in the image.

Visualization of Detected Bbox:



Prompt:

Localize each car in the image using bounding box. What is the image size of the input image?



GPT-4V:

The image size of the input image is 500 x 281 pixels.

Bounding box for the black car:

- Top left corner: (63, 114)
- Bottom right corner: (196, 220)

Bounding box for the white car:

- Top left corner: (287, 96)
- Bottom right corner: (474, 220)

Visualization of Detected Bbox:



Introduction

❖ LVLMs Applications

Prompt:

Count the number of oranges in the image.

**GPT-4V:**

There are 3 oranges in the image.

Prompt:

Count the number of people in the image.

**GPT-4V:**

There are 3 people in the image.

Object Counting

Introduction

❖ LVMs Applications

Prompt:

Please read the text in this image and return the information in the following JSON format (note xxx is placeholder, if the information is not available in the image, put "N/A" instead).

```
{"Surname": xxx, "Given Name": xxx, "USCIS #": xxx, "Category": xxx, "Country of Birth": xxx, "Date of Birth": xxx, "SEX": xxx, "Card Expires": xxx, "Resident Since": xxx}
```



GPT-4V:

```
{"Surname": "STEVENS", "Given Name": "ICHIKA", "USCIS #": "000-000-000", "Category": "IR1", "Country of Birth": "Japan", "Date of Birth": "18 AUG 1988", "SEX": "F", "Card Expires": "07/04/2027", "Resident Since": "07/05/2017"}
```

OCR-Free eKYC

Introduction

❖ LVLMs Applications

Prompt:

Who is the person in the image and what is the person doing?

**GPT-4V:**

The person in the image is the current President of the United States, Joe Biden. He is standing at a podium, likely delivering a speech, at the G7 Summit in Hiroshima, Japan in 2023.

Prompt:

Who is the person in the image and what is the person doing?

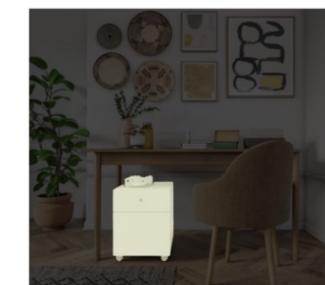
**GPT-4V:**

The person in the image is Jensen Huang, the CEO and co-founder of NVIDIA. He is holding and presenting a product from NVIDIA, likely a graphics processing unit (GPU).

Introduction

❖ LVLMs Applications

→ Visual Control
→ Language Control



English, Positive, Facutal

Chinese, Positive, Facutal

English, Positive, Facutal

English, Negative, Facutal

English, Positive, Facutal

English, Positive, Imagination

there is a chair that is sitting on a wooden floor

桌子前有一把椅子

there is a white cabinet with a basket on top of it

The bland white cabinet is topped with an unremarkable basket.

there is a plant that is sitting in a plot on the floor

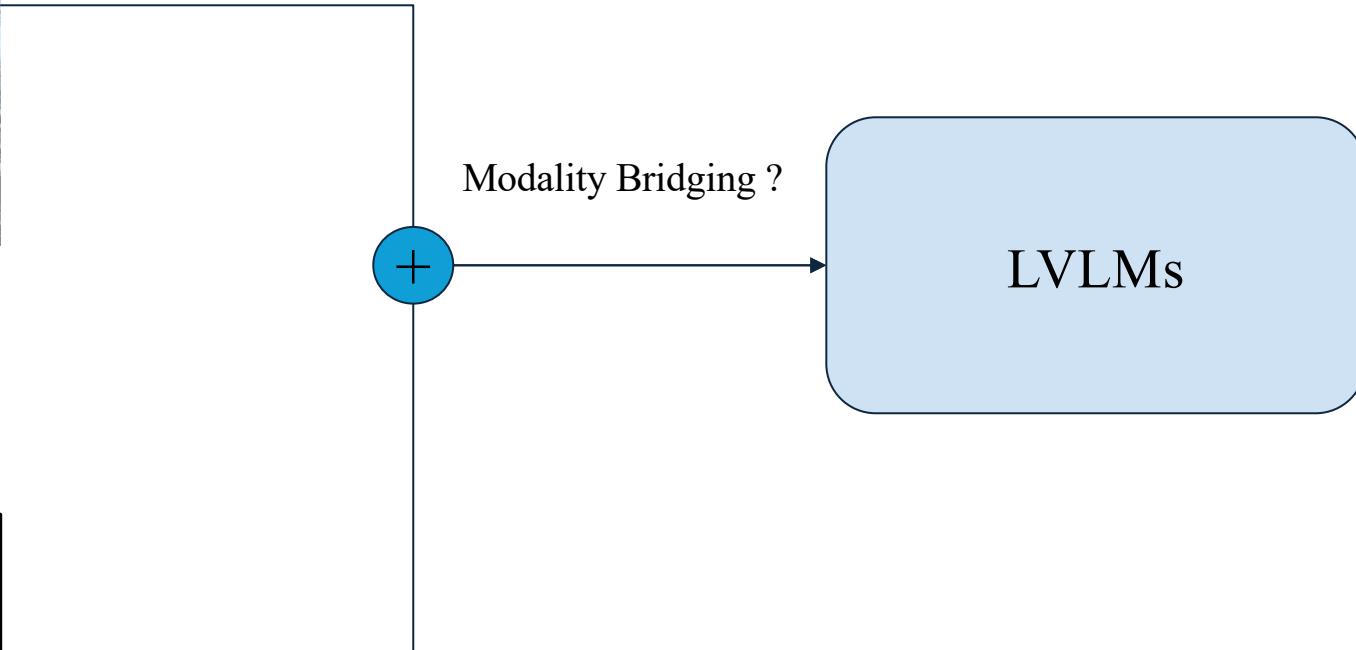
A lovely plant sits in a pot on the floor, adding a touch of nature to the room.

Introduction

❖ LVLMs Challenges: Modality Bridging

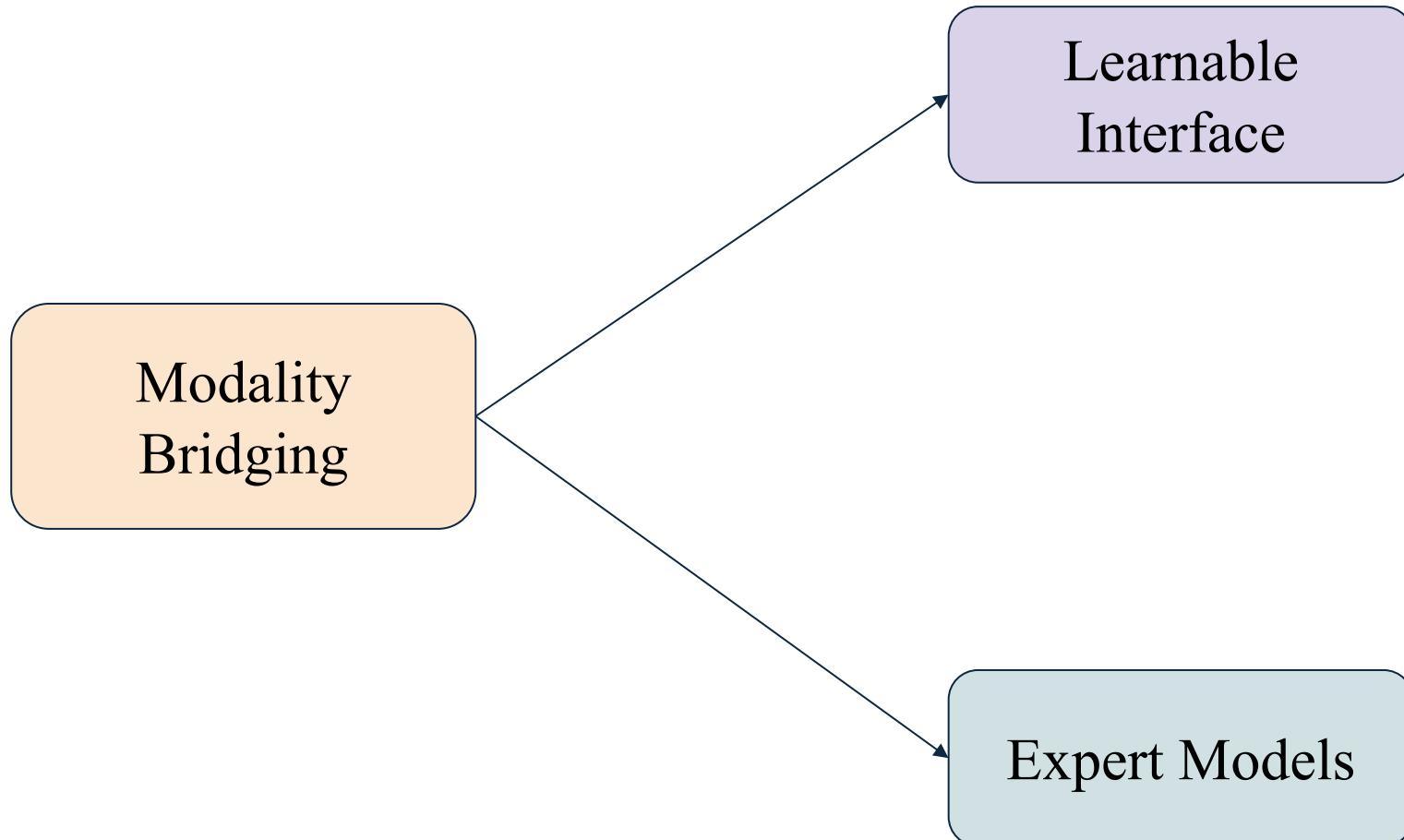


Instruction: Complete the
below task
Input: Describe the image



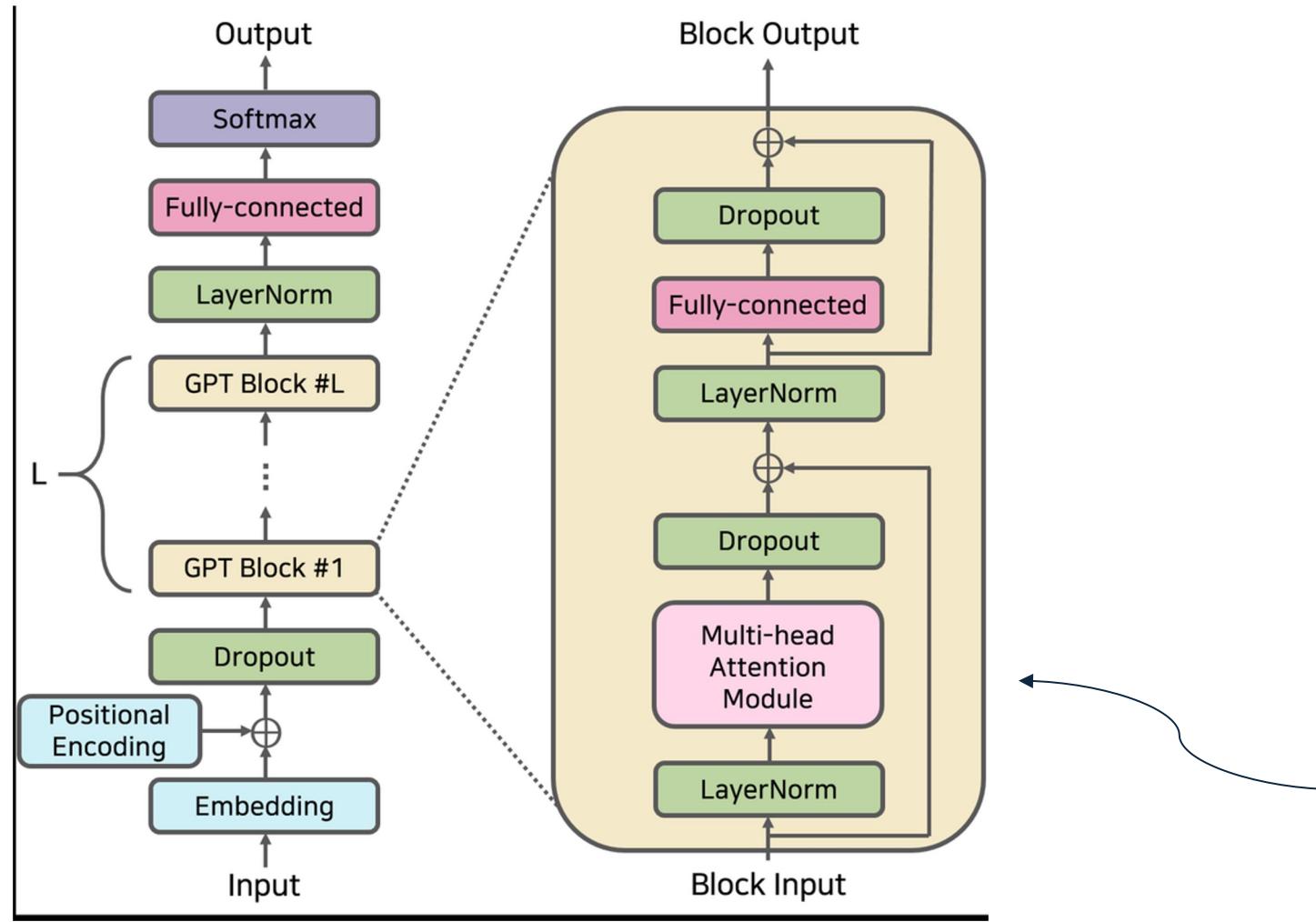
Introduction

❖ Modality Bridging Addresses



Introduction

❖ Modality Bridging: Learnable Interface



Freezed Visual
Encoder

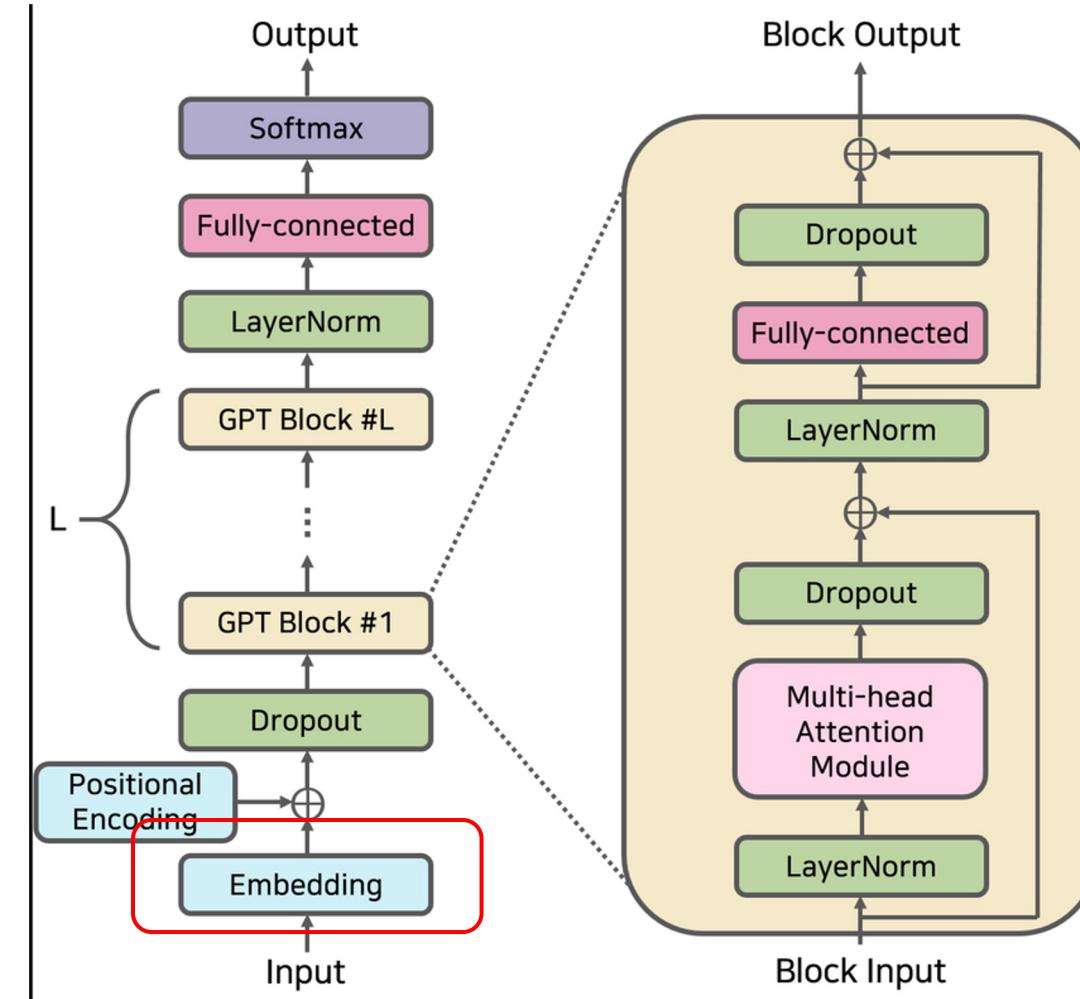
0.2	0.3	-0.6	0.5	0.2
0.5	0.8	0.2	0.5	0.9
-0.4	-0.4	0.6	0.2	0.7
0.1	-0.9	-0.4	-0.1	0.4

Introduction

❖ Modality Bridging: Learnable Interface

0.2	0.3	-0.6	0.5	0.2
0.5	0.8	0.2	0.5	0.9
-0.4	-0.4	0.6	0.2	0.7
0.1	-0.9	-0.4	-0.1	0.4

Image Features





Introduction

❖ Learnable Interface Example: LlaVA

Visual Instruction Tuning

Haotian Liu^{1*}, Chunyuan Li^{2*}, Qingyang Wu³, Yong Jae Lee¹

¹University of Wisconsin–Madison ²Microsoft Research ³Columbia University

<https://llava-vl.github.io>

Abstract

Instruction tuning large language models (LLMs) using machine-generated instruction-following data has been shown to improve zero-shot capabilities on new tasks, but the idea is less explored in the multimodal field. We present the first attempt to use language-only GPT-4 to generate multimodal language-image instruction-following data. By instruction tuning on such generated data, we introduce LLaVA: Large Language and Vision Assistant, an end-to-end trained large multimodal model that connects a vision encoder and an LLM for general-purpose visual and language understanding. To facilitate future research on visual instruction following, we construct two evaluation benchmarks with diverse and challenging application-oriented tasks. Our experiments show that LLaVA demonstrates impressive multimodal chat abilities, sometimes exhibiting the behaviors of multimodal GPT-4 on unseen images/instructions, and yields a 85.1% relative score compared with GPT-4 on a synthetic multimodal instruction-following dataset. When fine-tuned on Science QA, the synergy of LLaVA and GPT-4 achieves a new state-of-the-art accuracy of 92.53%. We make GPT-4 generated visual instruction tuning data, our model, and code publicly available.

Figure: Paper that introduces LLaVA model – an LVLM.

Introduction

❖ Learnable Interface Example: LlaVA Architecture

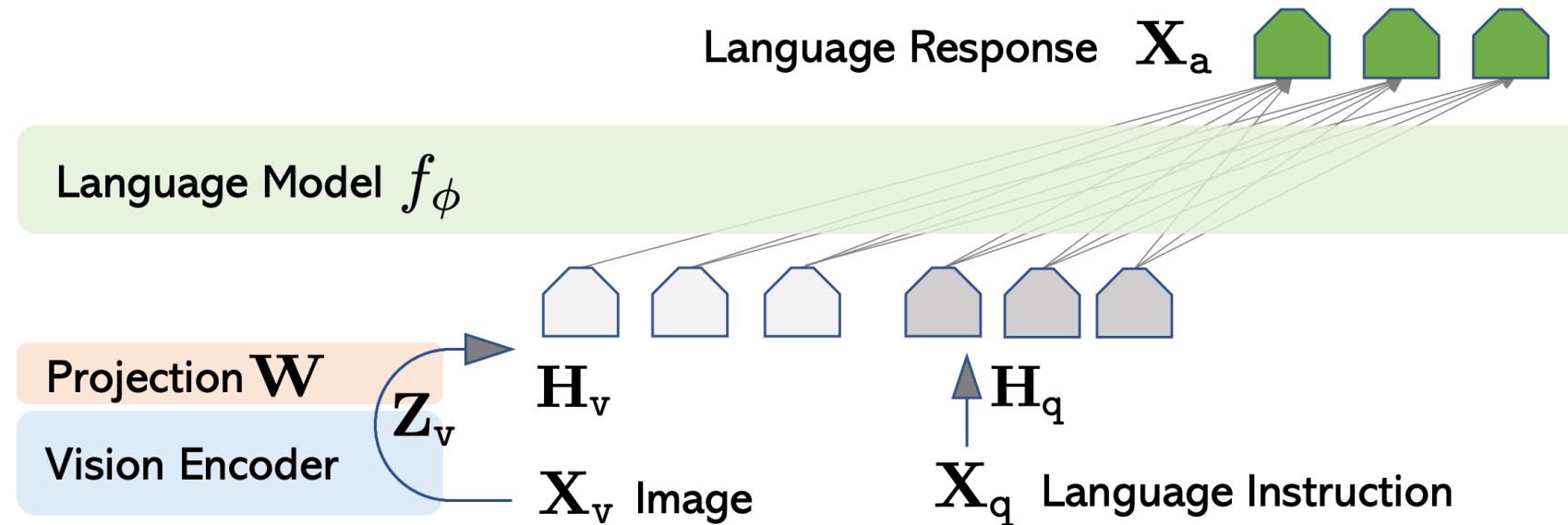


Figure: Illustration of LLaVA network architecture.

Introduction

❖ Learnable Interface Example: LlaVA Architecture

Output:

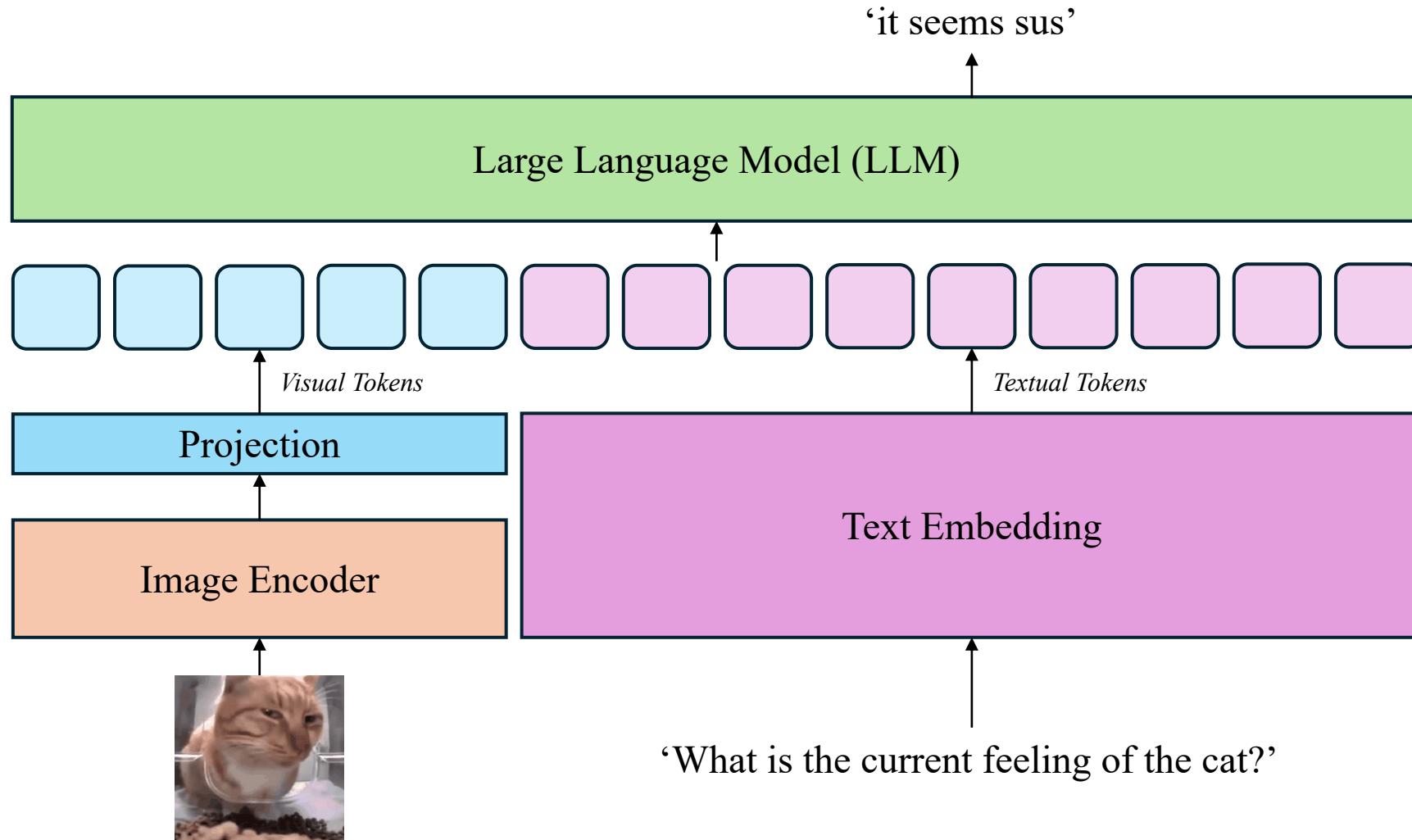


Figure: A typical architecture of LVLMs.

Introduction

❖ Learnable Interface Example: LlaVA

Context type 1: Captions

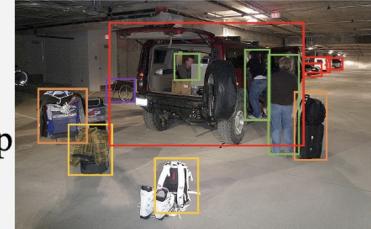
A group of people standing outside of a black vehicle with various luggage.

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], backpack: [0.384, 0.696, 0.485, 0.914], suitcase: ...<omitted>

Response type 1: conversation

Question: What type of vehicle is featured in the image?

Answer: The image features a black sport utility vehicle (SUV) ...<omitted>

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. ...<omitted>

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 ...<omitted>

Introduction

❖ Learnable Interface Example: LlaVA

Visual input example, Extreme Ironing:



Source: <https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg>

User
LLaVA

What is unusual about this image?

The unusual aspect of this image is a man ironing clothes on the back of a minivan or van. This is not a typical place to perform this activity, as one would usually iron clothes in a more stationary and safe location, such as a home, using a regular ironing board. The scene depicted in the image is peculiar as it involves a makeshift ironing setup on a vehicle, which can be both unsafe and unconventional. Additionally, it is not clear how the man is able to maintain balance and stability while ironing clothes in such an unstable environment.

[Start a new conversation, and clear the history]

User
LLaVA

What's happening in the scene?

The image depicts a man standing on top of a yellow SUV in a busy city street. He is holding a portable ladder, seemingly ironing clothes while standing on the vehicle. Around the scene, there are other cars, a traffic light, and a fire hydrant. Additionally, there are two other people in the vicinity, one close to the man on the car and another further away. The man on the ladder appears to be attracting attention due to his unconventional choice of ironing his clothes on top of a moving car. The city street around him is bustling with activity, adding to the unique nature of the scene.

Visual input example, Chicken Nugget Map:

Sometimes I just look at pictures of the earth from space and I marvel at how beautiful it all is.



User
LLaVA

Can you explain this meme in detail?

The meme in the image is a creative and humorous take on food, with a focus on chicken nuggets, as the center of the universe. The meme begins with the phrase "Sometimes I just look at pictures of the Earth from space and I marvel at how beautiful it all is..." The main image features a tray of cooked chicken nuggets, with the chicken pieces positioned to resemble the continents and islands on a world map. The meme playfully suggests that the chicken nuggets represent the Earth, and the various locations depicted in the photo are actually chicken nugget versions of different places. The punchline of the meme appears in smaller text at the bottom, reading "I mean, it's not the real Earth, but how beautiful it is all is." This lighthearted and entertaining meme combines food and humor, using the image of chicken nuggets to create a funny and imaginative representation of the world.

User
GPT-4 [36]

Can you explain this meme?

This meme is a joke that combines two unrelated things: pictures of the earth from space and chicken nuggets. The text of the meme suggests that the image below is a beautiful picture of the earth from space. However, the image is actually of chicken nuggets arranged to vaguely resemble a map of the world. The humor in this meme comes from the unexpected juxtaposition of the text and the image. The text sets up an expectation of a majestic image of the earth, but the image is actually something mundane and silly.



Introduction

❖ LlaVA Example

LLaVA: Large Language and Vision Assistant

[Project Page] [Code] [Model] | [LLaVA] [LLaVA-v1.5]

llava-v1.5-7b

Image

Drop Image Here
- OR -
Click to Upload

LLaVA Chatbot

which language does the signboard in this image use?

The signboard in this image uses Vietnamese.

Examples

	What is unusual about this image?
	What are the things I should be cautious about when I visit here?

Enter text and press ENTER

Send

Parameters

Upvote

Downvote

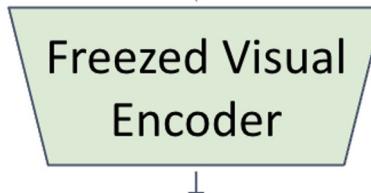
Flag

Regenerate

Clear

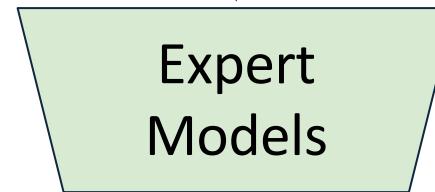
Introduction

❖ Modality Bridging: Expert Models



0.2	0.3	-0.6	0.5	0.2
0.5	0.8	0.2	0.5	0.9
-0.4	-0.4	0.6	0.2	0.7
0.1	-0.9	-0.4	-0.1	0.4

Break the modality gap by convert visual content to textual content



CAPTION/DESCRIPTION of the image

Introduction

❖ Modality Bridging: Expert Models



The image captures a tranquil scene set within a bamboo forest during winter. The path, blanketed in fresh snow, leads through towering rows of bamboo stalks that stretch upwards, their green hues almost obscured by the white frost clinging to them. Snowflakes continue to gently fall, adding to the wintry veil that enshrouds the scene. In the distance, a solitary figure walks along the path, a small, contrasting element of human presence in the midst of nature's quiet splendor. The overcast sky suggests the quiet chill of a snowy day, with the forest's natural symmetry providing a sense of peace and solitude.

Introduction

❖ Modality Bridging: Expert Models



Expert
Models

CAPTION/DESCRIPTION of
the image.



You

Instruction: Answer the input question based on the context information

Context: The image captures a tranquil scene set within a bamboo forest during winter. The path, blanketed in fresh snow, leads through towering rows of bamboo stalks that stretch upwards, their green hues almost obscured by the white frost clinging to them. Snowflakes continue to gently fall, adding to the wintry veil that enshrouds the scene. In the distance, a solitary figure walks along the path, a small, contrasting element of human presence in the midst of nature's quiet splendor. The overcast sky suggests the quiet chill of a snowy day, with the forest's natural symmetry providing a sense of peace and solitude.

Input: What is the weather like in the picture?

Output:



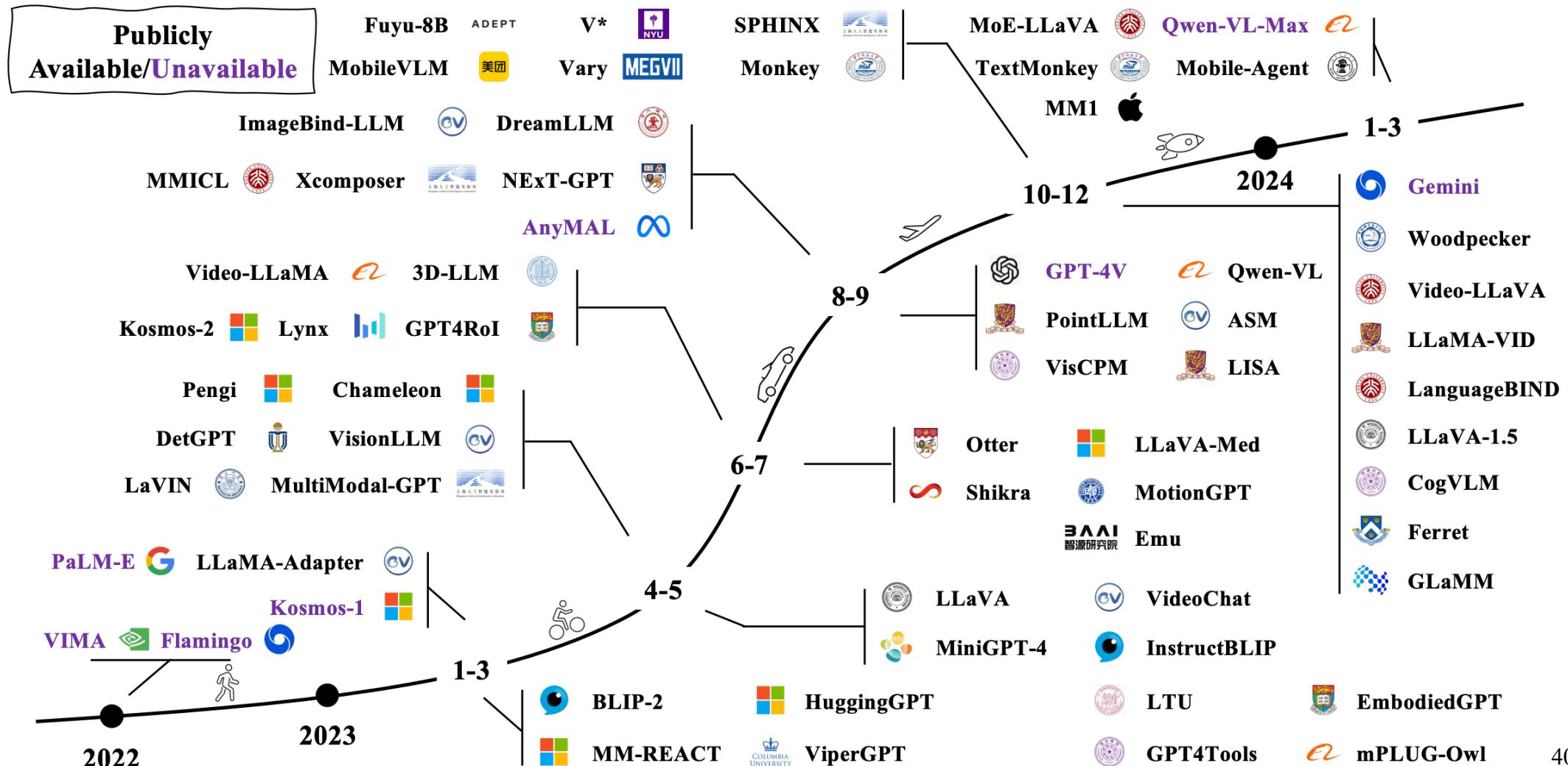
ChatGPT

The weather in the picture is wintry and snowy. The scene is set within a bamboo forest during winter, with fresh snow blanketing the path and clinging to the bamboo stalks. Snowflakes continue to gently fall, adding to the tranquil, wintry atmosphere. The sky is overcast, which conveys the chill of a snowy day.



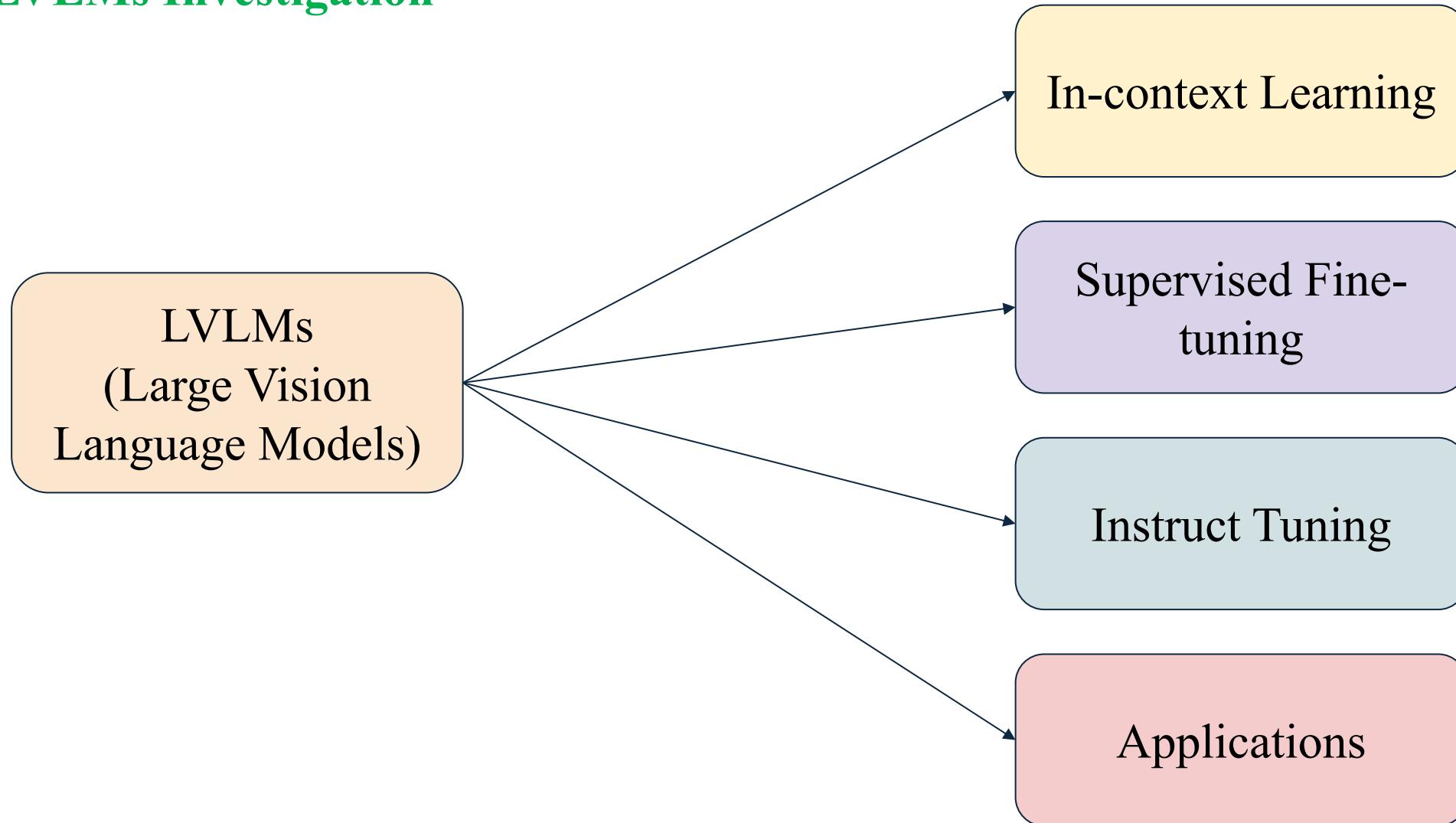
Introduction

❖ LVLMs Development Timeline (Up to Q1-2024)



Introduction

❖ LVLMs Investigation

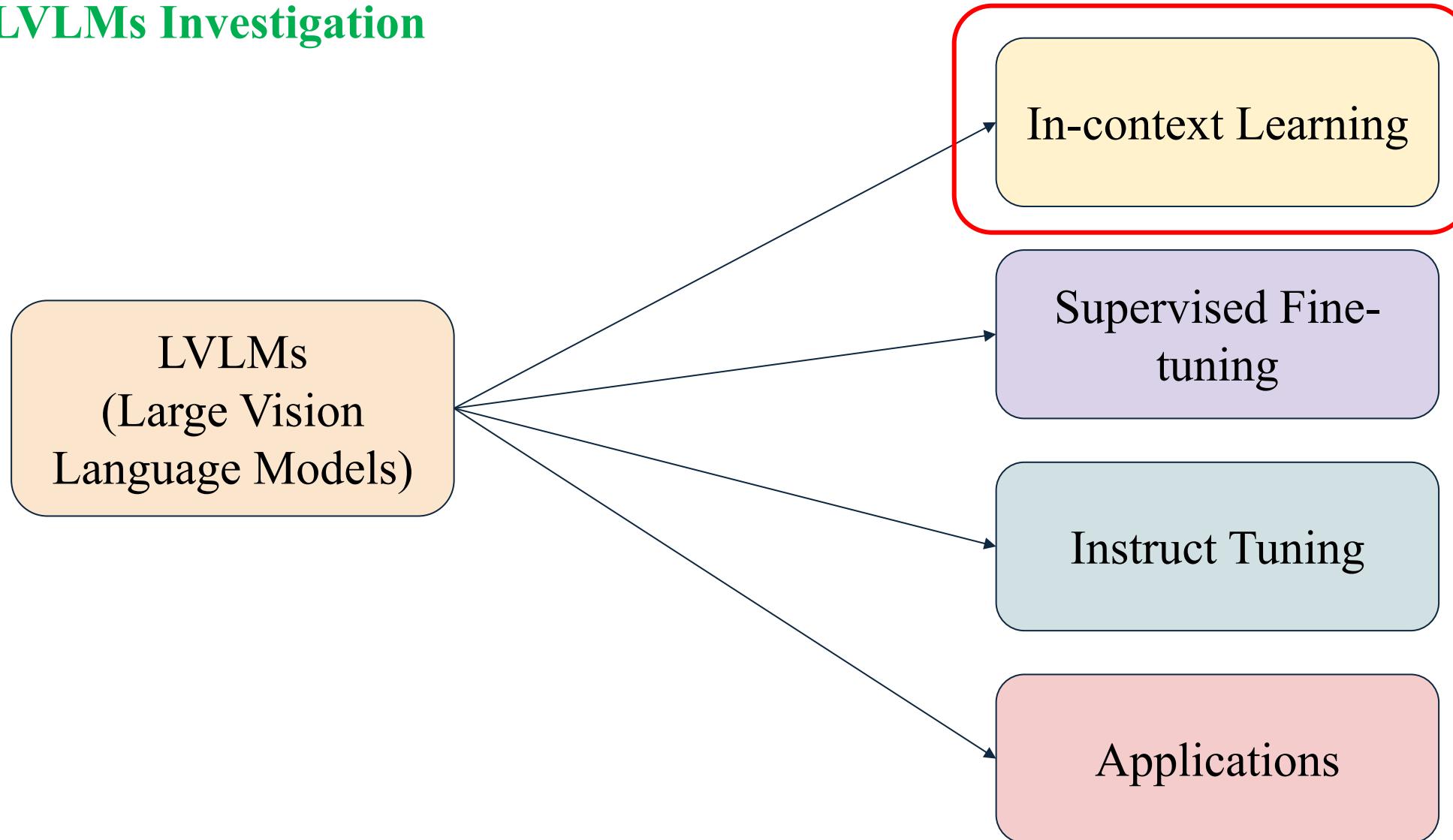




In-Context Learning with LVLMs

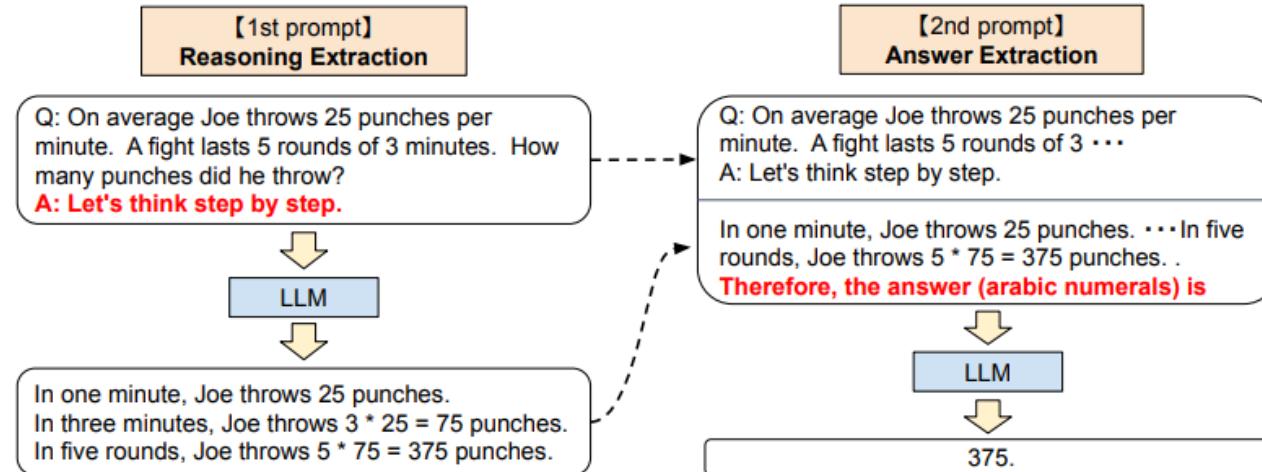
In-Context Learning with LVLMs

❖ LVLMs Investigation



In-Context Learning with LVLMs

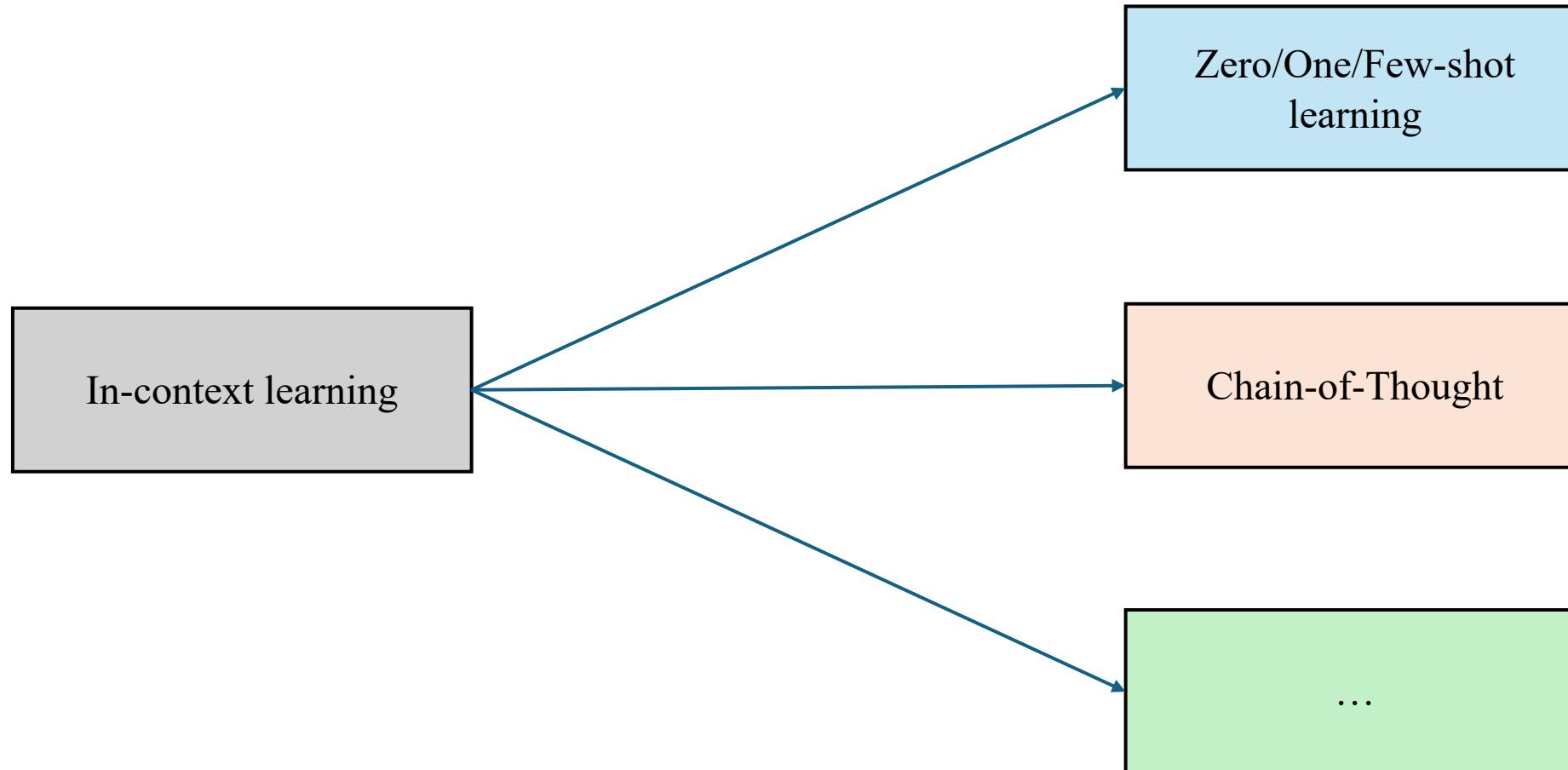
❖ Introduction



In-context learning (ICL) in LLMs is a technique where task demonstrations are integrated into the prompt in a natural language format. This approach allows pre-trained LLMs to address new tasks without fine-tuning the model.

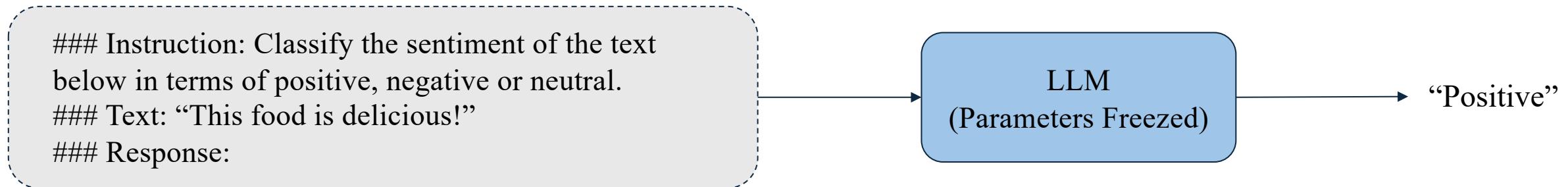
In-Context Learning with LVLMs

❖ In-context Learning types



In-Context Learning with LVLMs

❖ Introduction



Zero-shot learning: Prompting LLMs to do a task without any examples.

In-Context Learning with LVLMs

❖ Introduction

Instruction: Classify the sentiment of the text below in terms of positive, negative or neutral.

Text: “This food is delicious!”

Response: “Positive”

Text: “It’s raining outside. I bet we cannot go swimming today.”

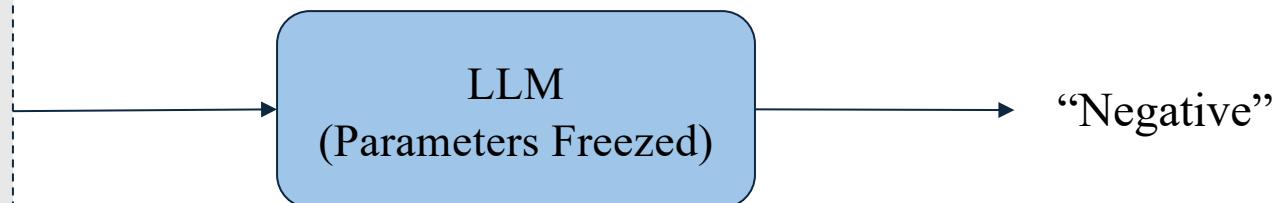
Response: “Negative”

Text: “Despite the bad service, this restaurant serves very tasty dishes.”

Response: “Neutral”

Text: “The new employee is getting so mean, that is not a proper manner as a beginner.”

Response:



Few-shot learning: Prompting LLMs to do a task with more than one example.

In-Context Learning with LVLMs

❖ ICL in LVLMs

<BOS> Below are some examples and an instruction that describes a task. Write a response that appropriately completes the request

Instruction: {instruction}

Image: <image>

Response: {response}

Image: <image>

Response: {response}

Image: <image>

Response: <EOS>

- ❖ **{instruction}**: A textual description of the task.
- ❖ **<image>**: A place-holder to represent an image.
- ❖ **{response}**: A response (answer) to the task related to **{instruction}** and **<image>**.
- ❖ **<BOS>**: The start input token.
- ❖ **<EOS>**: The end input token.

In-Context Learning with LVLMs

❖ CoT in LVLMs

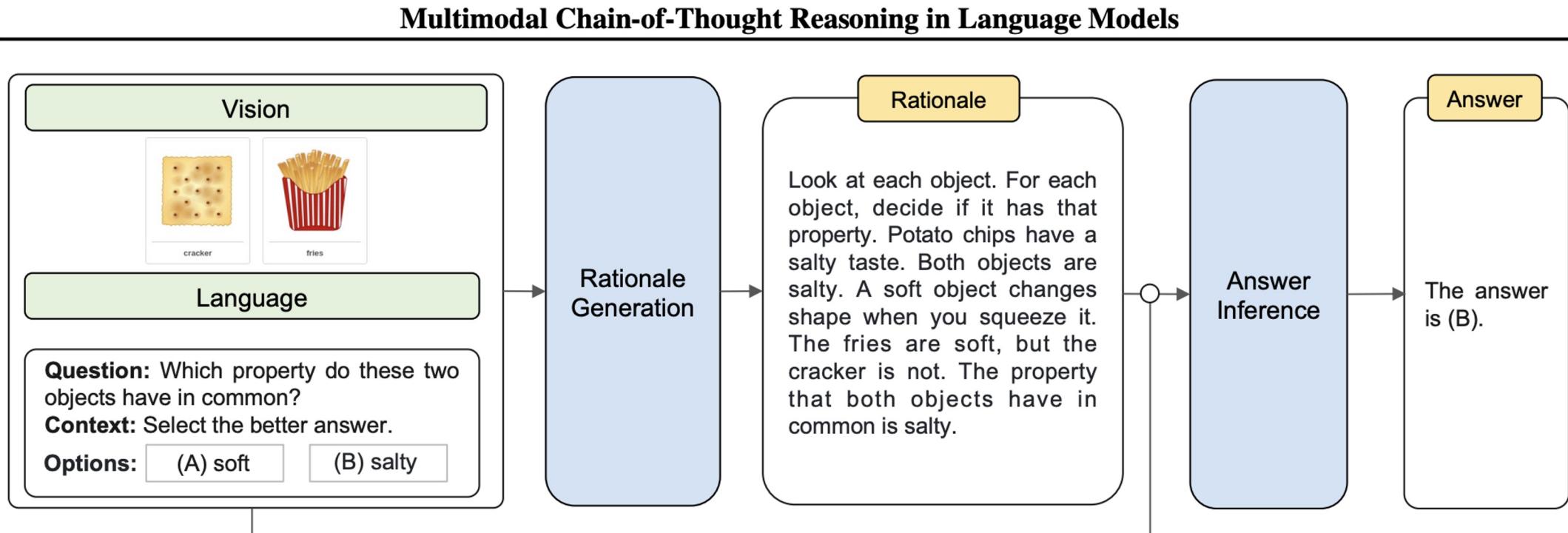


Figure 4. Overview of our Multimodal-CoT framework. Multimodal-CoT consists of two stages: (i) rationale generation and (ii) answer inference. Both stages share the same model architecture but differ in the input and output. In the first stage, we feed the model with language and vision inputs to generate rationales. In the second stage, we append the original language input with the rationale generated from the first stage. Then, we feed the updated language input with the original vision input to the model to infer the answer.

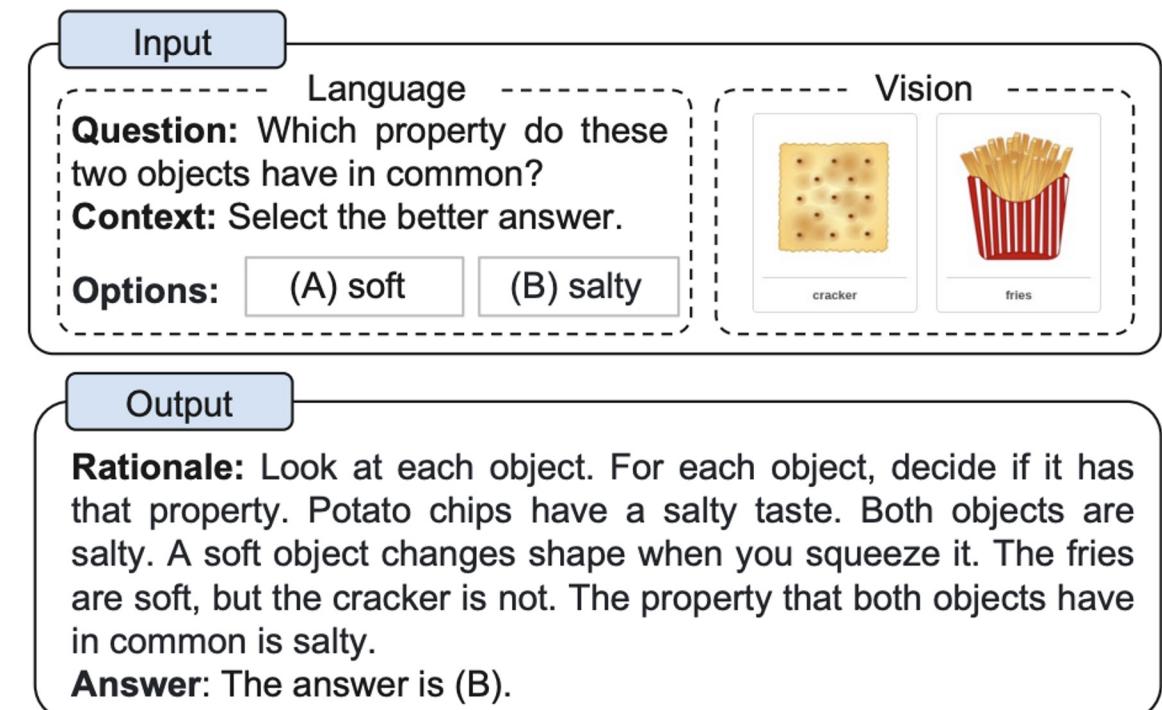
❖ CoT in LVLMs

Algorithm 1 Multimodal-CoT

Input: Language input X_{language}^1 , vision input X_{vision}

Output: Generated rationale R , inferred answer A

- 1: Construct the input $X = \{X_{\text{language}}, X_{\text{vision}}\}$
- 2: Generate rationale $R = F(X)$ using the model $F(\cdot)$
- 3: Append the rationale R to the original language input $X_{\text{language}}^2 = X_{\text{language}}^1 \circ R$.
- 4: Construct new input $X' = \{X_{\text{language}}^2, X_{\text{vision}}\}$
- 5: Infer the answer A by conditioning on the new input, $A = F(X')$.
- 6: **procedure** $F(X)$
- 7: Encode the language and vision inputs H_{language} and H_{vision} , respectively
- 8: Build the interaction between language and vision features by attention $H_{\text{vision}}^{\text{attn}}$
- 9: Fuse H_{language} and $H_{\text{vision}}^{\text{attn}}$ by a gated fusion mechanism to have H_{fuse}
- 10: Feed H_{fuse} to the decoder to obtain the target prediction Y
- 11: **return** Y
- 12: **end procedure**



In-Context Learning with LVLMs

❖ CoT in LVLMs



Input Image

```
37
38     question = "Between two objects in the image, which one is stronger?"
39     context = "Select the better answer."
40     choice = "(A) Lion (B) Human"
41     img_path = "./images/Human_vs_Lion.jpg"
42
43
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

(thuan_torch) server-ailab-12gb@serverailab12gb-System-Product-Name:/mnt/HDD/ThangDuongTeam/thuann
● /mm-cot\$ python predict.py
Solution: Look at the objects in the image.nFor each object, decide if it is stronger.nThe lion is
stronger than the other object.nThe other object is not strong. But the lion is stronger.
Answer: The answer is (A).

In-Context Learning with LVLMs

❖ CoT in LVLMs



Input Image

```
37
38 question = "Between two liquids in the image, which one is better for health?"
39 context = "Select the better answer."
40 choice = "(A) Left (B) Right"
41 img_path = "./images/water_vs_coca.jpg"
42
43
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(thuan_torch) server-ailab-12gb@serverailab12gb-System-Product-Name:/mnt/HDD/ThangDuongTeam/thuann
● /mm-cot$ python predict.py
Solution: Look at the picture of the two liquids.nThe two liquids are mostly black, but they are no
t actually health.nSo, the right liquid is better for health.
Answer: The answer is (B).
```

In-Context Learning with LVLMs

❖ CoT in LVLMs



Input Image

```
37
38 question = "Between two liquids in the image, which one is better for health?"
39 context = "Select the better answer."
40 choice = "(A) Water (B) Coca"
41 img_path = "./images/water_vs_coca.jpg"
42
43
```

In-Context Learning with LVLMs

❖ CoT in LVLMs



Input Image

```
38 question = "Between two foods on the table, which one is better for health?"
39 context = "Select the better answer."
40 choice = "(A) Left (B) Right"
41 img_path = "./images/food.png"
42
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
(thuan_torch) server-ailab-12gb@serverailab12gb-System-Product-Name:/mnt/HDD/ThangDuongTeam/thuannd/mm-cot$ python predict.py
Solution: Look at the picture of the table. It is shown that all food is unhealthy. But a diet of unhealthy food does not have all the health benefits of a healthy food. So, it is better for health.
Answer: The answer is (B).
```

In-Context Learning with LVLMs

❖ CoT in LVLMs

$$\frac{x+3}{6} = \frac{2}{3}$$

```
37
38     question = "Solve for X"
39     context = "Select the better answer."
40     choice = "(A) -1 (B) 1"
41     img_path = "./images/math.png"
42
43
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

(thuan_torch) server-ailab-12gb@serverailab12gb-System-Product-Name:/mnt/HDD/ThangDuongTeam/thuannnd
● /mm-cot\$ python predict.py
Solution: The better answer is -1.nThe better answer is -1.nThere is 1 left before the answer choice. So, 1 must be a right answer.
Answer: The answer is (A).

Input Image

❖ Prompting with Qwen2-VL

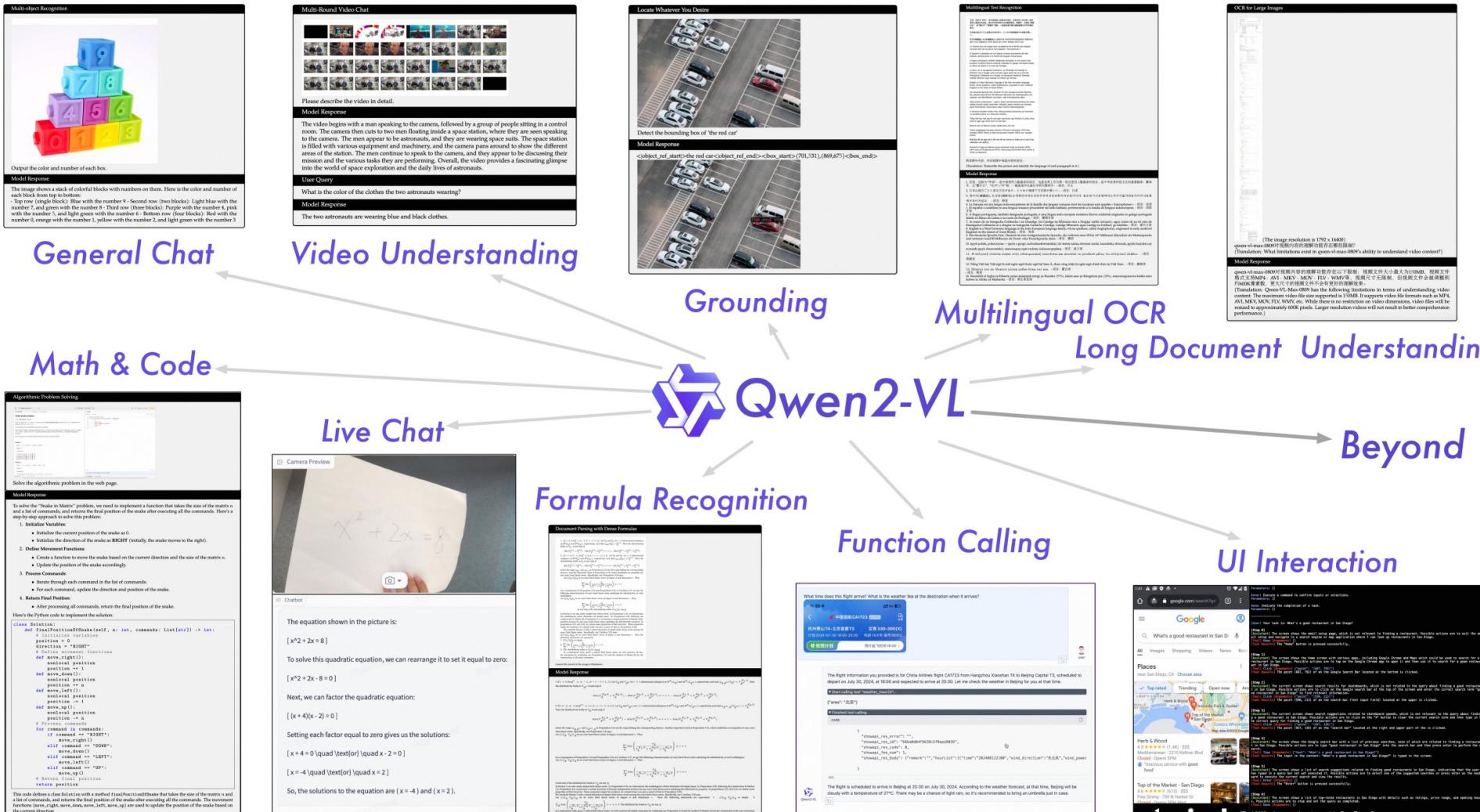


Qwen2-VL is a multimodal vision-language model developed by Alibaba Cloud, built upon the Qwen2 large language model (LLM) architecture. It integrates a vision encoder with the Qwen2 LLM to process and understand both visual and textual inputs, enabling tasks such as image captioning, visual question answering, and video comprehension.



In-Context Learning with LVLMs

❖ Qwen2-VL Capabilities



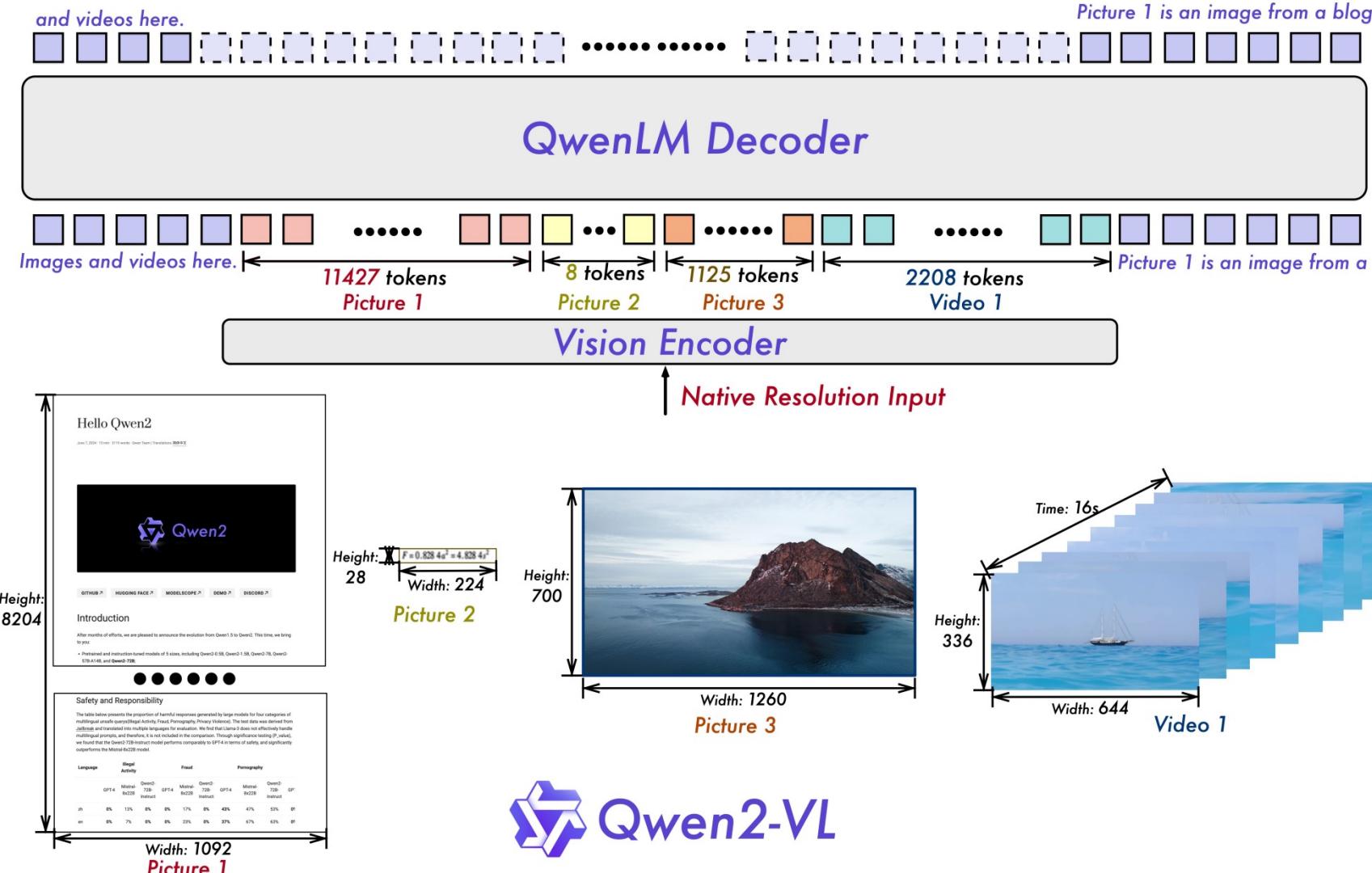
In-Context Learning with LVLMs

❖ Qwen2-VL Series

Model Name	Vision Encoder	LLM	Model Description
Qwen2-VL-2B	675M	1.5B	The most efficient model, designed to run on-device. It delivers adequate performance for most scenarios with limited resources.
Qwen2-VL-7B	675M	7.6B	The performance-optimized model in terms of cost, significantly upgraded for text recognition and video understanding capabilities. It delivers significant performance across a broad range of visual tasks.
Qwen2-VL-72B	675M	72B	The most capable model, further improvements in visual reasoning, instruction-following, decision-making, and agent capabilities. It delivers optimal performance on most complex tasks.

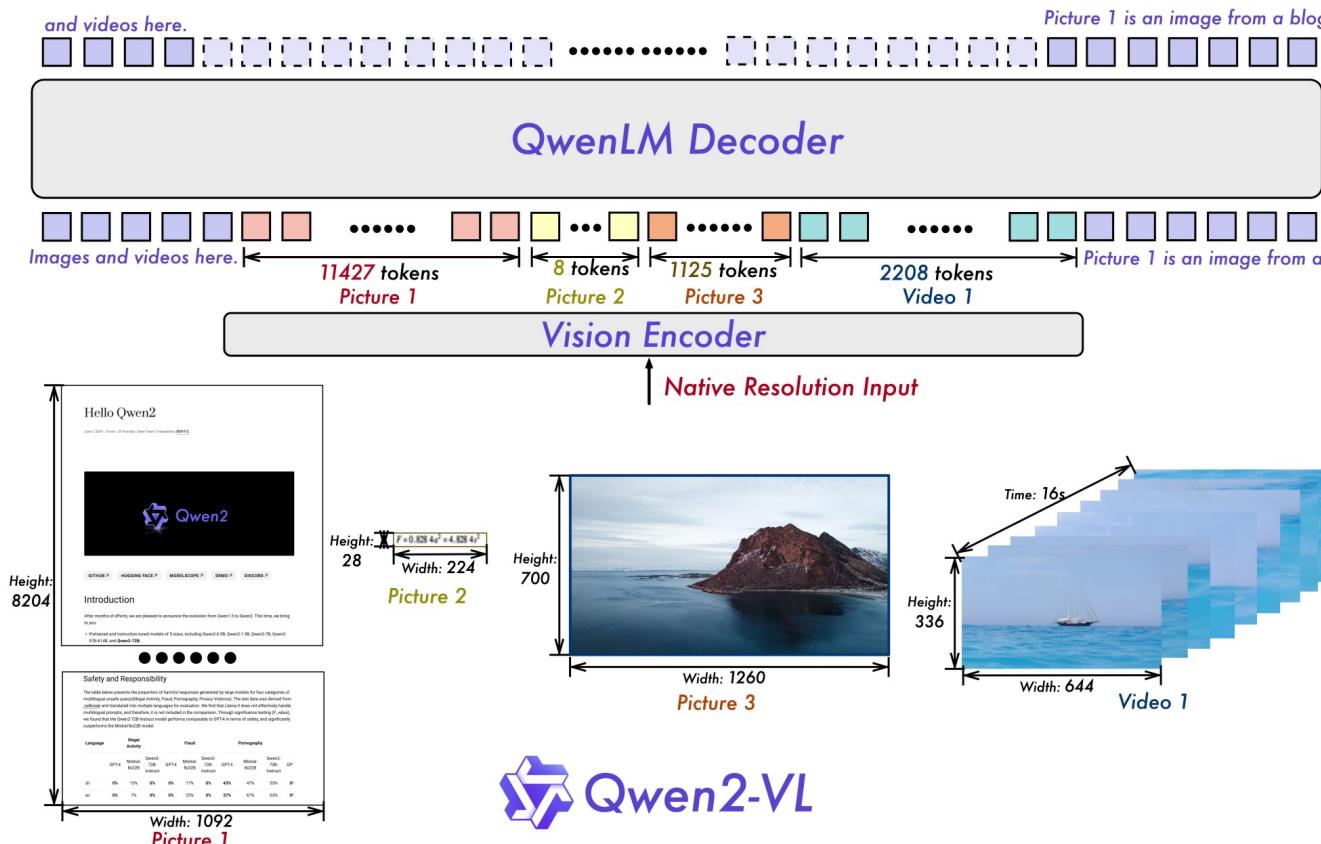
In-Context Learning with LVLMs

❖ Qwen2-VL Architecture



In-Context Learning with LVLMs

❖ Qwen2-VL Features



- ❖ **Naive Dynamic Resolution (NDR):** Allows the model to handle images of varying resolutions by dynamically adjusting the number of visual tokens, improving efficiency and visual representation.
- ❖ **Multimodal Rotary Position Embedding (M-RoPE):** Enhances the fusion of positional information across text, images, and videos, enabling better multimodal understanding.
- ❖ **Unified Vision-Language Architecture:** Adopts a single model structure to process both images and videos, supporting a wide range of vision-language tasks like captioning and question answering.

In-Context Learning with LVLMs

❖ Code prompting: Import necessary libraries

```
1 import os
2
3 os.environ["CUDA_VISIBLE_DEVICES"] = "2"
4
5 import torch
6 from datasets import load_dataset
7 from unsloth import FastVisionModel
8
9 from unsloth import is_bf16_supported
10 from transformers import TextStreamer
11 from PIL import Image
12
13 device = "cuda" if torch.cuda.is_available() else "cpu"
```





❖ Code prompting: Load Qwen2-VL model



```
1 model_name = "unslloth/Qwen2-VL-2B-Instruct"
2 model, tokenizer = FastVisionModel.from_pretrained(
3     model_name,
4     load_in_4bit = True,
5     use_gradient_checkpointing = "unslloth",
6 )
```



In-Context Learning with LVLMs

❖ Chat-style model: Conversation

```
[  
  {  
    "role": "system",  
    "content": "You are a helpful assistant that summarizes content clearly."  
  },  
  {  
    "role": "user",  
    "content": "Please summarize the following:\n\nMachine learning is a field of AI that allows computers to learn from data without being explicitly programmed."  
  },  
  {  
    "role": "assistant",  
    "content": "Machine learning helps computers learn from data automatically, without needing explicit instructions."  
  }]
```

Single-turn

```
[  
  {  
    "role": "system",  
    "content": "You are a helpful assistant that summarizes content clearly."  
  },  
  {  
    "role": "user",  
    "content": "Please summarize the following:\n\nMachine learning is a field of AI that allows computers to learn from data without being explicitly programmed."  
  },  
  {  
    "role": "assistant",  
    "content": "Machine learning helps computers learn from data automatically, without needing explicit instructions."  
  },  
  {  
    "role": "user",  
    "content": "Can you also summarize this?\n\nDeep learning is a subset of machine learning that uses neural networks with many layers."  
  },  
  {  
    "role": "assistant",  
    "content": "Deep learning is a type of machine learning that uses multi-layered neural networks to learn complex patterns from data."  
  }]
```

Multi-turn

In-Context Learning with LVLMs

❖ Code prompting: Create prepare input function

```
1 def prepare_chat_inputs(image, instruction, tokenizer,
2                         device="cuda", add_generation_prompt=True):
3     messages = [
4         {
5             "role": "user",
6             "content": [
7                 {"type": "image"},
8                 {"type": "text", "text": instruction}
9             ]
10        }
11    ]
12
13    input_text = tokenizer.apply_chat_template(
14        messages, add_generation_prompt=add_generation_prompt)
15
16    inputs = tokenizer(image, input_text,
17                      add_special_tokens=False, return_tensors="pt").to(device)
18
19    return inputs
```

```
messages = [
    {
        "role": "user",
        "content": [
            {"type": "image"},
            {"type": "text", "text": instruction}
        ]
    }
]
```

Conversation format



❖ Code prompting: Create inference function

```
1 def run_inference(model, inputs, tokenizer,max_new_tokens=128,
                     temperature=1.5, min_p=0.1):
2     FastVisionModel.for_inference(model)
3
4     text_streamer = TextStreamer(tokenizer, skip_prompt=True)
5     output_text = model.generate(
6         **inputs,
7         streamer=text_streamer,
8         max_new_tokens=max_new_tokens,
9         use_cache=True,
10        temperature=temperature,
11        min_p=min_p
12    )
13
14     return output_text
```



In-Context Learning with LVLMs

❖ Code prompting: Zero-shot prompting

```
1 test_image_path = "cat.jpg"
2 image = Image.open(test_image_path).convert("RGB")
3 instruction = "How many cats are in the image?"
4
5 inputs = prepare_chat_inputs(
6     image=image,
7     instruction=instruction,
8     tokenizer=tokenizer,
9     device=device)
10
11 output_text = run_inference(
12     model=model,
13     inputs=inputs,
14     tokenizer=tokenizer,
15     max_new_tokens=128,
16     temperature=1.5,
17     min_p=0.1)
```



Response from Qwen2-VL-2B-Instruct: There are two cats in the image.

In-Context Learning with LVLMs

❖ Code prompting: Few-shot prompting

• • •

```
1 test_image_path = "cat.jpg"
2 image = Image.open(test_image_path).convert("RGB")
3 instruction = """
4 Answer the question based on the image in the following format: Answer: <answer>
5 Examples:
6 - Input: Question: How many cats are in the image?
7 - Output: Answer: 2
8 Here is the input:
9 Question: What is the color of the background?
10 """
11 inputs = prepare_chat_inputs(
12     image=image,
13     instruction=instruction,
14     tokenizer=tokenizer,
15     device=device)
16 output_text = run_inference(
17     model=model,
18     inputs=inputs,
19     tokenizer=tokenizer,
20     max_new_tokens=128,
21     temperature=1.5,
22     min_p=0.1)
```



Response from Qwen2-VL-2B-Instruct: Answer:
Blue

In-Context Learning with LVLMs

❖ Code prompting: CoT prompting

• • •

```
1 test_image_path = "cat.jpg"
2 image = Image.open(test_image_path).convert("RGB")
3 instruction = """
4 Please think step by step and answer the following question based on the image:
    Which cat is taller?
5 """
6 inputs = prepare_chat_inputs(
7     image=image,
8     instruction=instruction,
9     tokenizer=tokenizer,
10    device=device)
11
12 output_text = run_inference(
13     model=model,
14     inputs=inputs,
15     tokenizer=tokenizer,
16     max_new_tokens=128,
17     temperature=2.0,
18     min_p=0.1)
```



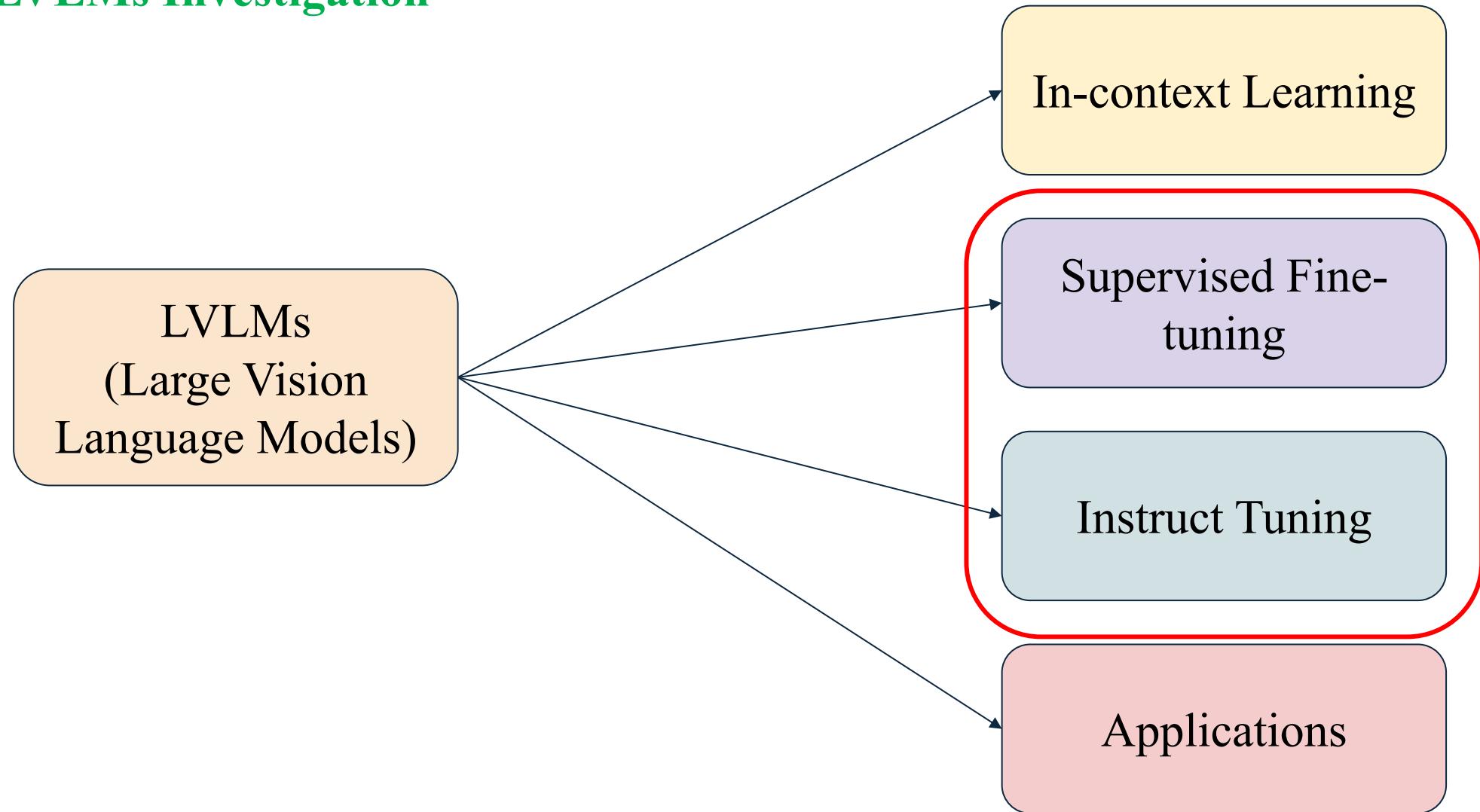
Response from Qwen2-VL-2B-Instruct: Based on the image, the cat on the right appears to be taller.



Training LVLMs

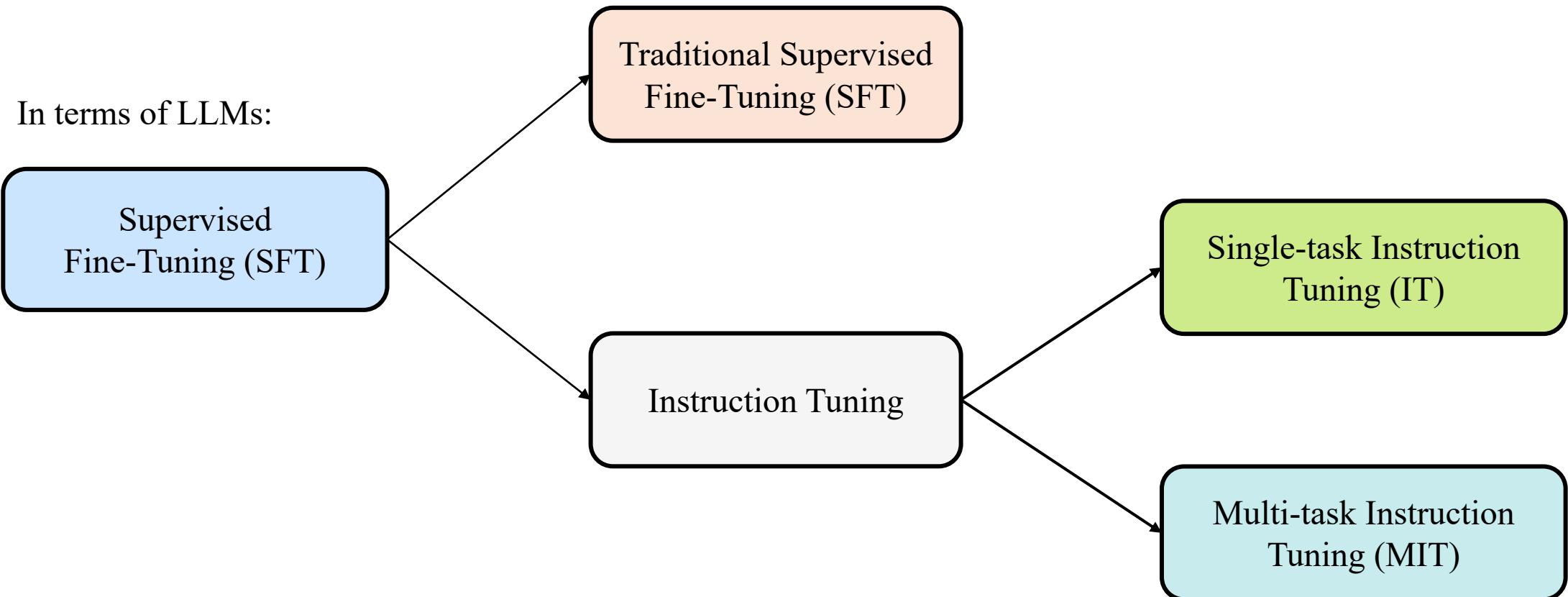
Training LVLMs

❖ LVLMs Investigation



Training LVLMs

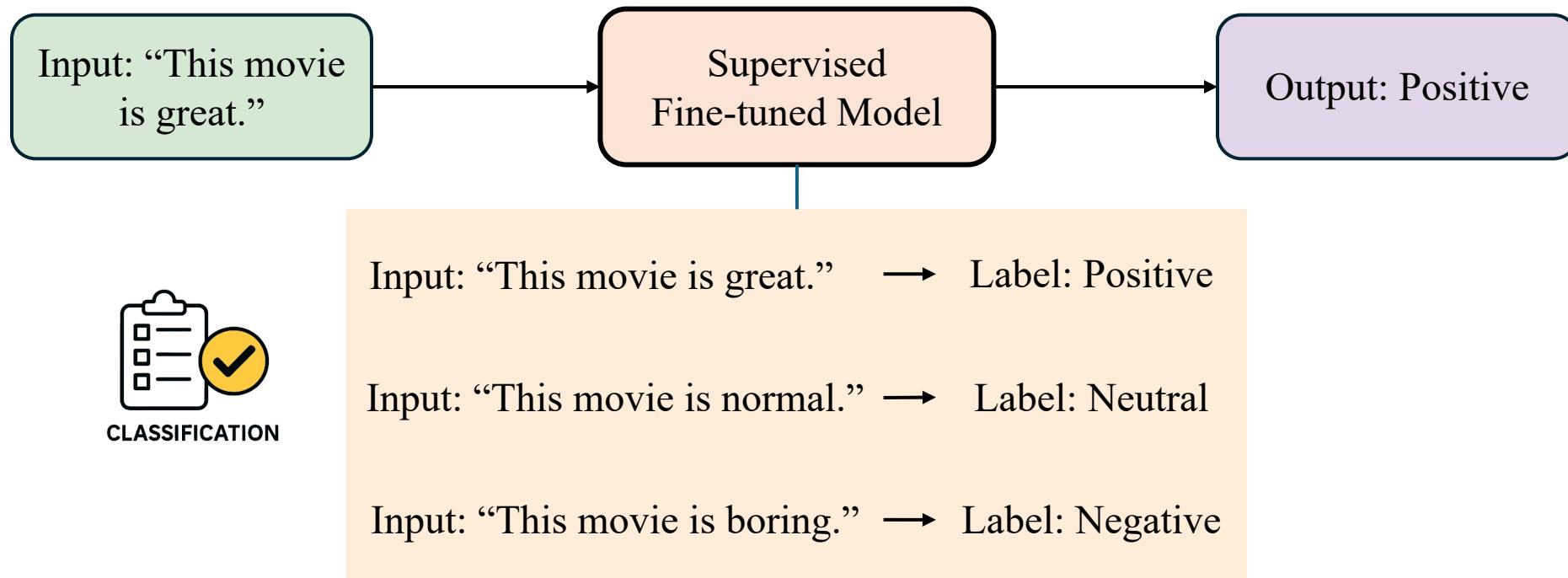
❖ LLM Fine-Tuning Strategies



Training LLMs

❖ Traditional Supervised Fine-Tuning (SFT)

Traditional Supervised Fine-Tuning (SFT) is the process of training a language model on task-specific input-output pairs, where each input is paired with a correct label or response. The goal is to optimize accuracy for known, structured tasks.

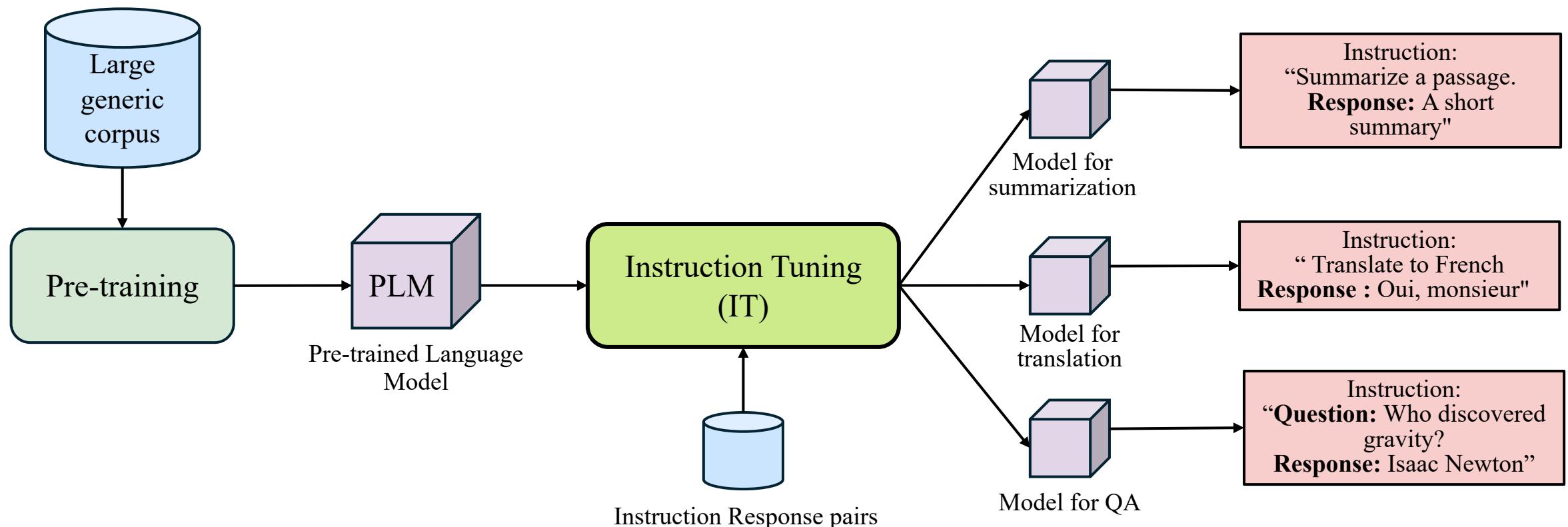


Training examples

Training LVLMs

❖ Instruction Tuning (IT)

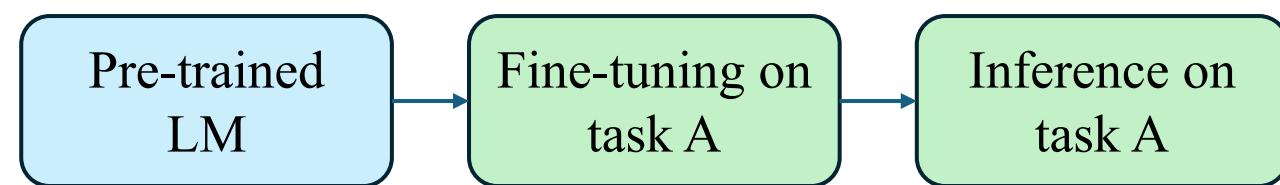
Instruction Tuning (IT) for short teaches a model to follow natural language instructions by training it on diverse prompts and corresponding responses. This helps the model better understand and execute user intent.



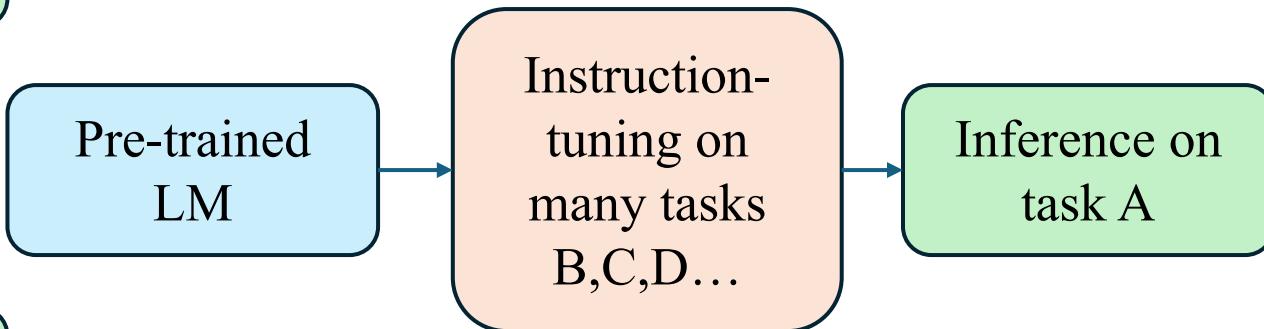
Training LVMs

❖ Instruction Tuning

A. Pre-train fine-tuning



C. Instruction tuning



B. Prompting (Zero-shot)



Training LVLMs

❖ Investigation on training LVLMs

I: Traditional
Supervised
Fine-Tuning (SFT)



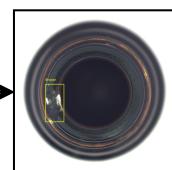
Có bao nhiêu người trong quán DodamDodam ?

MiniGPT

Có một người.



MiniGPT



<p>broken-defect</p>
<{<167><411><107><218>}>

II: Instruction
Tuning (IT)



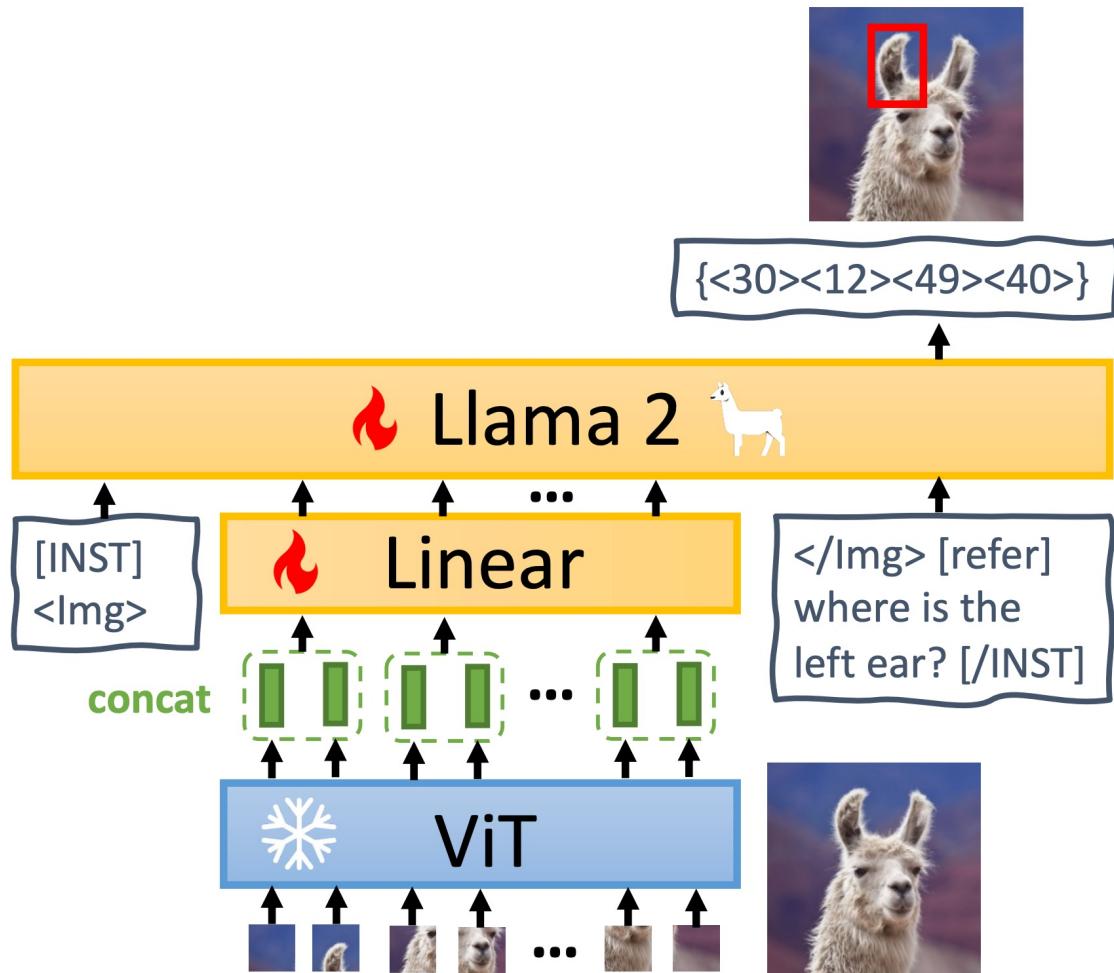
 Qwen2-VL



Summarization: Hóa đơn bán hàng của Vincommerce, ngày 14/08/2020, 21:37, mã hóa đơn H:00073344, NVBH:09021213...

Training LVLMs

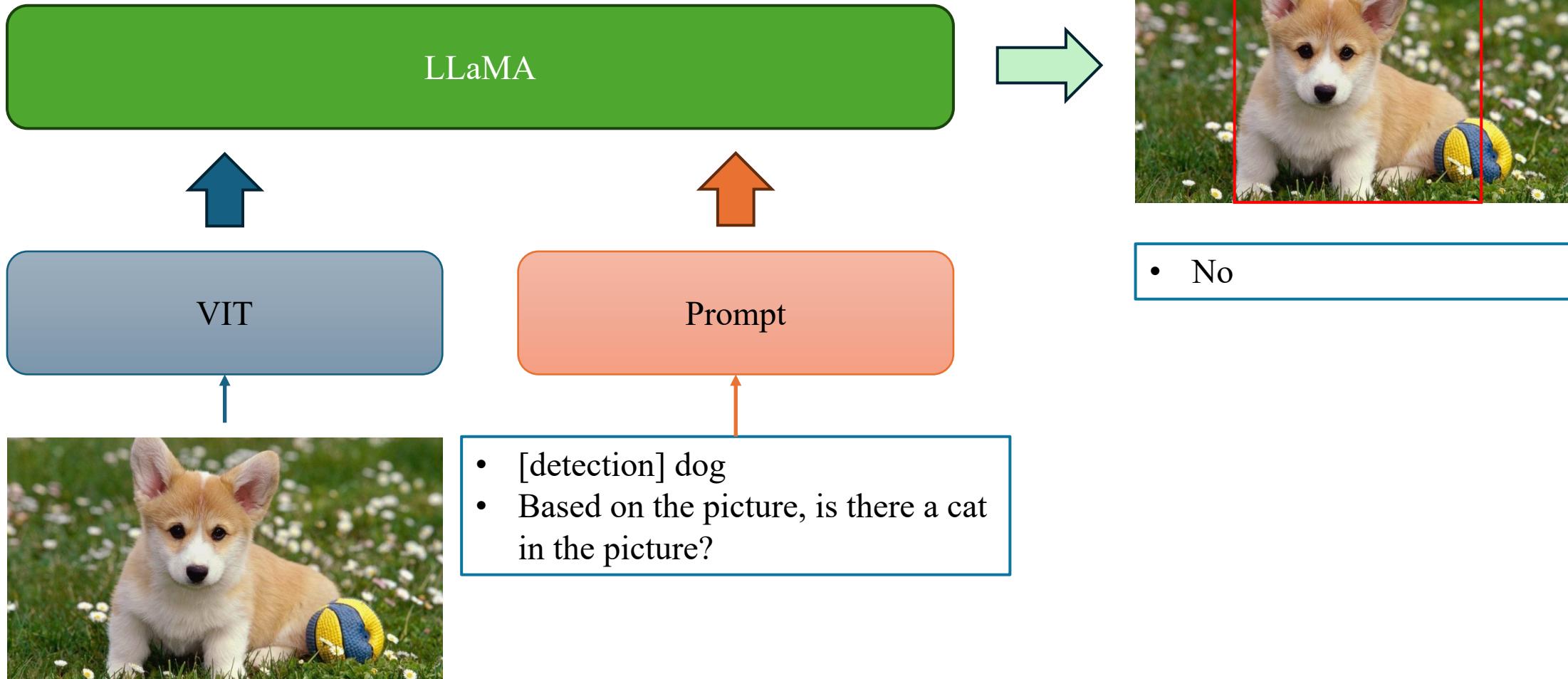
❖ Supervised Fine-tuning (SFT) on MiniGPT-V



MiniGPT-v2 is a unified vision-language model designed to handle diverse tasks such as image captioning, visual question answering, and visual grounding through simple multi-modal instructions. By introducing task-specific identifiers during training, the model improves its ability to distinguish and perform each task efficiently, achieving strong results across multiple benchmarks.

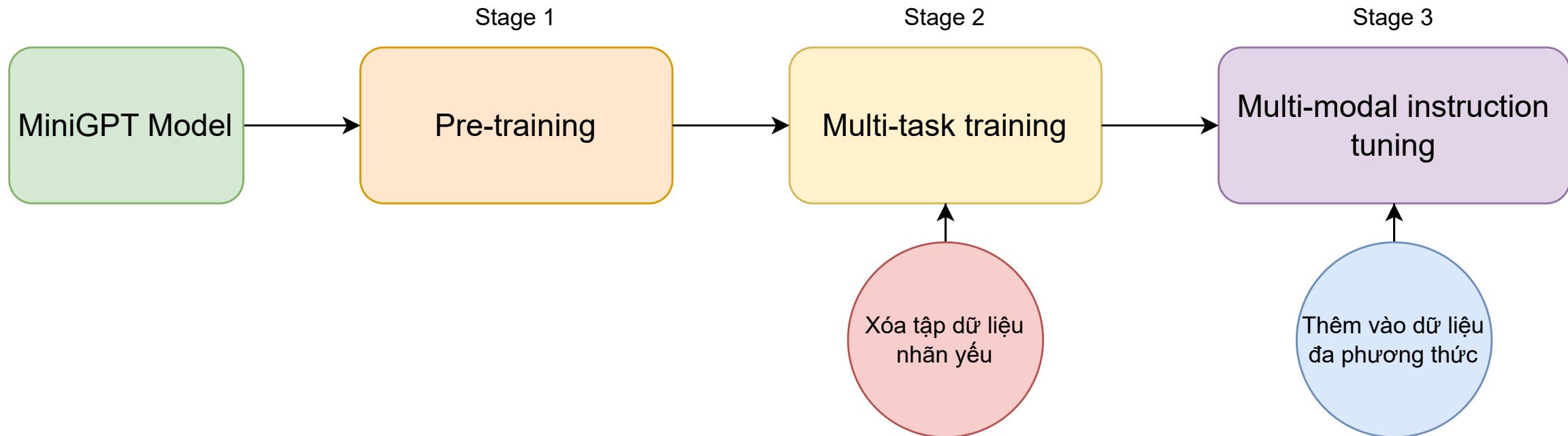
Training LVLMs

❖ Overview model



Training LVLMs

❖ MiniGPT training stages



Data types	Dataset	Stage 1	Stage 2	Stage 3
Weakly-labeled	GRIT-20M (REC and REG), LAION, CC3M, SBU	✓	✗	✗
Grounded caption	GRIT-20M	✓	✗	✗
Caption	COCO caption, Text Captions	✓	✓	✓
REC	RefCOCO, RefCOCO+, RefCOCOg, Visual Genome	✓	✓	✓
REG	RefCOCO, RefCOCO+, RefCOCOg	✓	✓	✓
VQA	GQA, VQAv2, OCR-VQA, OK-VQA, AOK-VQA	✓	✓	✓
Multimodal instruction	LLaVA dataset, Flickr30k, Multi-task conversation	✗	✗	✓
Langauge dataset	Unnatural Instructions	✗	✗	✓

Training LVLMs

❖ Stage 1: Pre-training - Vision-language knowledge

Weakly-labeled datasets



CC3M LAION
SBU GRIT-20M



Referring expression comprehension (REC)
Referring expression generation (REG)
Grounded image captioning

Fine-grained datasets



COCO caption Text Captions
RefCOCO RefCOCO+,...
VQA-v2, OCR-VQA OK-VQA,...

image captioning
Referring expression comprehension (REC)
VQA

Training LVLMs

❖ Stage 2: Multi-task training - Improve the performance on each task

Data types	Dataset	Drop Weakly-labeled		
		Stage 1	Stage 2	Stage 3
Weakly-labeled	GRIT-20M (REC and REG), LAION, CC3M, SBU	✓	✗	✗
Grounded caption	GRIT-20M	✓	✗	✗
Caption	COCO caption, Text Captions	✓	✓	✓
REC	RefCOCO, RefCOCO+, RefCOCOg, Visual Genome	✓	✓	✓
REG	RefCOCO, RefCOCO+, RefCOCOg	✓	✓	✓
VQA	GQA, VQAv2, OCR-VQA, OK-VQA, AOK-VQA	✓	✓	✓
Multimodal instruction	LLaVA dataset, Flickr30k, Multi-task conversation	✗	✗	✓
Langauge dataset	Unnatural Instructions	✗	✗	✓

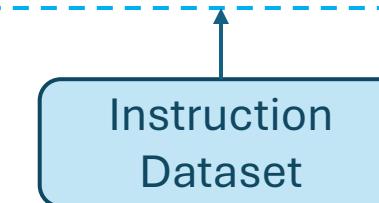
↓

Improve the performance on each task

Training LVLMs

❖ Stage 3: Multi-model instruction training

Data types	Dataset	Stage 1	Stage 2	Stage 3
Weakly-labeled	GRIT-20M (REC and REG), LAION, CC3M, SBU	✓	✗	✗
Grounded caption	GRIT-20M	✓	✗	✗
Caption	COCO caption, Text Captions	✓	✓	✓
REC	RefCOCO, RefCOCO+, RefCOCOg, Visual Genome	✓	✓	✓
REG	RefCOCO, RefCOCO+, RefCOCOg	✓	✓	✓
VQA	GQA, VQAv2, OCR-VQA, OK-VQA, AOK-VQA	✓	✓	✓
Multimodal instruction	LLaVA dataset, Flickr30k, Multi-task conversation	✗	✗	✓
Language dataset	Unnatural Instructions	✗	✗	✓



Boosting its ability to engage in conversations as a chatbot

Training LVMs

❖ Overview method

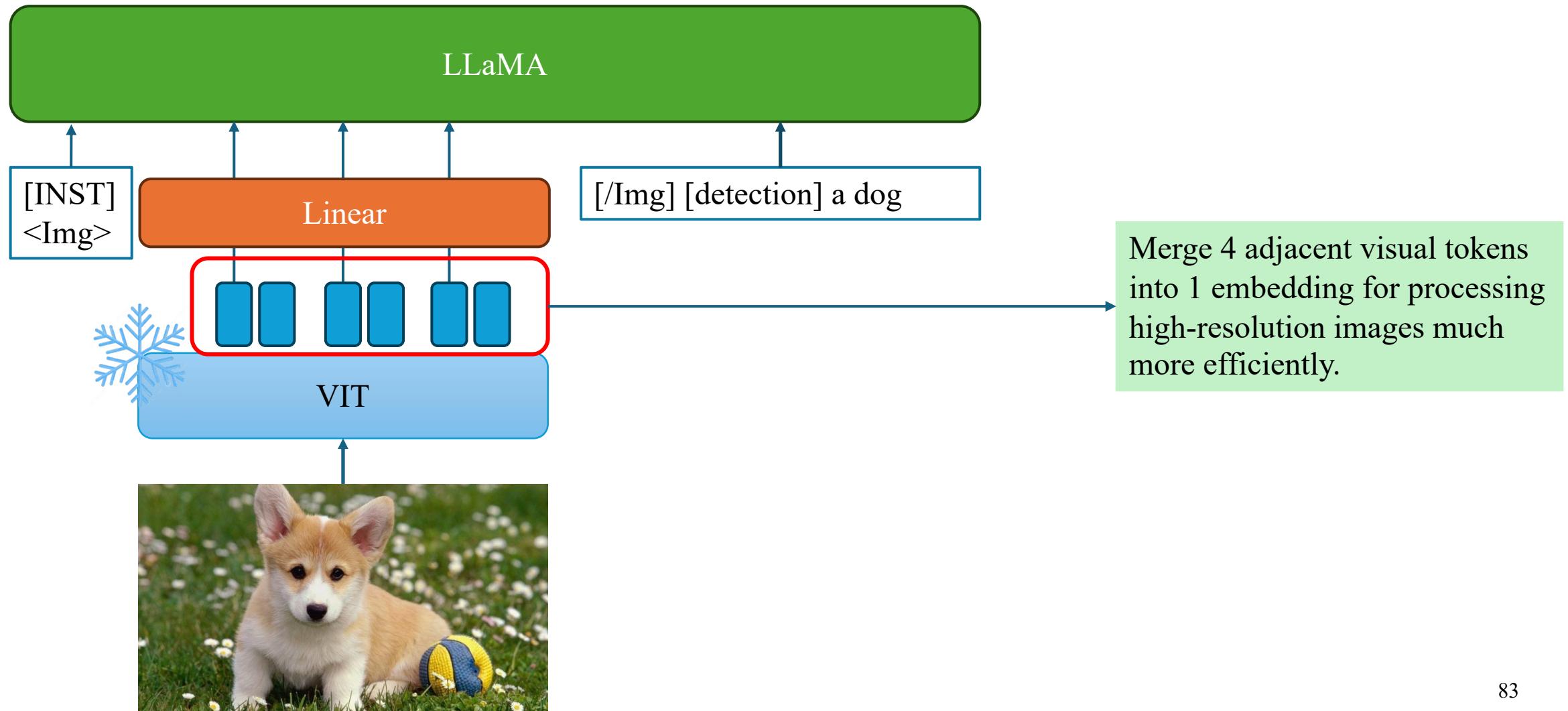
Task	Identifier
VQA (Visual Question Answering)	[vqa]
Caption (Image Captioning)	[caption]
Grounded Image Captioning	[grounding]
REC (Referring Expression Comprehension)	[refer]
REG (Referring Expression Generation)	[identify]
Object Parsing and Grounding	[detection]

Prompt Instruction

[INST] <ImageHere> [Task Identifier] Instruction [/INST]

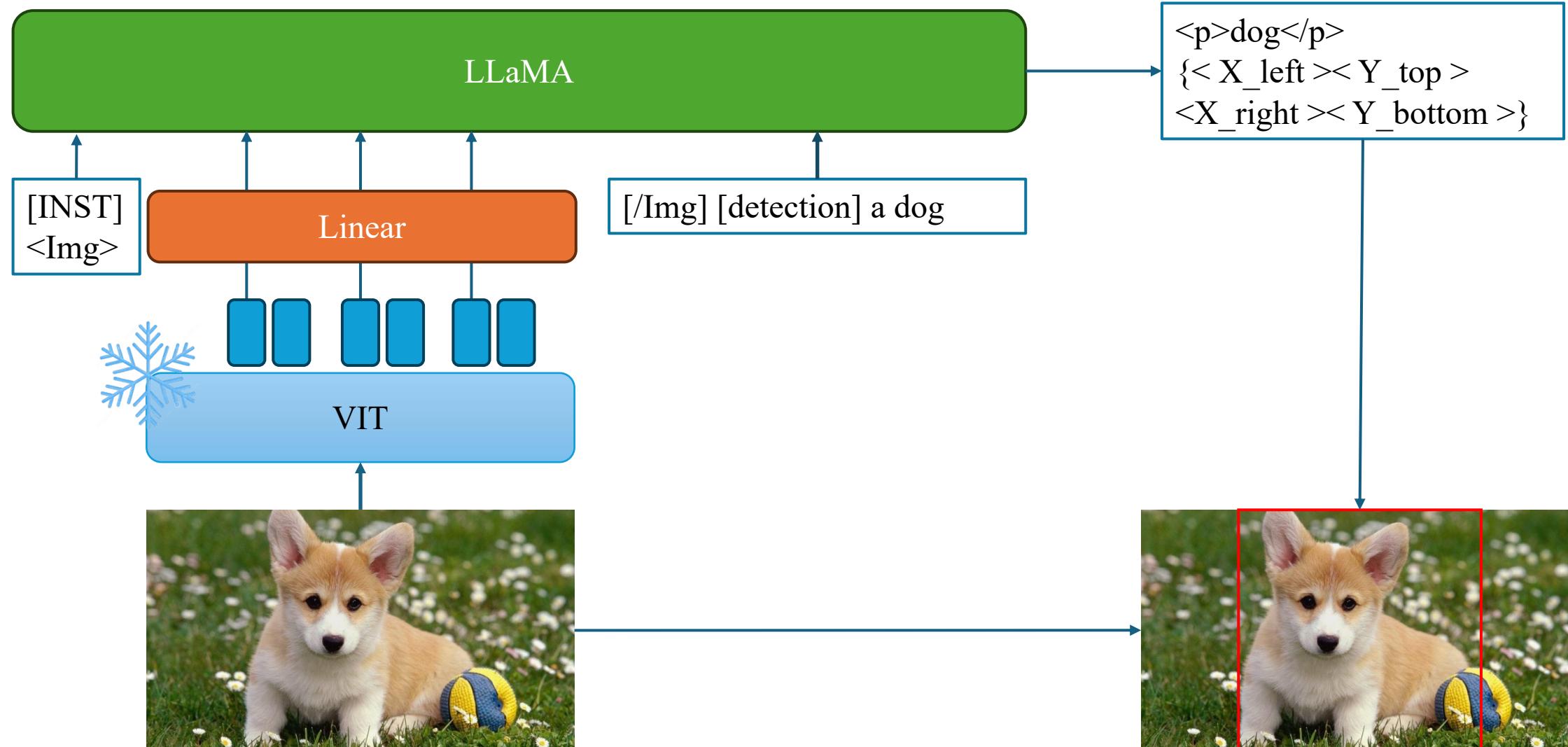
Training LVLMs

❖ Overview method



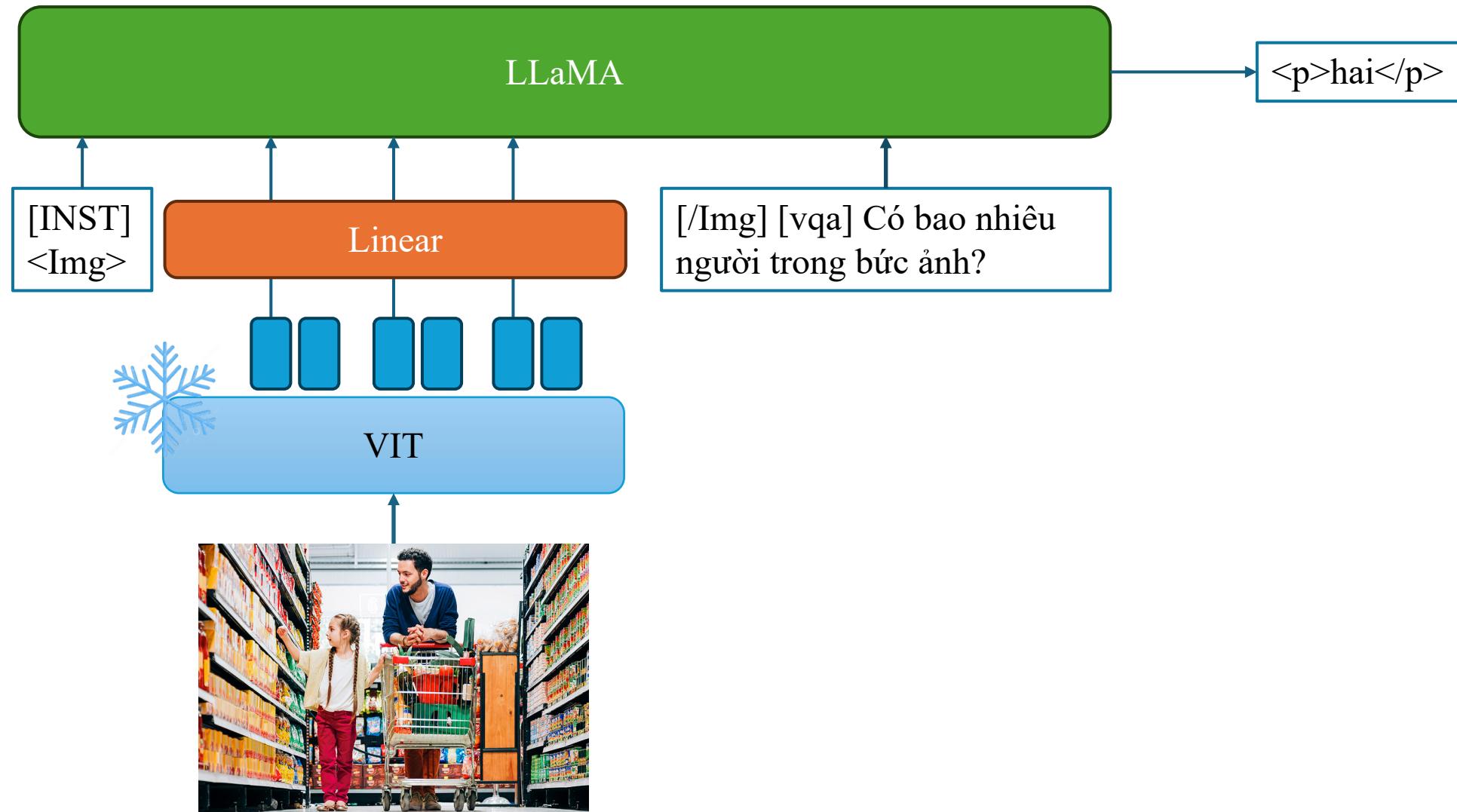
Training LVLMs

❖ Overview method



Training LVLMs

❖ Overview method

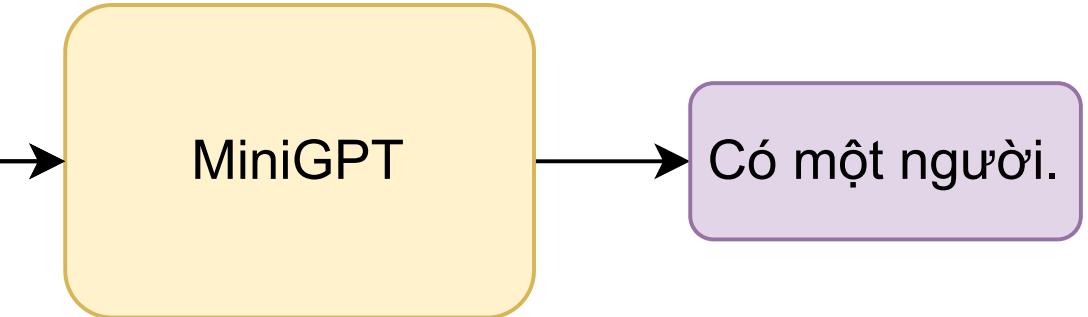


Training LLMs

❖ Finetuning VQA task

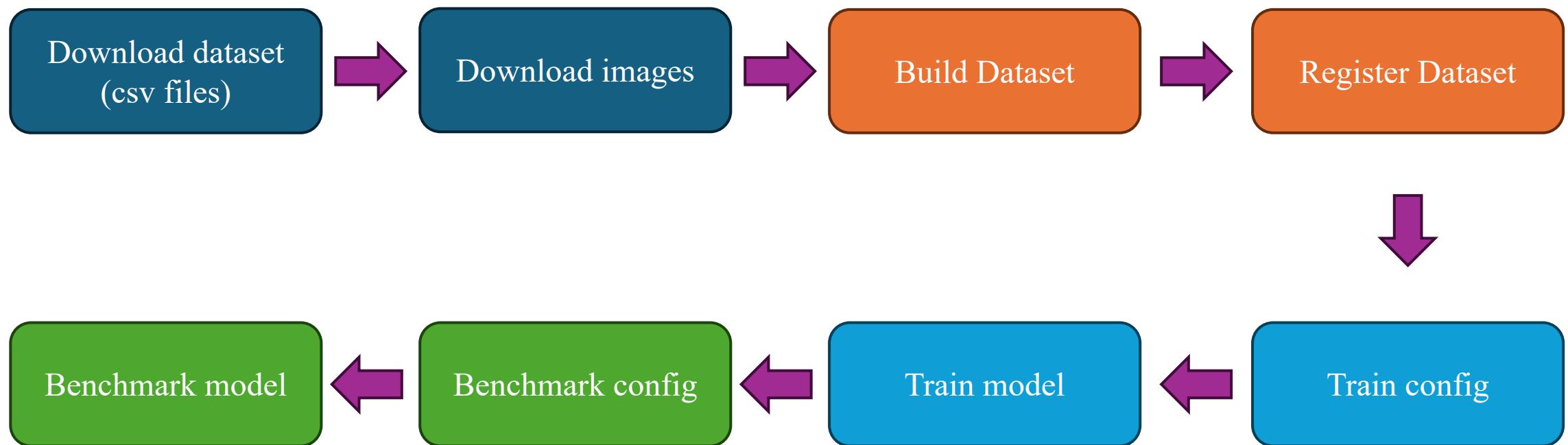


Có bao nhiêu người trong quán DodamDodam ?



Training LVLMs

❖ Pipeline



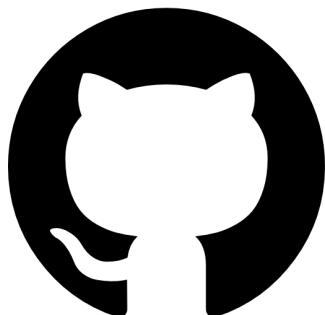


Training LVLMs

❖ Download dataset

Path: ./data

```
git clone https://github.com/kh4nh12/ViVQA
cd ViVQA
gdown 1jvFik5rNEk--in1Ismk-aiA6NqDfJyXI
unzip -q ./vivqa_images.zip
```





Training LVMs

❖ Build Dataset

Path: minigpt4/datasets/datasets/vqa_datasets.py

```
● ● ●

1  class ViVQADataset(torch.utils.data.Dataset):
2      def __init__(self, vis_processor, text_processor, vis_root, ann_path):
3          self.vis_root = vis_root
4          self.vis_processor = vis_processor
5          self.text_processor = text_processor
6
7          self.instruction_pool = [
8              "[vqa] {}",
9              "[vqa] Based on the image, respond to this question with a short answer: {}"
10         ]
11         self.data = pd.read_csv(ann_path)
12         self.img_name_prefix = "COCO_train2014_"
13
14     def __len__(self):
15         return len(self.data)
```



Training LVMs

❖ Build Dataset

Path: minigpt4/datasets/datasets/vqa_datasets.py

```
● ● ●

1 def __getitem__(self, index):
2     sample = self.data.iloc[index]
3     image_id = str(sample['img_id']).zfill(12)
4     image_path = os.path.join(self.vis_root, f"{image_id}.jpg")
5     image = Image.open(image_path).convert("RGB")
6     image = self.vis_processor(image)
7     question = self.text_processor(sample["question"])
8     answer = self.text_processor(sample["answer"])
9     instruction = random.choice(self.instruction_pool).format(question)
10    instruction = "<Img><ImageHere></Img> {}".format(instruction)
11    return {
12        "image": image,
13        "instruction_input": instruction,
14        "answer": answer,
15        "image_id": image_id
16    }
```



Training LVMs

❖ Register Dataset

Path: minigpt4/datasets/builders/image_text_pair_builder.py

```
● ● ●
1  from minigpt4.datasets.datasets.vqa_datasets import ViVQADataset
2
3  @registry.register_builder("vivqa")
4  class ViVQABuilder(BaseDatasetBuilder):
5      train_dataset_cls = ViVQADataset
6      DATASET_CONFIG_DICT = {
7          "default": "configs/datasets/vivqa/default.yaml",
8      }
9      def build_datasets(self):
10         logging.info("Building datasets...")
11         self.build_processors()
12         build_info = self.config.build_info
13         datasets = dict()
14
15         # create datasets
16         dataset_cls = self.train_dataset_cls
17         datasets['train'] = dataset_cls(
18             vis_processor=self.vis_processors["train"],
19             text_processor=self.text_processors["train"],
20             vis_root=build_info.image_path,
21             ann_path=build_info.ann_path,
22         )
23         return datasets
```

Training LVLMs

❖ Train config

```
model:  
    arch: minigpt_v2  
    model_type: pretrain  
    max_txt_len: 1024  
    image_size: 448  
    end_sym: "</s>"  
    llama_model: "meta-llama/Llama-2-7b-chat-hf"  
    ckpt: "./ckpt/checkpoint_stage3.pth"  
    use_grad_checkpoint: True  
    low_resource: True  
    chat_template: True  
    lora_r: 64  
    lora_alpha: 16
```

```
datasets:  
    vivqa:  
        batch_size: 2  
        vis_processor:  
            train:  
                name: "blip2_image_train"  
                image_size: 448  
                text_processor:  
                    train:  
                        name: "blip_caption"  
                        sample_ratio: 100
```

Path: train_configs/minigptv2_finetune_vivqa.yaml

```
run:  
    task: image_text_pretrain  
    lr_sched: "linear_warmup_cosine_lr"  
    init_lr: 1e-5  
    min_lr: 1e-6  
    warmup_lr: 1e-6  
  
    weight_decay: 0.05  
    max_epoch: 2  
    num_workers: 6  
    warmup_steps: 500  
    iters_per_epoch: 2000  
  
    seed: 42  
    output_dir: "vivqa_outputs"
```

Script:
python train.py \
--cfg-path train_configs/minigptv2_finetune_vivqa.yaml

Training LVMs

❖ Benchmark config

Path: eval_configs/minigptv2_benchmark_evaluation.yaml

```
1 model:  
2   arch: minigpt_v2  
3   model_type: pretrain  
4   max_txt_len: 500  
5   end_sym: "</s>"  
6   low_resource: True  
7   prompt_template: '[INST] {} [/INST]'  
8   llama_model: "meta-llama/Llama-2-7b-chat-hf"  
9   ckpt: "./minigpt4/vivqa_outputs/20250516072/checkpoint_1.pth"  
10  lora_r: 64  
11  lora_alpha: 16  
12  
13 datasets:  
14   cc_sbu_align:  
15     vis_processor:  
16       train:  
17         name: "blip2_image_eval"  
18         image_size: 448  
19     text_processor:  
20       train:  
21         name: "blip_caption"
```

```
1 evaluation_datasets:  
2   vivqa:  
3     batch_size: 4  
4     eval_file_path: ./data/ViVQA/test.csv  
5     img_path: ./data/ViVQA/images  
6     max_new_tokens: 20
```

Script:
python eval_vivqa.py \
--cfg-path ./eval_configs/minigptv2_benchmark_evaluation.yaml



Training LVLMs

❖ Demo config



```
1 model:
2   arch: minigpt_v2
3   model_type: pretrain
4   max_txt_len: 500
5   end_sym: "</s>"
6   low_resource: True
7   prompt_template: '[INST] {} [/INST]'
8   ckpt: "./ckpt/v2_checkpoint_4_vivqa.pth"
9   lora_r: 64
10  lora_alpha: 16
11
12
13 datasets:
14   cc_sbu_align:
15     vis_processor:
16       train:
17         name: "blip2_image_eval"
18         image_size: 448
19     text_processor:
20       train:
21         name: "blip_caption"
22
23 run:
24   task: image_text_pretrain
```

Path: eval_configs/minigptv2_eval_vivqa.yaml

Training LVMs

❖ Demo

Script: python demo_v2.py --cfg-path eval_configs/minigptv2_eval_vivqa.yaml --gpu-id 0

MiniGPT-v2 Demo

Project Page

Image 

Temperature 0,6

Restart

For Abilities Involving Visual Grounding:

1. Grounding: CLICK **Send** to generate a grounded image description.
2. Refer: Input a referring object and CLICK **Send**.
3. Detection: Write a caption or phrase, and CLICK **Send**.
4. Identify: Draw the bounding box on the uploaded image window and CLICK **Send** to generate the bounding box. (CLICK "clear" button before re-drawing next time).
5. VQA: Input a visual question and CLICK **Send**.

Task Shortcuts: No Tag, Grounding, Refer, Detection, Identify, VQA

Hint: Send a question to get a short answer

Upload your image and chat

Send

95

[vqa] có bao nhiêu người trong bức ảnh

hai

Training LVLMs

❖ Finetuning detection task



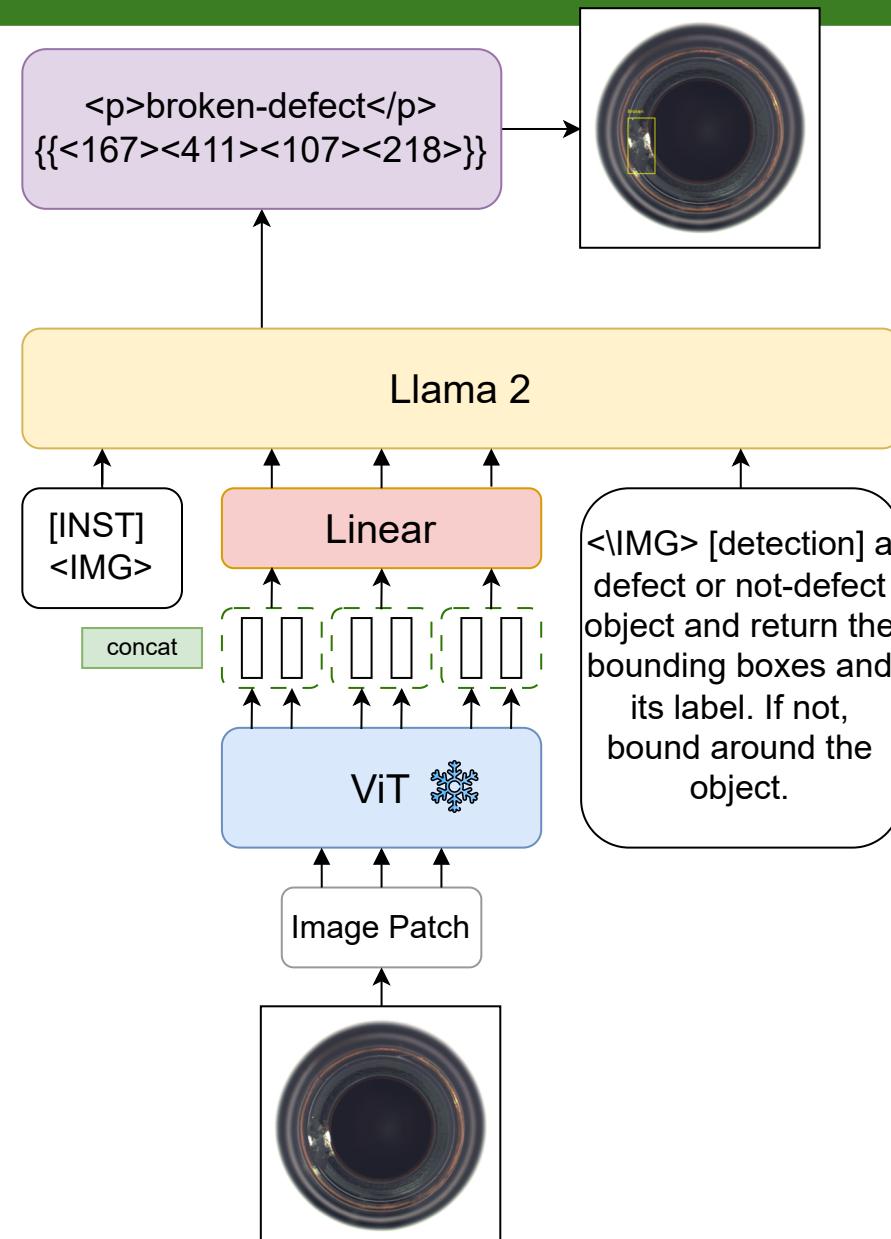
<p>broken-defect</p>
{<167><411><107><218>}

Training LVLMs

❖ Finetuning detection task

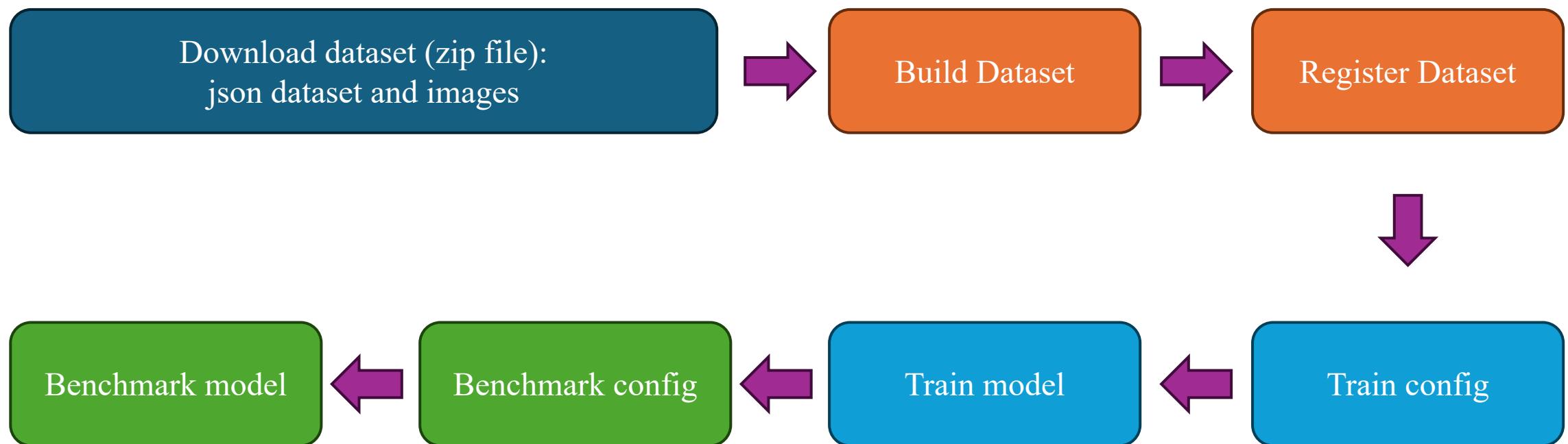
MVTec
dataset

```
● ● ●  
1 [  
2 {  
3     "image_path": "./capsule/broken/057.png",  
4     "bbox": [  
5         52,  
6         36,  
7         64,  
8         42  
9     ],  
10    "class": "capsule",  
11    "is_broken": true,  
12    "height": 1000,  
13    "width": 1000  
14 },  
15 {  
16     "image_path": "./wood/good/004.png",  
17     "bbox": [  
18         0,  
19         0,  
20         100,  
21         100  
22     ],  
23     "class": "wood",  
24     "is_broken": false,  
25     "height": 1024,  
26     "width": 1024  
27 }  
28 ]
```



Training LVLMs

❖ Pipeline

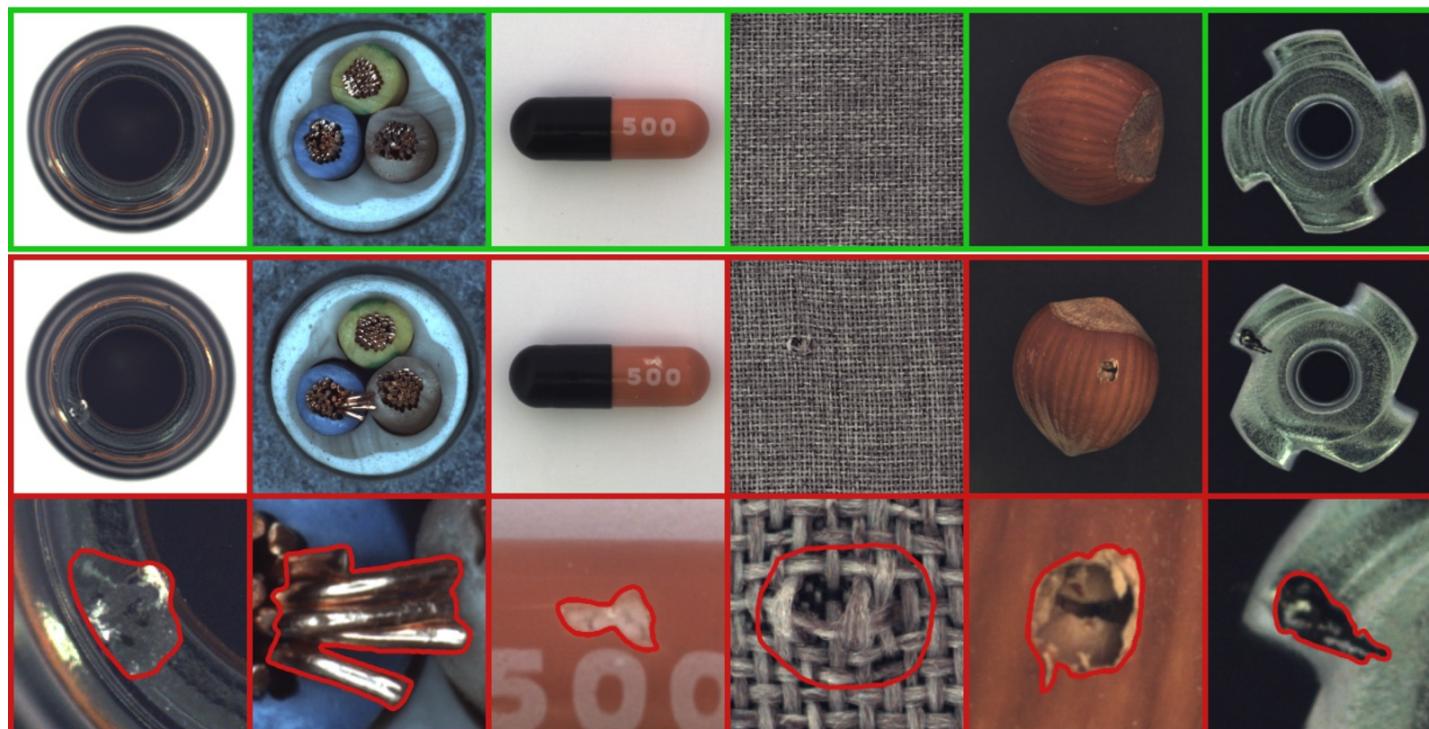


Training LVMs

❖ Download dataset

Path: ./data

```
gdown 1eOIJVGKLiyIPoGA5kQLS3LJtWmC8obK8  
unzip -q ./MVTEC_det.zip
```





Training LVMs

❖ Build Dataset

Path: minigpt4/datasets/datasets/mvtec_dataset.py

```
● ● ●

1  class MVTecDataset(Dataset):
2      def __init__(self, vis_processor, text_processor, vis_root, ann_path):
3          """
4              vis_root (string): Root directory of images (e.g. coco/images/)
5              ann_root (string): directory to store the annotation file
6          """
7          self.vis_root = vis_root
8
9          self.vis_processor = vis_processor
10         self.text_processor = text_processor
11
12         self.instruction_pool = [
13             '[detection] {}',
14         ]
15
16         with open(ann_path, 'r') as f:
17             self.ann = json.load(f)
18
19     def __len__(self):
20         return len(self.ann)
```



Training LVLMs

❖ Build Dataset

Path: minigpt4/datasets/datasets/mvtec_dataset.py

```
● ● ●  
1  def __getitem__(self, index):  
2      info = self.ann[index]  
3      gt_bbox = info["bbox"]  
4      ans_cls = info["class"]  
5  
6      image_path = os.path.join(self.vis_root, info['image_path'])  
7      image = Image.open(image_path).convert("RGB")  
8      image = self.vis_processor(image)  
9  
10     input = "a defect or not-defect object and return the bounding boxes and its label. If not, bound around the object."  
11  
12     ans_defect = "defect" if info["is_broken"] == True else "not-defect"  
13     ans_para = f"<p>{ans_cls}-{ans_defect}</p>"  
14     answer = f"{ans_para}{{<{gt_bbox[0]}><{gt_bbox[1]}><{gt_bbox[2]}><{gt_bbox[3]}>}}"  
15  
16     instruction = random.choice(self.instruction_pool).format(input)  
17     instruction = "<Img><ImageHere></Img> {}".format(instruction)  
18  
19     return {  
20         "image": image,  
21         "instruction_input": instruction,  
22         "answer": answer,  
23         "image_id": info['image_path'],  
24     }
```



Training LVMs

❖ Register Dataset

Path: minigpt4/datasets/builders/image_text_pair_builder.py

```
● ● ●

1 @registry.register_builder("mvtec_ad")
2 class MVTECADBuilder(BaseDatasetBuilder):
3     train_dataset_cls = MVTecDataset
4     DATASET_CONFIG_DICT = {
5         "default": "configs/datasets/mvtec/default.yaml",
6     }
7     def build_datasets(self):
8         logging.info("Building datasets...")
9         self.build_processors()
10        build_info = self.config.build_info
11        datasets = dict()
12
13        # create datasets
14        dataset_cls = self.train_dataset_cls
15        datasets['train'] = dataset_cls(
16            vis_processor=self.vis_processors["train"],
17            text_processor=self.text_processors["train"],
18            ann_path=build_info.ann_path,
19            vis_root=build_info.image_path,
20        )
21        return datasets
```

Training LVLMs

❖ Train config

```
● ● ●  
1 model:  
2   arch: minigpt_v2  
3   model_type: pretrain  
4   max_txt_len: 1024  
5   image_size: 448  
6   end_sym: "</s>"  
7   llama_model: "meta-llama/Llama-2-7b-chat-hf"  
8   ckpt: "./ckpt/checkpoint_stage3.pth"  
9   use_grad_checkpoint: True  
10  low_resource: True  
11  chat_template: True  
12  lora_r: 64  
13  lora_alpha: 16
```

Path: train_configs/minigptv2_finetune_mvtec.yaml

```
● ● ●  
1 run:  
2   task: image_text_pretrain  
3   lr_sched: "linear_warmup_cosine_lr"  
4   init_lr: 1e-5  
5   min_lr: 1e-6  
6   warmup_lr: 1e-6
```

```
weight_decay: 0.05  
max_epoch: 2  
num_workers: 6  
warmup_steps: 1000  
iters_per_epoch: 2000  
  
seed: 42  
output_dir: "mvtec_outputs"
```

```
● ● ●  
1 datasets:  
2   mvtec_ad:  
3     batch_size: 2  
4     vis_processor:  
5       train:  
6         name: "blip2_image_train"  
7         image_size: 448  
8     text_processor:  
9       train:  
10      name: "blip_caption"  
11      sample_ratio: 100
```

Training LVLMs

❖ Train config

Path: train_configs/minigptv2_finetune_mvtec.yaml

```
1 model:  
2   arch: minigpt_v2  
3   model_type: pretrain  
4   max_txt_len: 1024  
5   image_size: 448  
6   end_sym: "</s>"  
7   llama_model: "meta-llama/Llama-2-7b-chat-hf"  
8   ckpt: "./ckpt/checkpoint_stage3.pth"  
9   use_grad_checkpoint: True  
10  low_resource: True  
11  chat_template: True  
12  lora_r: 64  
13  lora_alpha: 16
```

Script:

```
python train.py \  
--cfg-path train_configs/minigptv2_finetune_mvtec.yaml
```

```
1 datasets:  
2   mvtec_ad:  
3     batch_size: 2  
4     vis_processor:  
5       train:  
6         name: "blip2_image_train"  
7         image_size: 448  
8     text_processor:  
9       train:  
10        name: "blip_caption"  
11        sample_ratio: 100
```

```
text_pretrain  
linear_warmup_cosine_lr":  
5  
warmup_lr: 1e-6  
weight_decay: 0.05  
max_epoch: 2  
num_workers: 6  
warmup_steps: 1000  
iters_per_epoch: 2000  
  
seed: 42  
output_dir: "mvtec_outputs"
```

Training LVLMs

❖ Benchmark config

Path: eval_configs/minigptv2_benchmark_evaluation.yaml

```
● ● ●  
1 model:  
2   arch: minigpt_v2  
3   model_type: pretrain  
4   max_txt_len: 500  
5   end_sym: "</s>"  
6   low_resource: True  
7   prompt_template: '[INST] {} [/INST]'  
8   llama_model: "meta-llama/Llama-2-7b-chat-hf"  
9   ckpt: "./minigpt4/mvtec_outputs/20250331072/checkpoint_1.pth"  
10  lora_r: 64  
11  lora_alpha: 16
```

```
● ● ●  
1 datasets:  
2   cc_sbu_align:  
3     vis_processor:  
4       train:  
5         name: "blip2_image_eval"  
6         image_size: 448  
7     text_processor:  
8       train:  
9         name: "blip_caption"
```

```
● ● ●  
1 evaluation_datasets:  
2   mvtec_ad:  
3     batch_size: 4  
4     eval_file_path: ./data/MVTEC_det/val_data.json  
5     max_new_tokens: 40
```

Script:
`python eval_mvtec.py \
--cfg-path ./eval_configs/minigptv2_benchmark_evaluation.yaml`



Training LVLMs

❖ Demo config

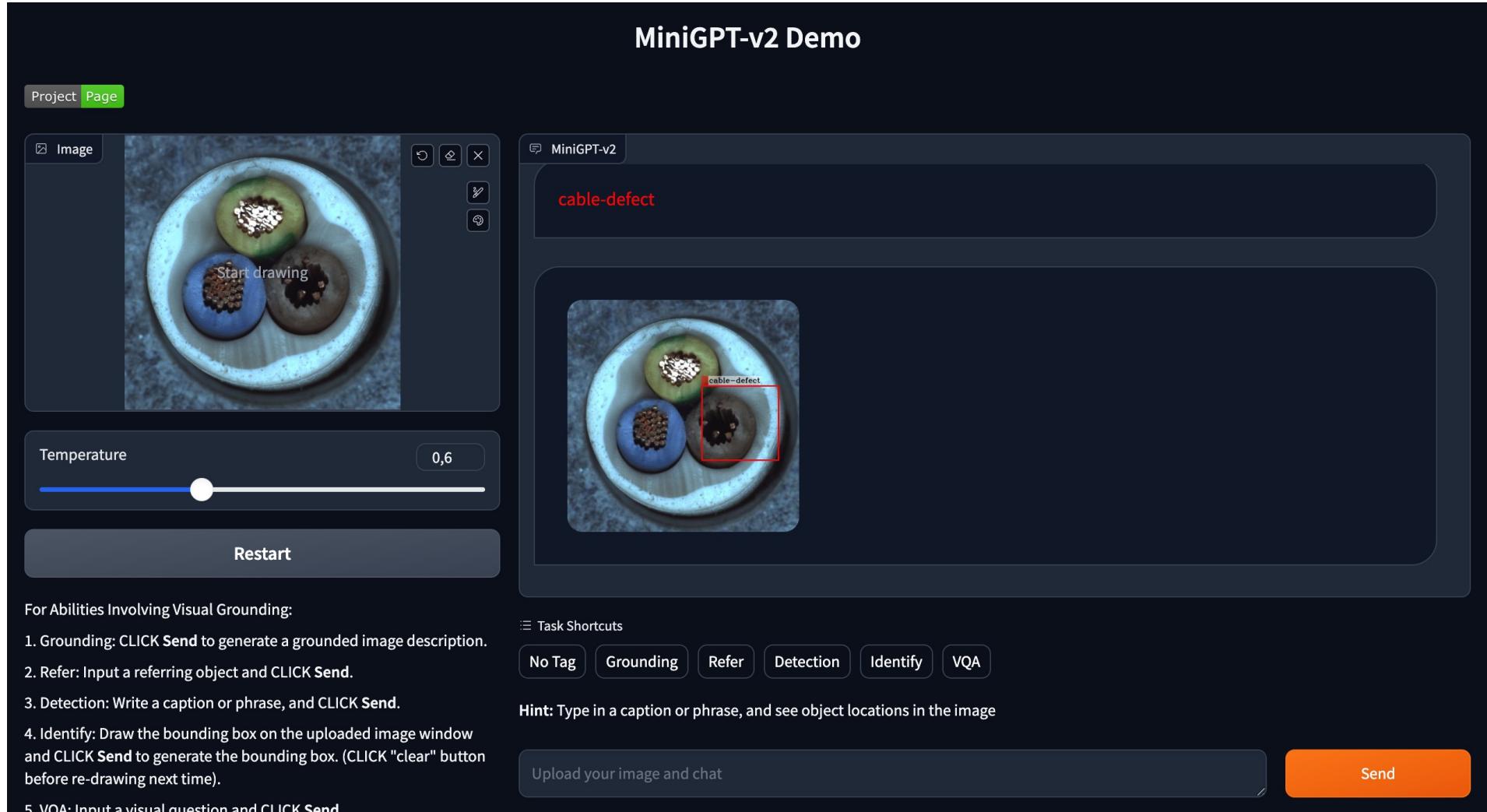
```
● ● ●  
1 model:  
2   arch: minigpt_v2  
3   model_type: pretrain  
4   max_txt_len: 500  
5   end_sym: "</s>"  
6   low_resource: True  
7   prompt_template: '[INST] {} [/INST]'  
8   # ckpt: "./ckpt/checkpoint_stage3.pth"  
9   ckpt: "./ckpt/v2_checkpoint_4_mvtec.pth"  
10  lora_r: 64  
11  lora_alpha: 16  
12  
13  
14 datasets:  
15   cc_sbu_align:  
16     vis_processor:  
17       train:  
18         name: "blip2_image_eval"  
19         image_size: 448  
20     text_processor:  
21       train:  
22         name: "blip_caption"  
23  
24 run:  
25   task: image_text_pretrain  
26
```

Path: eval_configs/minigptv2_eval_mvtec.yaml

Training LVMs

❖ Demo

Script: `python demo_v2.py --cfg-path eval_configs/minigptv2_eval_mvtec.yaml --gpu-id 0`



Training LVLMs

❖ Investigation on training LVLMs

I: Traditional
Supervised
Fine-Tuning (SFT)



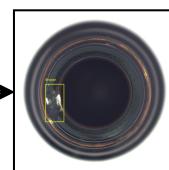
Có bao nhiêu người trong quán DodamDodam ?

MiniGPT

Có một người.



MiniGPT



<p>broken-defect</p>
<{<167><411><107><218>}>

II: Instruction
Tuning (IT)



Qwen2-VL



Summarization: Hóa đơn bán hàng của Vincommerce, ngày 14/08/2020, 21:37, mã hóa đơn H:00073344, NVBH:09021213...



Training LVMs

❖ Instruction Data

instruction string	input string	output string
Detect hate speech in the following sentence:	We should hate all those immigrants.	Yes
Find the total price for the items in the given list, given...	List: laptop, cellphone, computer mouse	30
Given a product description, identify the main product...	The Samsung Galaxy Fold has three AMOLED screens, one o...	The Samsung Galaxy Fold boasts a unique triple-screen design and an impressive long...
Output a way to reduce the effects of a given issue.	Air Pollution	One way to reduce the effects of air pollution is to reduce the amount of...
Give three proverbs related to the input.	Love	1. Love conquers all. 2. Love makes the world go round. 3. Love knows no bounds.
Summarize an article about the importance of being eco-...		Reducing consumption of resources and increasing respect and regard for our...

Instruction Prompting: Specific, directive statements or questions that guide the LLMs to perform a particular action.

Training LLMs

❖ Instruction Data



Positive



Neutral



Negative

Instruction: Given a sentence, classify its sentiment into one of three classes: positive, negative, neutral

Input: What is the weather like today?

Output: Neutral

Input: I feel bad

Output:



You

Instruction: You will be given a sentence, please classify its sentiment into one of three classes: positive, negative, neutral

Input: This is a very bad food.

Output: Negative

Input: Today's weather is not bad, don't you think?

Output:



ChatGPT

Neutral



You

Input: This restaurant is well-known by its customer service



ChatGPT

Positive



Training LVMs

❖ Instruction Data for LVMs

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

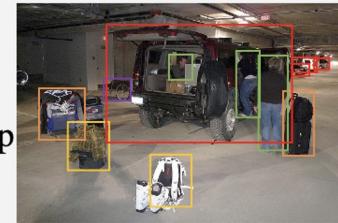
People try to fit all of their luggage in an SUV.

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], backpack: [0.384, 0.696, 0.485, 0.914], suitcase: ...<omitted>



Response type 1: conversation

Question: What type of vehicle is featured in the image?

Answer: The image features a black sport utility vehicle (SUV) ...<omitted>

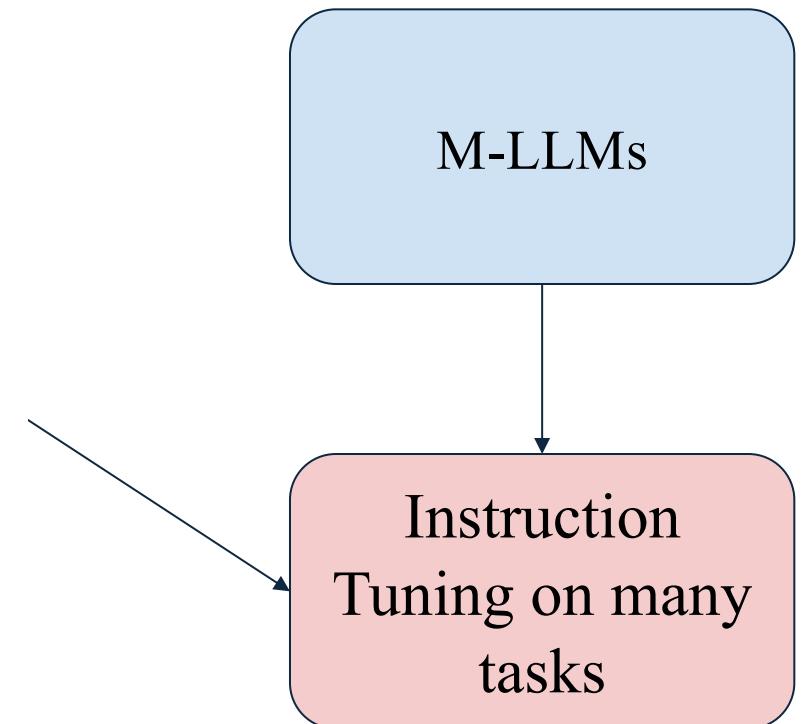
Response type 2: detailed description

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. ...<omitted>

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 ...<omitted>



Training LVMs

❖ Instruction Data for LVMs

- <Image> {Question}
- <Image> Question: {Question}
- <Image> {Question} A short answer to the question is
- <Image> Q: {Question} A:
- <Image> Question: {Question} Short answer:
- <Image> Given the image, answer the following question with no more than three words. {Question}
- <Image> Based on the image, respond to this question with a short answer: {Question}. Answer:
- <Image> Use the provided image to answer the question: {Question} Provide your answer as short as possible:
- <Image> What is the answer to the following question? "{Question}"
- <Image> The question "{Question}" can be answered using the image. A short answer is

Table 2. Instruction templates for VQA datasets, cited from [23]. <Image> and {Question} are the image and the question in the original VQA datasets, respectively.

Training LVLMs

❖ Instruction Data Template for LVLMs

<BOS> Below is an instruction that describes a task.
Write a response that appropriately completes the request

```
### Instruction: <instruction>  
### Input: {<image>, <text>}  
### Response: <output><EOS>
```

- ❖ **{instruction}**: A textual description of the task.
- ❖ **{<image>, <text>}**: Input of the data sample.
- ❖ **{output}**: Output if the data sample.
- ❖ **<BOS>**: The start input token.
- ❖ **<EOS>**: The end input token.

Training LVMs

❖ Instruction Tuning Qwen2-VL

id	int32	image	image · width (px)	description	string · lengths	extractions	string · lengths
1	1.66k		352	1.02k	45	2	1.45k
443		Hóa đơn bán hàng của Vincommerce,...	{'Tên cửa hàng': 'Vincommerce', 'Địa chỉ': ...}				
7		Hóa đơn bán hàng của Saigon Co.op...	{'Tên cửa hàng': 'Saigon Co.op', 'Chi nhánh': ...}				
305		Hóa đơn được xuất bởi Saigon Co.op,...	{'Tên cửa hàng': 'Saigon Co.op', 'Chi nhánh': 'Fo...}				

Dataset Viet Receipt VQA:
<https://huggingface.co/datasets/5CD-AI/Viet-Receipt-VQA>



Qwen2-VL



Summarization: Hóa đơn bán hàng của Vincommerce, ngày 14/08/2020, 21:37, mã hóa đơn H:00073344, NVBH:09021213...

Training LVMs

❖ Code: Import necessary libraries

```
1 import os
2
3 os.environ["CUDA_VISIBLE_DEVICES"] = "2"
4
5 import torch
6 from datasets import load_dataset
7 from unsloth import FastVisionModel
8
9 from unsloth import is_bf16_supported
10 from unsloth.trainer import UnslothVisionDataCollator
11 from trl import SFTTrainer, SFTConfig
12 from transformers import TextStreamer
13
14 device = "cuda" if torch.cuda.is_available() else "cpu"
```





Training LVLMs

❖ Code: Load Qwen2-VL model



```
1 model_name = "unslloth/Qwen2-VL-2B-Instruct"
2 model, tokenizer = FastVisionModel.from_pretrained(
3     model_name,
4     load_in_4bit = True,
5     use_gradient_checkpointing = "unslloth",
6 )
```



Training LVLMs

❖ Code: Load LoRA Config

```
1 model = FastVisionModel.get_peft_model(  
2     model,  
3     finetune_vision_layers = True,  
4     finetune_language_layers = True,  
5     finetune_attention_modules = True,  
6     finetune_mlp_modules = True,  
7     r = 16,  
8     lora_alpha = 16,  
9     lora_dropout = 0,  
10    bias = "none",  
11    random_state = 59,  
12    use_rslora = False,  
13    loftq_config = None)
```





Training LVMs

❖ Code: Load and process dataset

```
● ● ●

1 dataset_name = "5CD-AI/Viet-Receipt-VQA"
2 dataset = load_dataset(dataset_name, split = "train").select(range(500))
3
4 instruction = "Tổng hợp và mô tả nội dung của hóa đơn trong ảnh thành một đoạn văn"
5
6 def convert_to_conversation(sample):
7
8     conversation = [
9         { "role": "user",
10            "content" : [
11                {"type" : "text", "text" : instruction},
12                {"type" : "image", "image" : sample["image"]} ]
13        },
14        { "role" : "assistant",
15            "content" : [
16                {"type" : "text", "text" : sample["description"]} ]
17        },
18    ]
19
20    return { "messages" : conversation }
21
22 converted_dataset = [convert_to_conversation(sample) for sample in dataset]
```

```
conversation = [
{ "role": "user",
"content" : [
{"type" : "text", "text" : instruction},
{"type" : "image", "image" : sample["image"]} ],
},
{ "role" : "assistant",
"content" : [
{"type" : "text", "text" : sample["description"]} ],
}]
```

Training LVLMs

❖ Code: Configure training config

```
●●●  
1 training_args = SFTConfig(  
2     per_device_train_batch_size = 1,  
3     gradient_accumulation_steps = 2,  
4     warmup_steps = 5,  
5     max_steps = 200,  
6     # num_train_epochs = 1,  
7     learning_rate = 2e-4,  
8     fp16 = not is_bf16_supported(),  
9     bf16 = is_bf16_supported(),  
10    logging_steps = 1,  
11    optim = "adamw_8bit",  
12    weight_decay = 0.01,  
13    lr_scheduler_type = "linear",  
14    seed = 59,  
15    output_dir = "outputs",  
16    report_to = "none",  
17    remove_unused_columns = False,  
18    dataset_text_field = "",  
19    dataset_kwargs = {"skip_prepare_dataset": True},  
20    dataset_num_proc = 4,  
21    max_seq_length = 2048,  
22 )
```

```
●●●  
1 FastVisionModel.for_training(model)  
2  
3 trainer = SFTTrainer(  
4     model = model,  
5     tokenizer = tokenizer,  
6     data_collator = UnslotVisionDataCollator(model, tokenizer),  
7     train_dataset = converted_dataset,  
8     args = training_args)  
9  
10 trainer_stats = trainer.train()
```

Training LVMs

❖ Code: Inference



Response from fine-tuned Qwen2-VL-2B-Instruct:
Hóa đơn bán hàng của Minimart Anan, Chợ Sủi Phú Thị Gia Lâm với 2 mặt hàng: 8934868149064 nrc sunlight mềm điều chỉnh 7 N0 và 291028000000 sáp thơm glade NT. Tổng cộng 114.000 VND, khách trả 200.000 VND, tiền trả lại 86.000 VND. Số GD: 000AC2212008002095, ngày 15/08/2020-17:17, thu ngân: BH3

Training LVLMs

❖ Evaluation LVLMs

Response A

Because the audio experience
is unrivaled

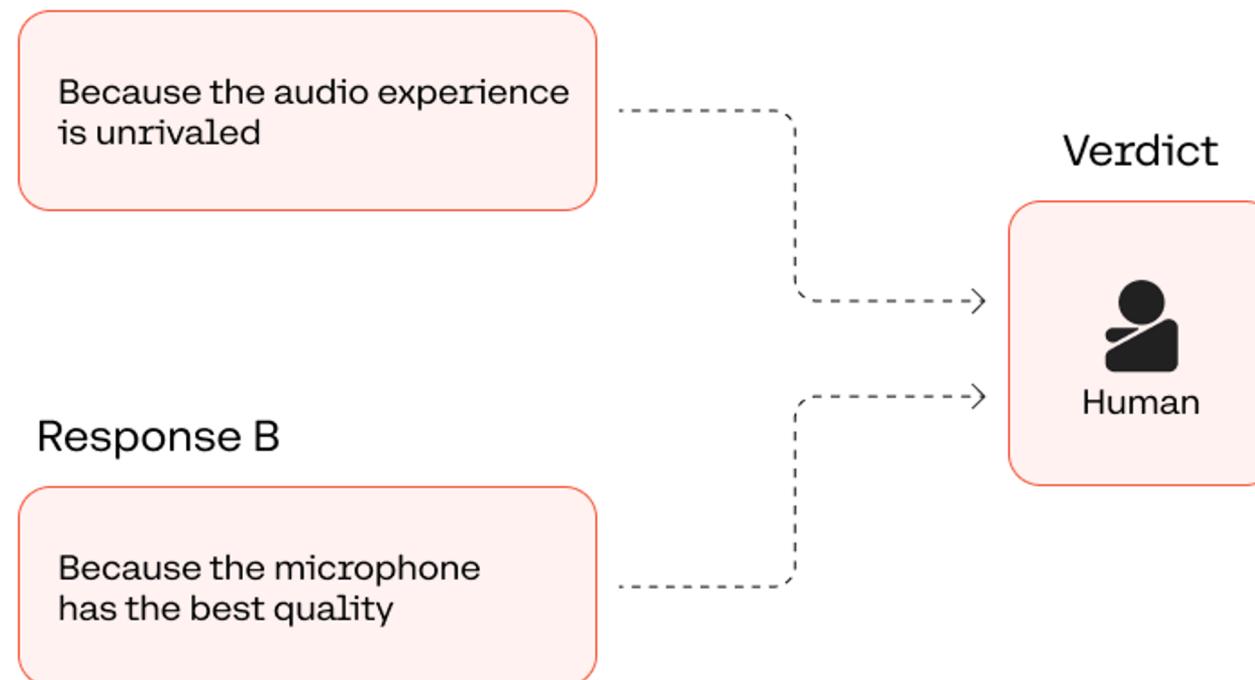
Response B

Because the microphone
has the best quality

Verdict

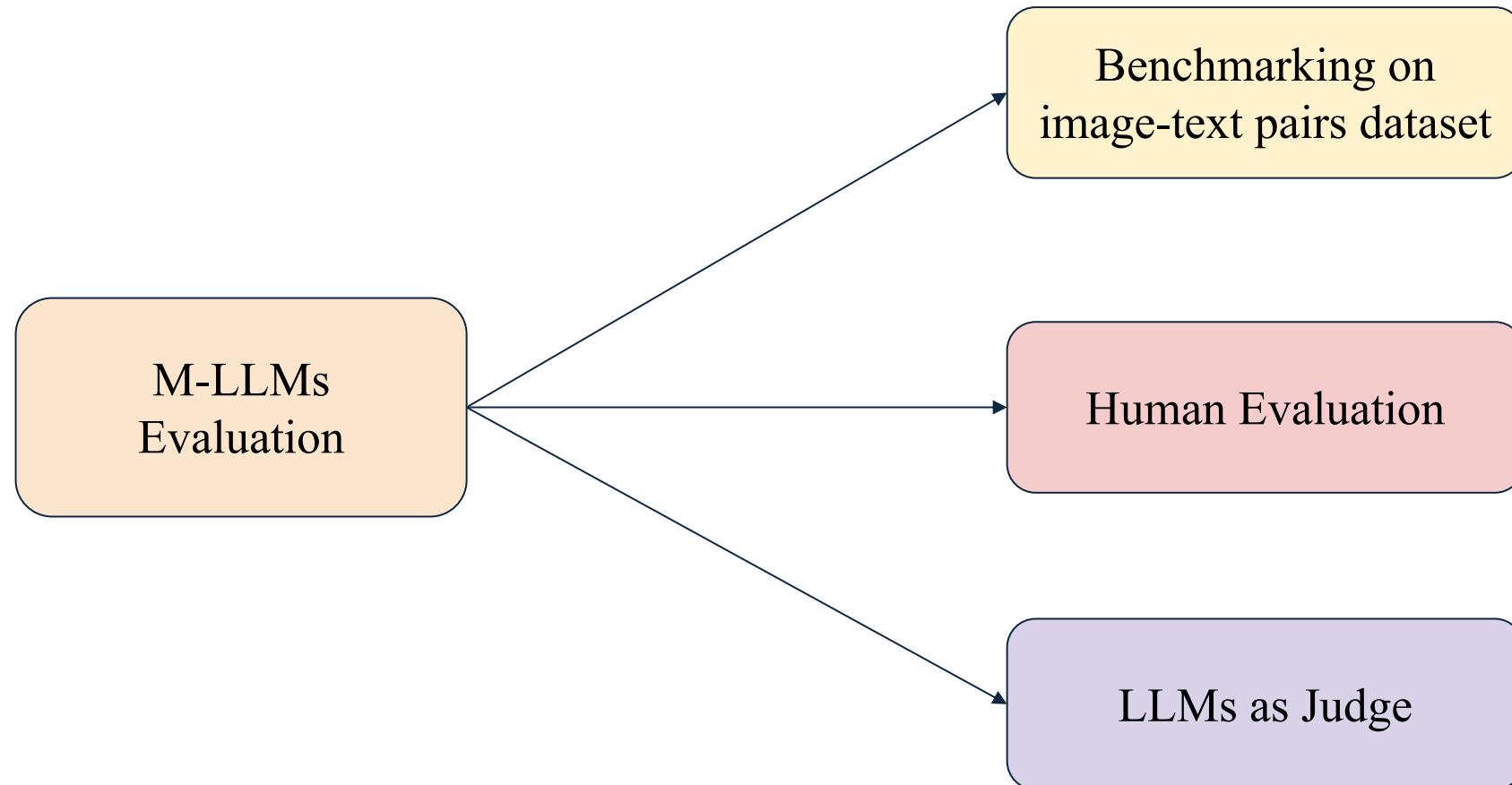


Human



Training LVLMs

❖ Evaluation LVLMs



Training LVLMs

❖ Evaluation LVLMs



Instruction: Complete the below task
Input: Describe the image

Our model

GPT-4



Scoring using GPT-4
between two responses
[1-10]

Training LVLMs

❖ Evaluation using LVLMs: A case study



Image

What kind of flowers are in the vase?

Question

VQA Model

Answer

tulips

Explanation

They are yellow and red with petals shaped in an oval.

Training LVLMs

❖ Evaluation using LVLMs: A case study

VQA-X



ViVQA-X

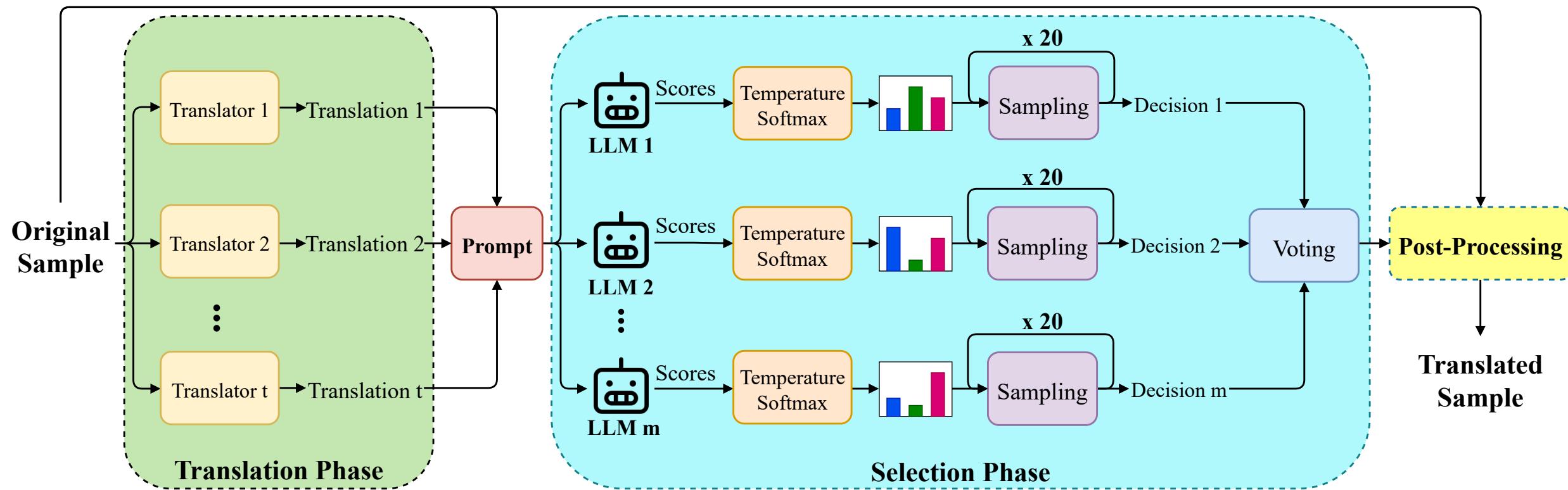


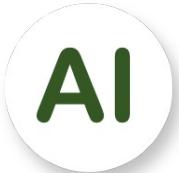
- **Question:** What sport is he doing?
- **Answers:** [surfing, surfing ...]
- **Explanations:**
 - the person is standing on a board in the ocean
 - he is standing on a board in the water on a small wave
 - he is on top of waves

- **Question:** Anh ấy đang chơi môn thể thao gì?
- **Answer:** lướt sóng
- **Explanations:**
 - người đó đang đứng trên một tấm ván trên biển
 - anh ấy đang đứng trên một tấm ván trên mặt nước trên một con sóng nhỏ
 - anh ấy đang ở trên đỉnh sóng

Training LVLMs

❖ Evaluation using LVLMs: A case study





AI

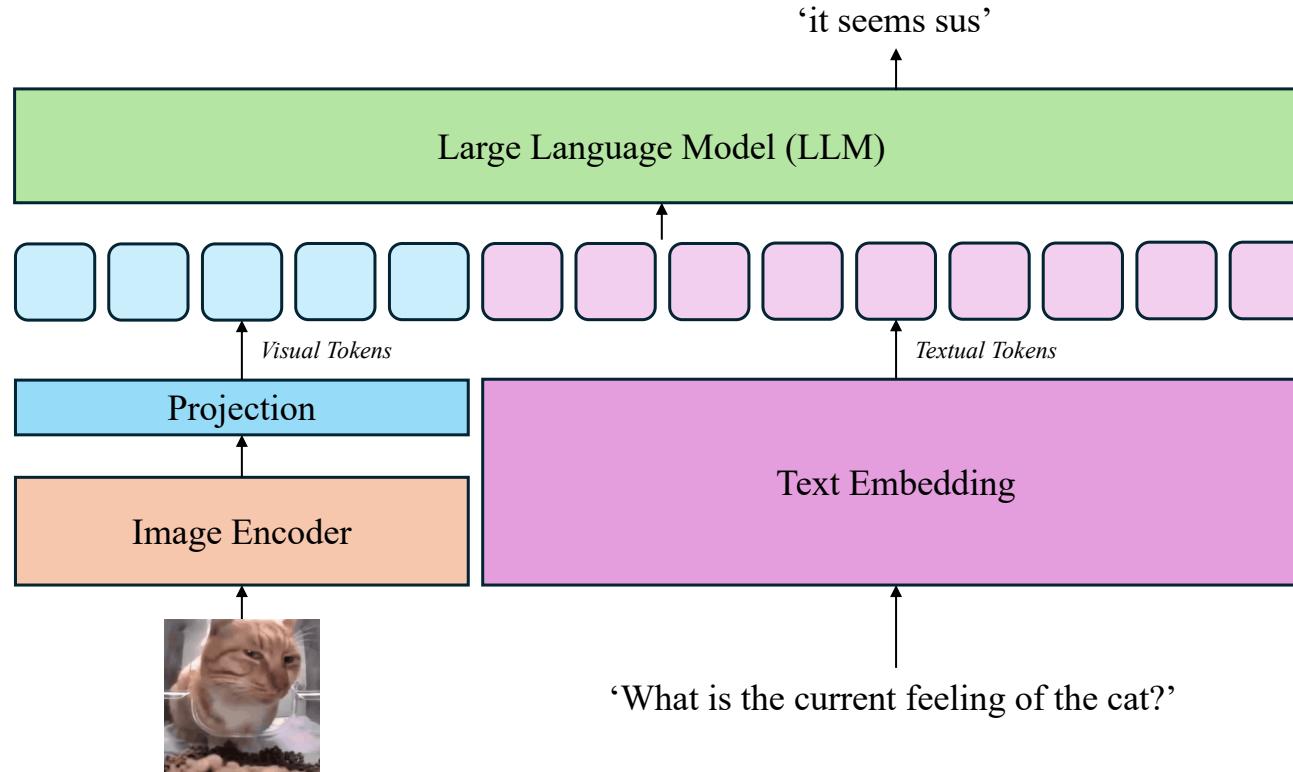
AI VIET NAM
@aivietnam.edu.vn

QUIZ

Summarization and Q&A

Summarization and Q&A

Output:



In this study session, we have discussed about:

1. Introduction to Multimodal LLMs and Large Vision Language Models (LVLMs).
2. In-context Learning to solve vision tasks using LVLMs.
3. Training techniques for LVLMs.

Summarization and Q&A

