Multimodal Models

Yushi Hu

yushi-hu.github.io/

Slides adapted from Prof. Yejin Choi's CSE 517 winter 2024 slides

MultiModal Systems

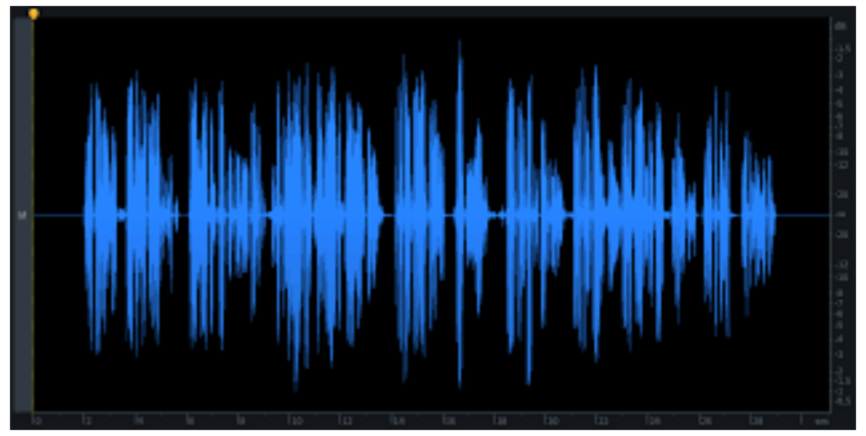
- **Multimodal AI**: System that integrates various data types and sensory inputs (images, videos, audio, other sensory information) to create a unified representation or understanding.

A person throwing a frisbee.

Text

Image

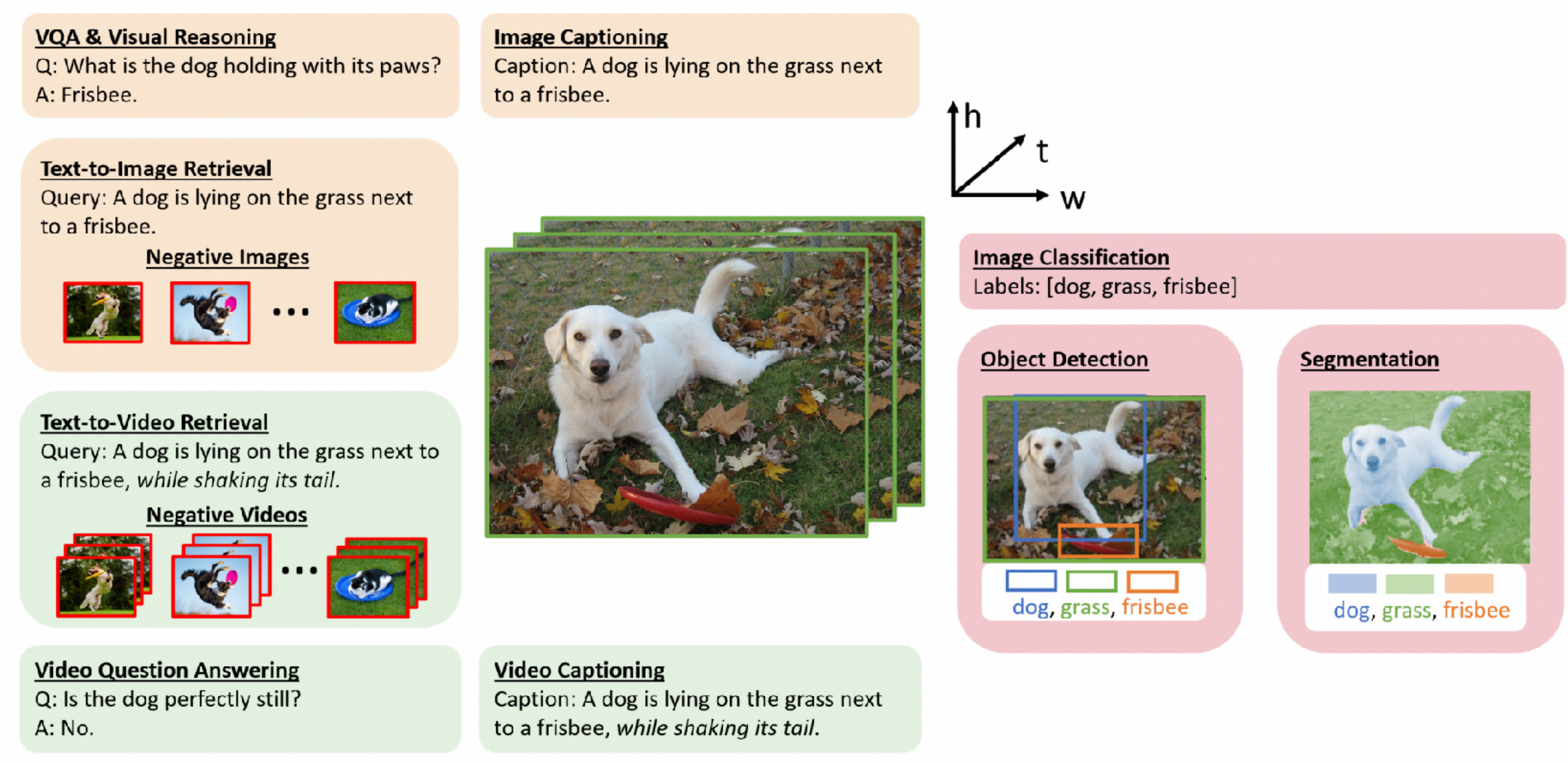
Video



Audio

- This lecture: will focus on **image** & **text** only.

Examples of Multimodal Tasks



Multimodal Language Models

How to train these models?

User: Can you explain this meme?

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

GPT-4: 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.

Multimodal Learning (for Image & Text)

Image & Text Alignment

A person throwing
a frisbee.

Image + Text Understanding

What is the object
being thrown?

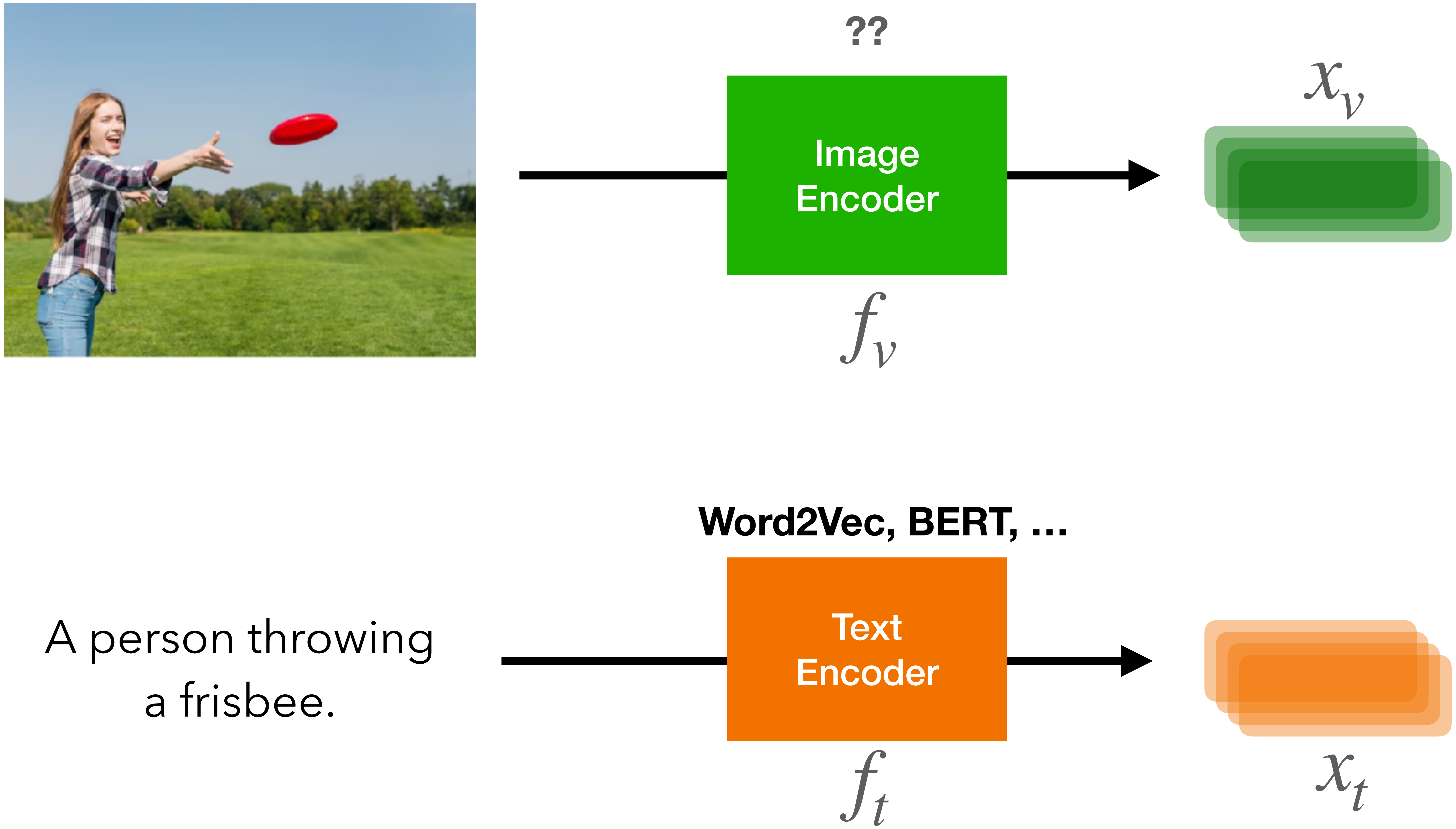
A frisbee

Text to Image Generation

A person throwing
a frisbee.

Note: For simplicity, we will cover image and text as the two modalities.

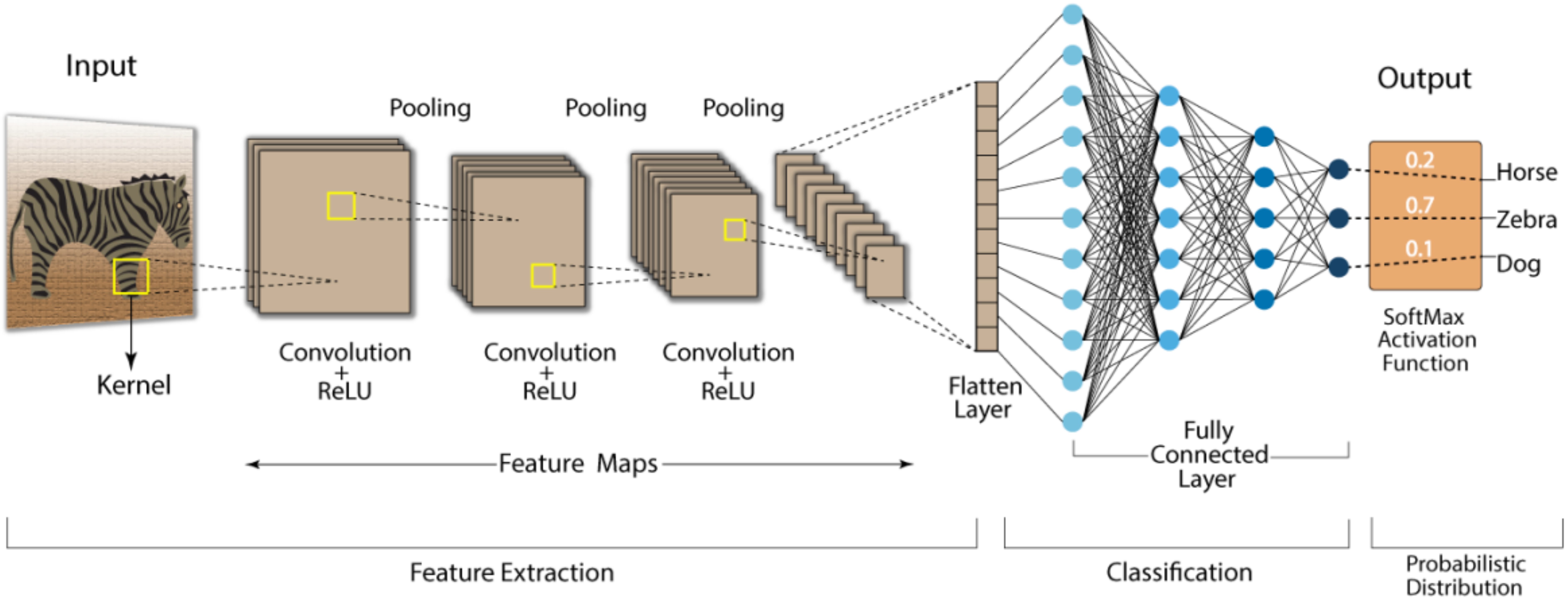
Steps of Image-Text Alignment



- **Step1:** Encode different modalities into shared embeddings.
- **Step2**: Bring modalities that encode same meaning into the same space.

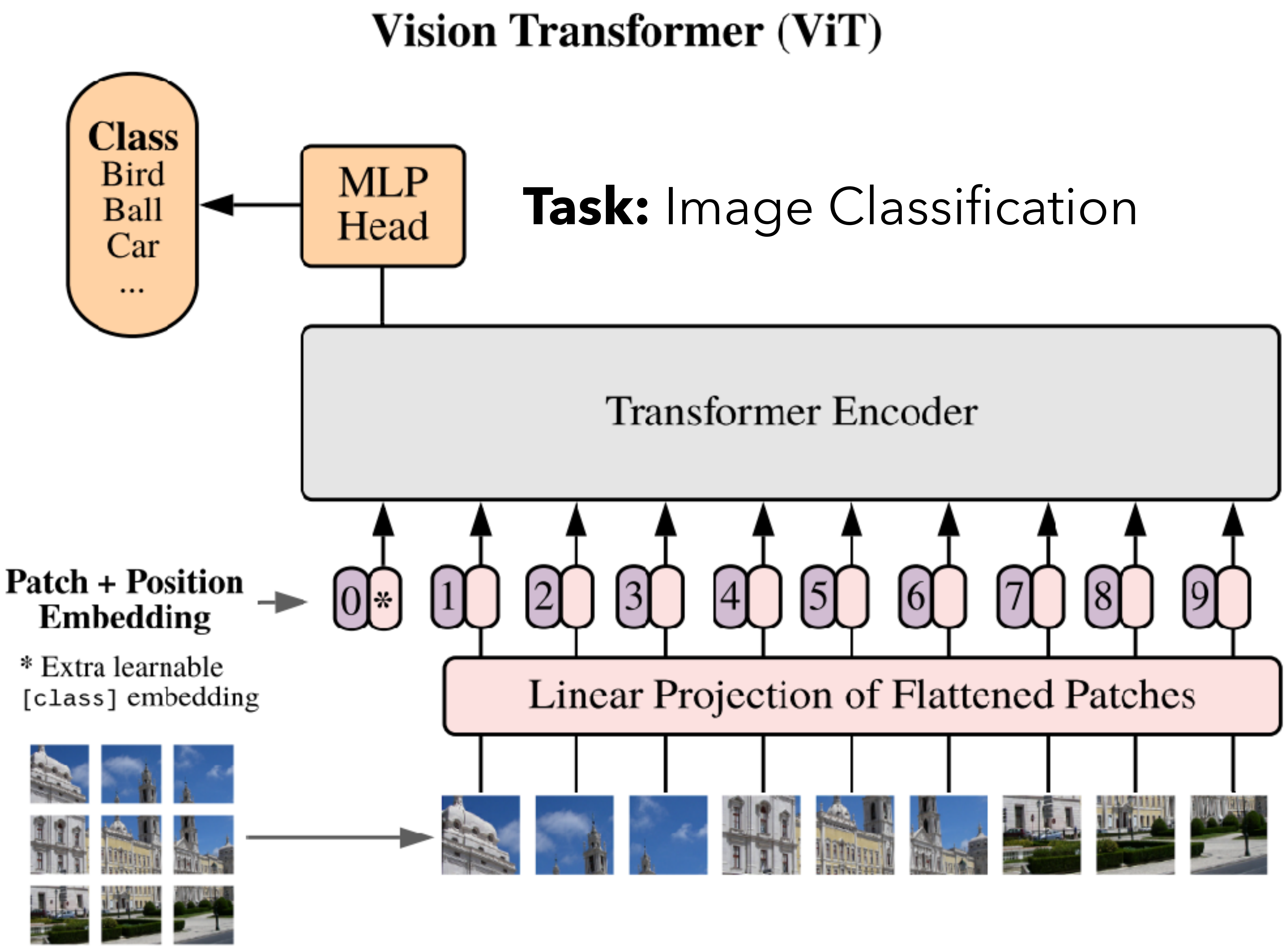
Vision Encoder: Convolutional Neural Networks

- **CNNs**: Extract features that encode spatial and temporal relationships in image with convolution operations.
  - **Pooling**: Reduce dimensionality of the convoluted features for efficient computation
- De facto model for Image Classification



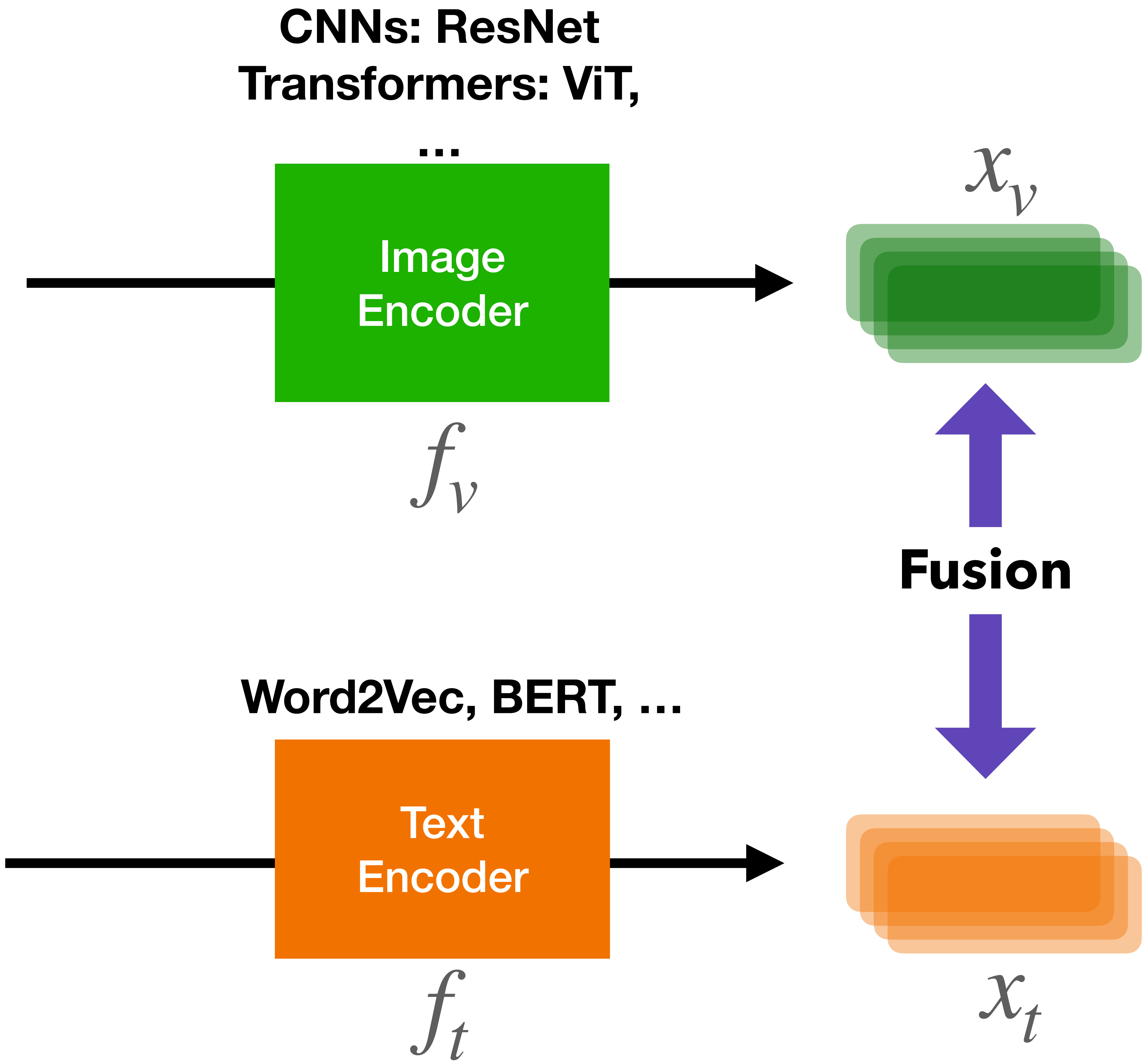
Vision Transformer (ViT): Image Encoding via Patch Tokens

- **Tokenize** images as sequence of "**patches**" of fixed size (e.g. 16 x16 px)
  - Resize images to same size to ensure same number of patches in training.
  - Image Size 224*224px = 14*14 patches
- Use the same transformer encoder architecture in NLP
  - Add [CLS] token for classification tasks.
  - Add positional embedding to be aware of location of patches.
- **Less image-specific inductive bias** than CNNs that encodes translation equivariance and locality.



Task: Image Classification

Steps of Image-Text Alignment



A person throwing a frisbee.

- **Step1:** Encode different modalities into shared embeddings.
- **Step2**: Bring modalities that encode same meaning into the same space.

Step2: Learning to Align Embeddings

$x_v \in \mathbb{R}^v$

Linear Projection

$z_v = W_v x_v^T + b_v^T \in \mathbb{R}^m$

- How to define the **loss function**?

A person throwing a frisbee.

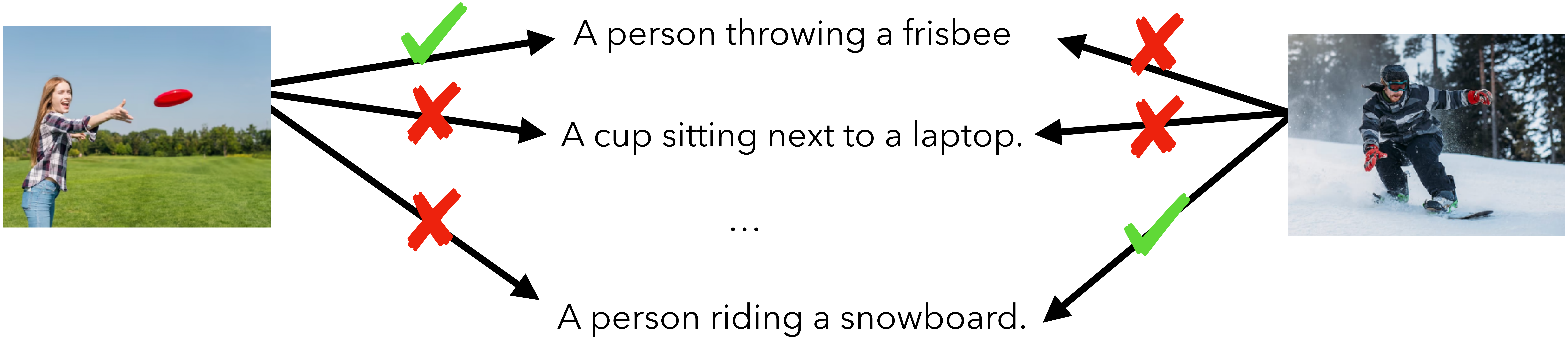
$x_t \in \mathbb{R}^t$

Linear Projection

$z_t = W_t x_t^T + b_t^T \in \mathbb{R}^m$

Contrastive Learning

- **Contrastive Learning**: learn the shared embedding by **contrasting** **positive** and **negative** pairs of instances
  - **Positives**: matched image-text pairs
  - **Negatives**: image-text from mismatched instances
- **Idea: Positive** instances should be closer together in a learned embedding space, while **Negatives** should be farther apart.



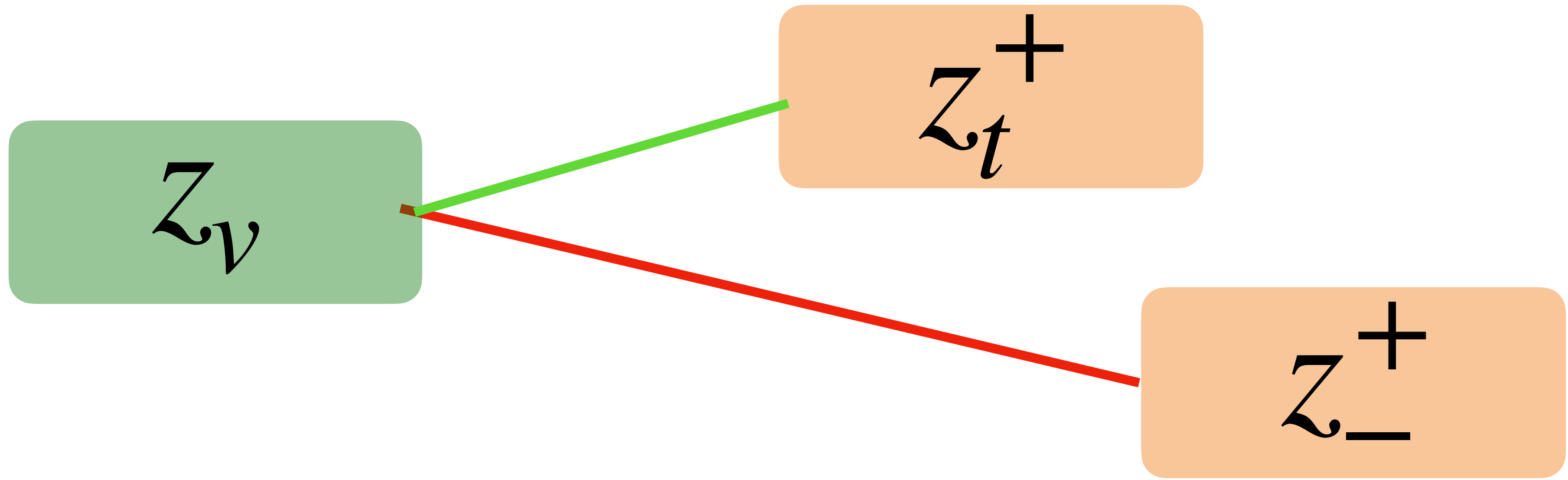
Contrastive Learning

- Adjust similarity of learned embeddings with a distance metric.
  - Euclidean Distance
  - Cosine Similarity $\cos(u, v) = \dfrac{u \cdot v}{||u||_2 ||v||_2}$

Why preferred over Euclidean Distance?

- $\text{sim}(z_v, z_t^+) >> \text{sim}(z_v, z_{-}^{+})$

A person throwing a frisbee



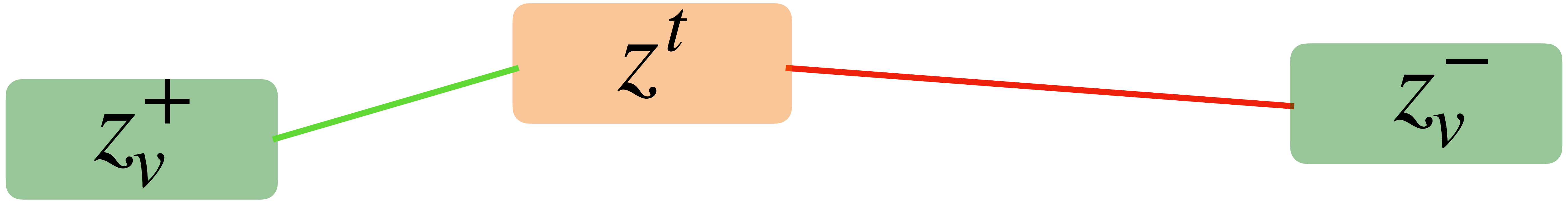
A person riding a snowboard.

Contrastive Learning

- Adjust similarity of learned embeddings with a distance metric.
  - Euclidean Distance
  - Cosine Similarity $\cos(u, v) = \dfrac{u \cdot v}{||u||_2 ||v||_2}$

- $\text{sim}(z_v, z_t^{+}) >> \text{sim}(z_v, z_{-}^{+})$ + $\text{sim}(z_v^{+}, z_t) >> \text{sim}(z_v^{-}, z_t)$

A person throwing a frisbee



Contrastive Learning

margin parameter: min distance b.w. positive and negatives

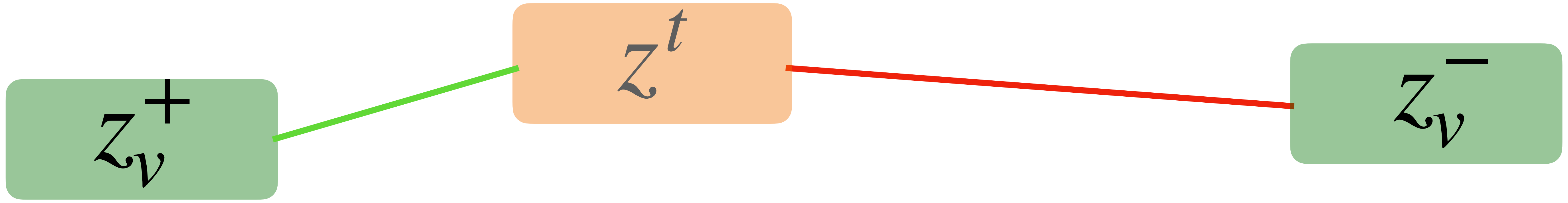
- Adjust similarity of learned embeddings with a distance metric.
  - Euclidean Distance
  - Cosine Similarity

Triplet Loss

$$\max(0, \text{sim}(z_v, z_t^{+}) - \text{sim}(z_v, z_{-}^{+}) + m) +$$
$$\max(0, \text{sim}(z_v^{+}, z_t) - \text{sim}(z_v^{-}, z_t) + m)$$

- $\text{sim}(z_v, z_t^{+}) >> \text{sim}(z_v, z_{-}^{+}) \ + \ \text{sim}(z_v^{+}, z_t) >> \text{sim}(z_v^{-}, z_t)$

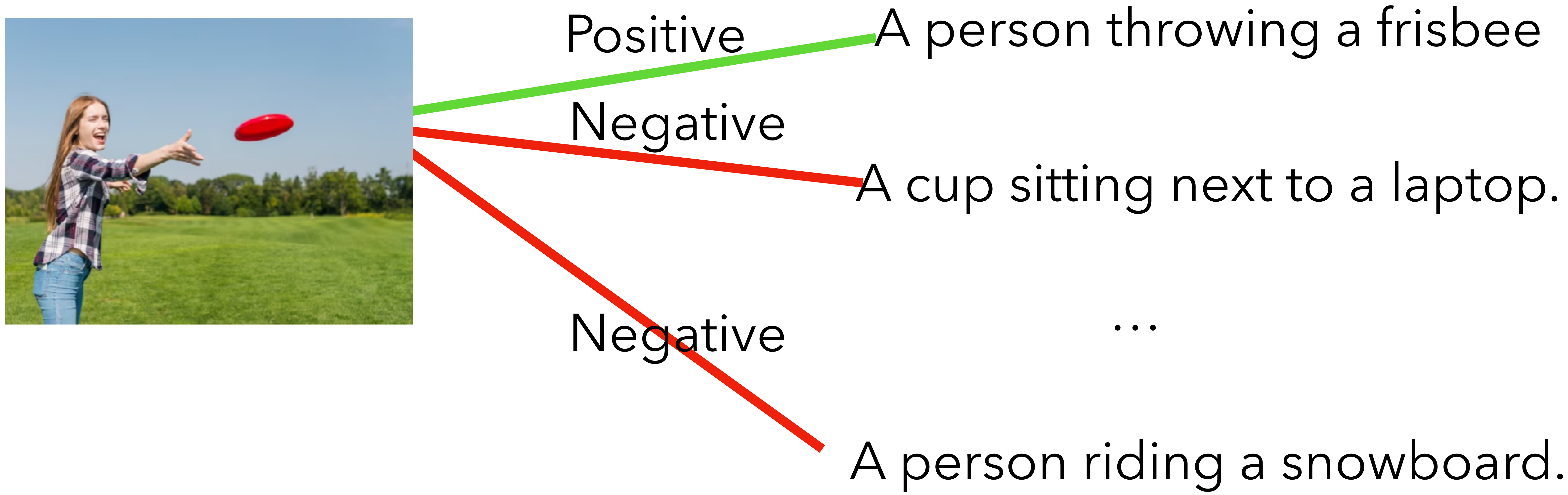
A person throwing a frisbee



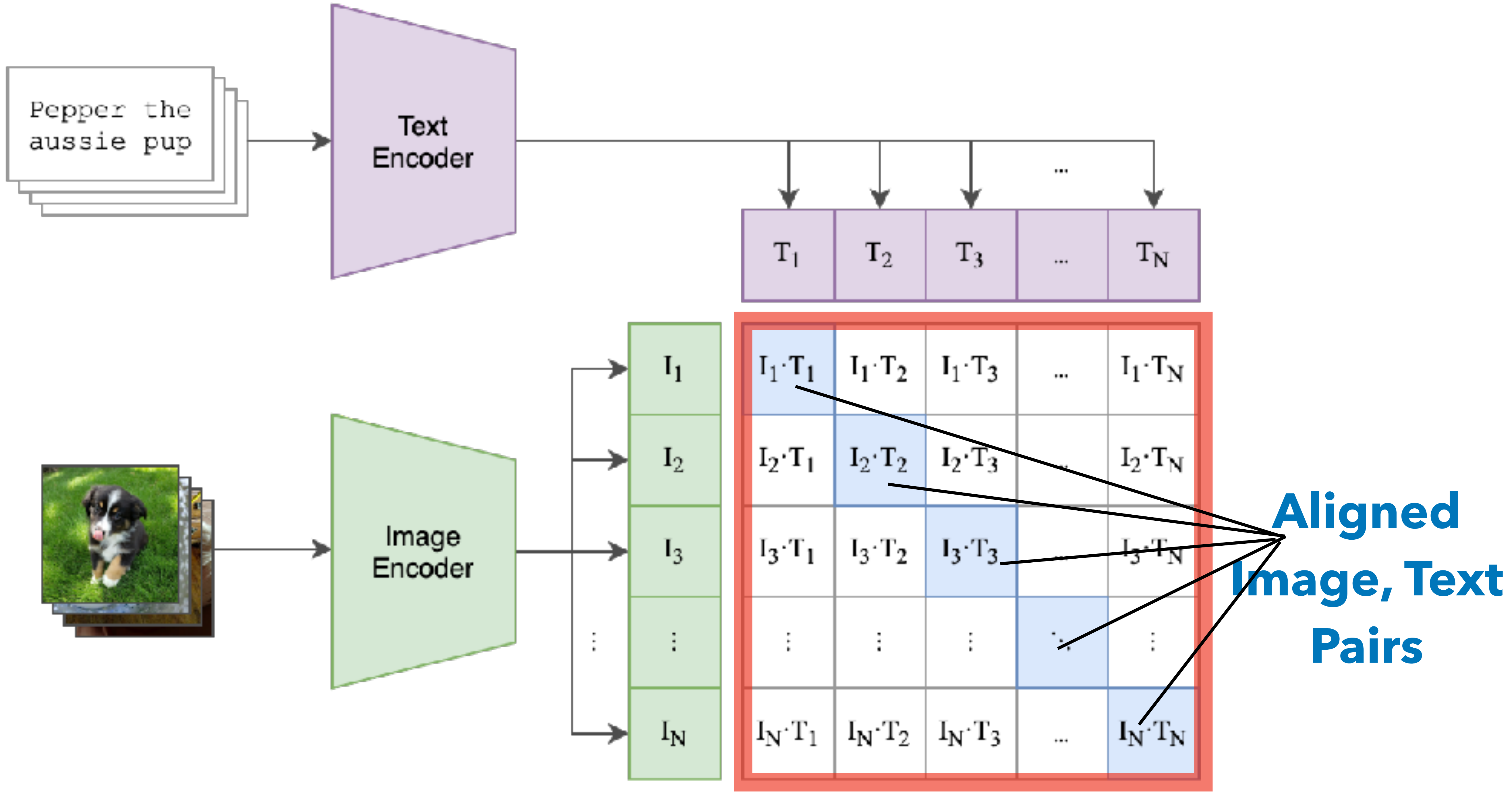
https://www.v7labs.com/blog/triplet-loss#:~:text=Triplet%20loss%20is%20a%20way,a%20dissimilar%20item%20(negative).

A Different View of Contrastive Learning

- What does this look like?
- Classification over distance embedding!



CLIP: Contrastive Language-Image Pre-Training (OpenAI, 2021)



Objective: given a batch of N (image, text) pairs, predict which of the N × N possible (image, text) pairings across a batch actually occurred.

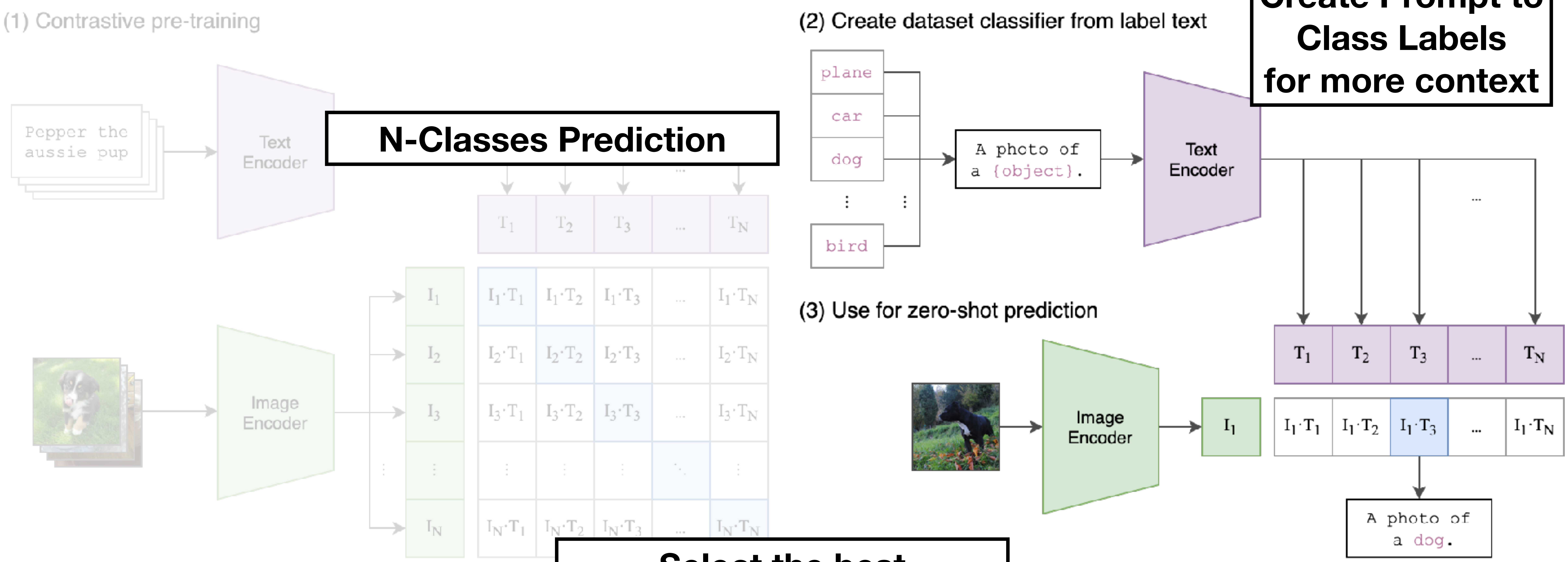
Minimize InfoNCE Loss

$$L_{NCE} = -\log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_j / \tau)}{\sum_{k=0}^{N} \exp(\mathbf{z}_i \cdot \mathbf{z}_k / \tau)}$$

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Use the [CLS] token for transformers

CLIP: Contrastive Language-Image Pre-Training



Create Prompt to Class Labels for more context

N-Classes Prediction

Select the best text prompt that gives the highest similarity.

➡ **Enables Open Vocabulary Classification class labels.**

Image-Text Training Dataset

- Previous Image-Text Pre-Training Dataset
  - Leverage filtered, carefully annotated dataset for academic research
  - 10M was considered as "large-scale" pre-training

	COCO	VG	SBU	CC3M	Total
#Images	113K	108K	875K	3.1M	4.2M
#Captions	567K	5.4M	875K	3.1M	10M

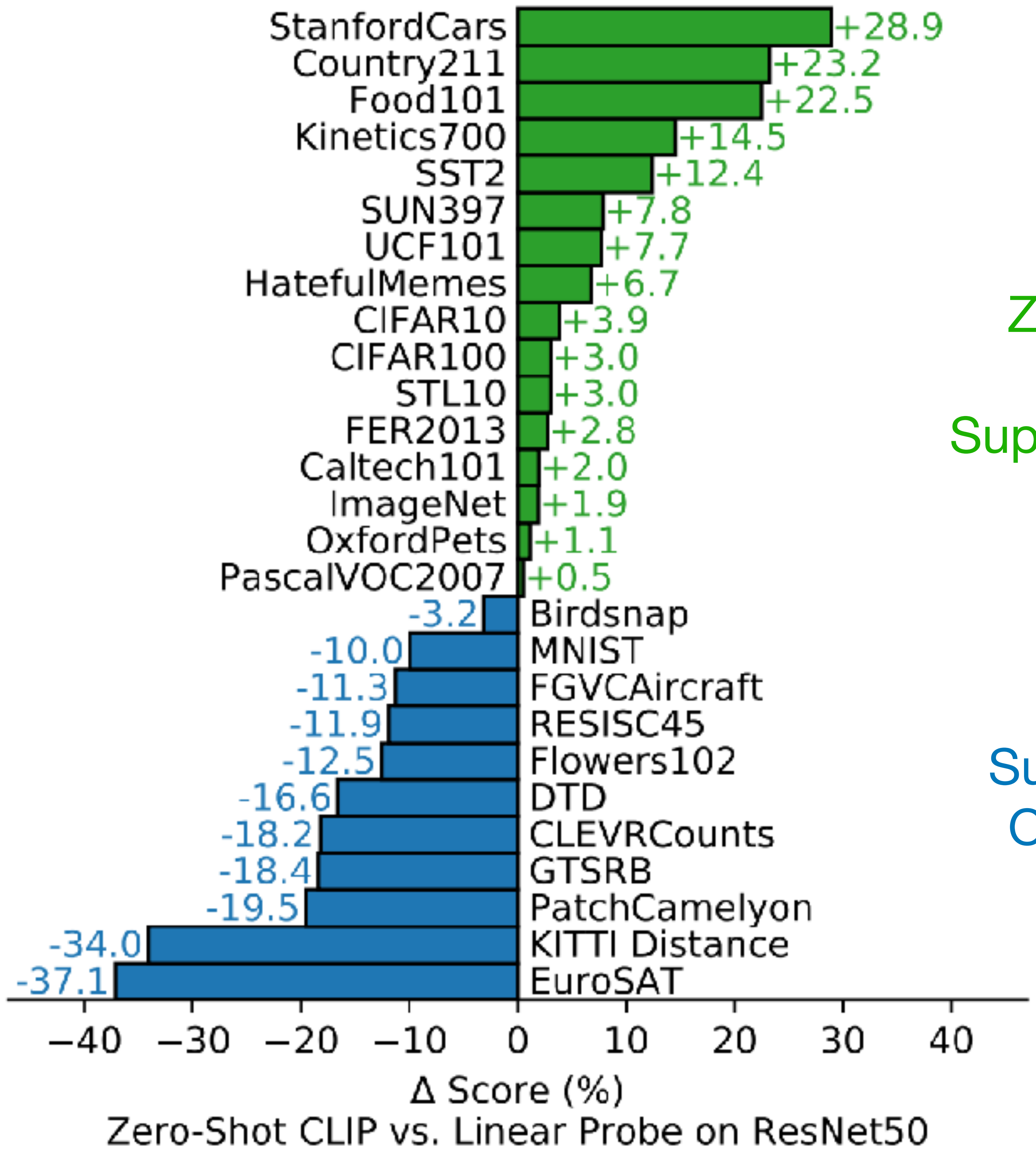
Table 3.2: Statistics of the pre-training datasets used in a typical academic setting.

Image-Text Training Dataset

- Previous Image-Text Pre-Training Dataset
  - Leverage filtered, carefully annotated dataset for academic research
  - 10M was considered as "large-scale" pre-training
- **CLIP: 400M** Image-Text pairs crawled from web
  - Unfiltered, highly varied, and highly noisy data
  - Covers much more diverse concepts and images

Text Supervision Enables Strong Zero-Shot Performance in Vision Tasks

- Large-Scale Training on Noisy Image-Text Data -> Great Zero-Shot Performance
- **Zero-Shot CLIP** is **competitive with fully supervised** Resnet50 in Image Classification
  - *Linear Probe*: Train linear layer on top of fixed, pre-trained embeddings.

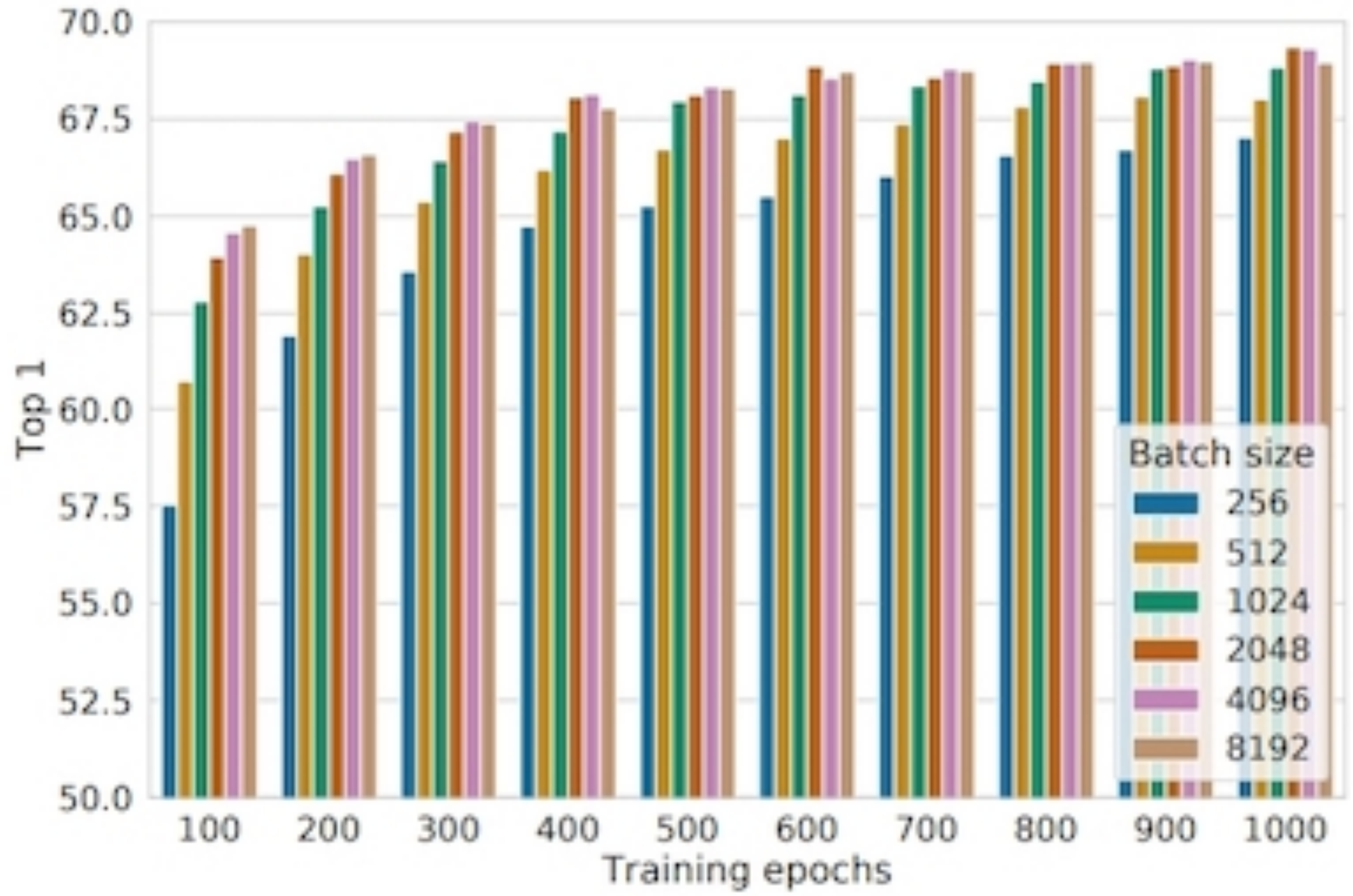


Zero-Shot CLIP Outperforms Supervised baseline.

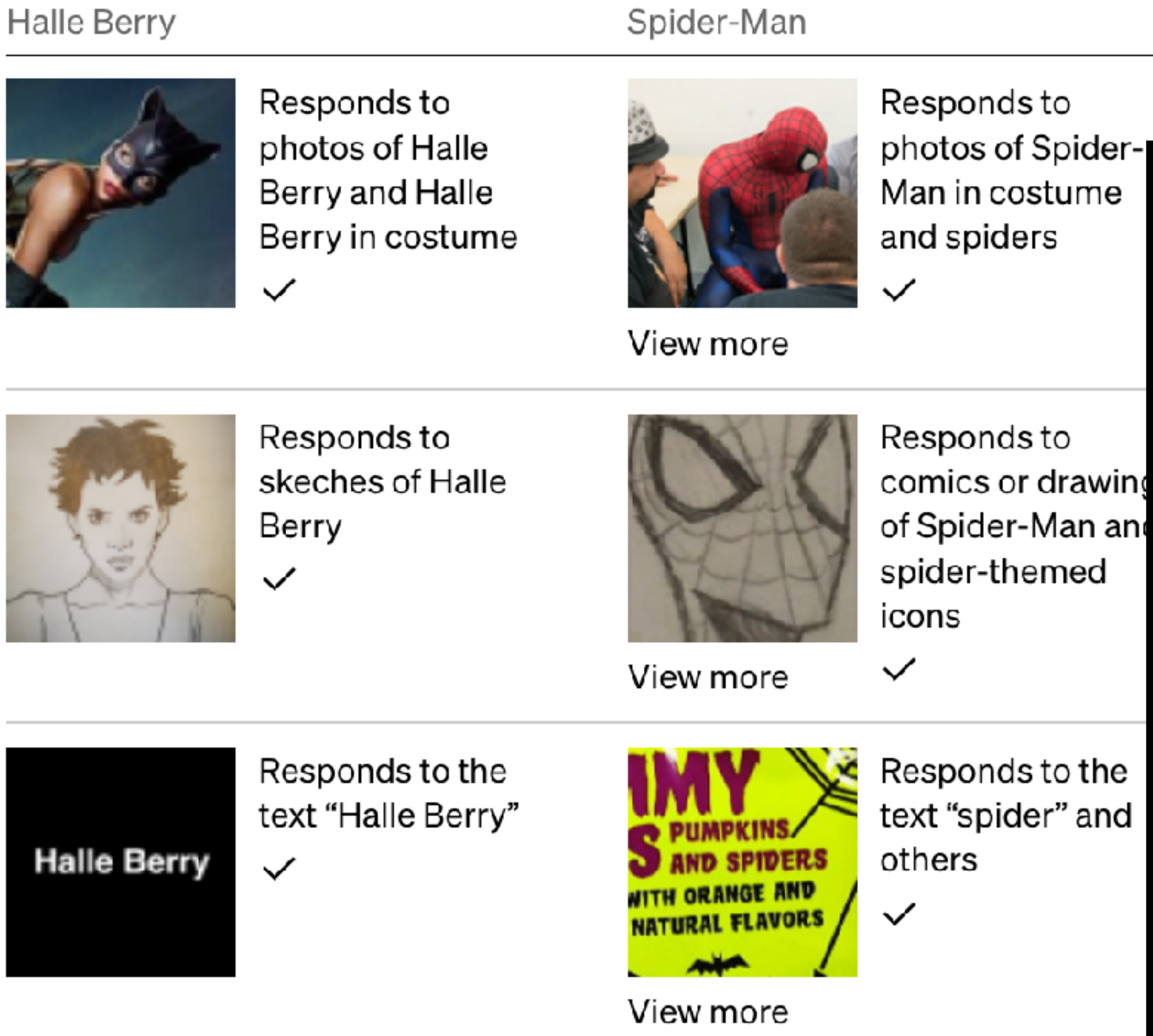
Supervised baseline Outperforms CLIP.

Why is CLIP so good?

- Learning **visual representation** with **language supervision**: learns visual concepts much more efficiently.
- Exploited Scalability benefits:
  - 256 GPUS + 4096 batch size with 2 weeks of training
  - Large batch size in Contrastive Learning
    - More negatives to compare against.
    - More challenging task to distinguish the negatives, requiring fine-grained visual recognition.



Understanding Multimodal Capabilites of CLIP



- Aligns images to **semantic concepts** thanks to **language supervision**, rather than just aligning texture and shapes.
- Case where multimodal learning was a big breakthrough for learning high-quality, unimodal representations (image)

Vision and Language Systems

Image & Text Alignment

A person throwing
a frisbee.

Image to Text Understanding

What is the object
being thrown?

A frisbee

Text to Image Generation

A person throwing
a frisbee.

Note: For simplicity, we will cover image and text as the two modalities.

CLIP for Visual Reasoning?

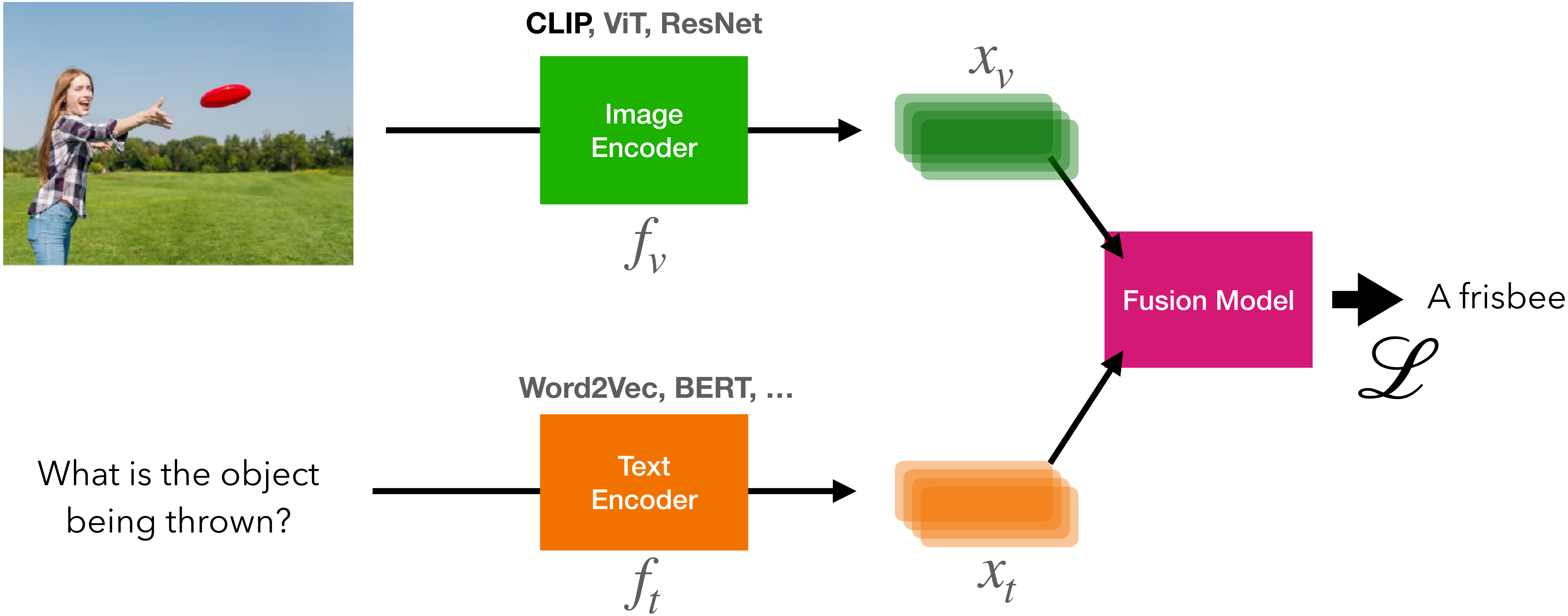
- Supports retrieval but not capable of generation
- **VQA Prompt**: *"question: [question text] answer: [answer text]"*
- Note: CLIP is trained to align images with alt-text captions
  - Not suitable for reasoning tasks such as question answering.

Model	VQA Question Type		
	yes/no	number	other
CLIP-Res50	0.037	0.057	0.0
CLIP-ViT-B_{PE}	0.019	0.0	0.0
CLIP-Res50$_{PE}$	0.055	0.057	0.0
CLIP-Res101$_{PE}$	0.260	0.0	0.0
CLIP-Res50x4$_{PE}$	0.446	0.118	0.034

Table 7: Zero-shot performance of CLIP on VQA v2.0 `mini-eval`, "PE" denotes we follow similar prompt engineering as suggested in CLIP paper.

Near Chance Performance

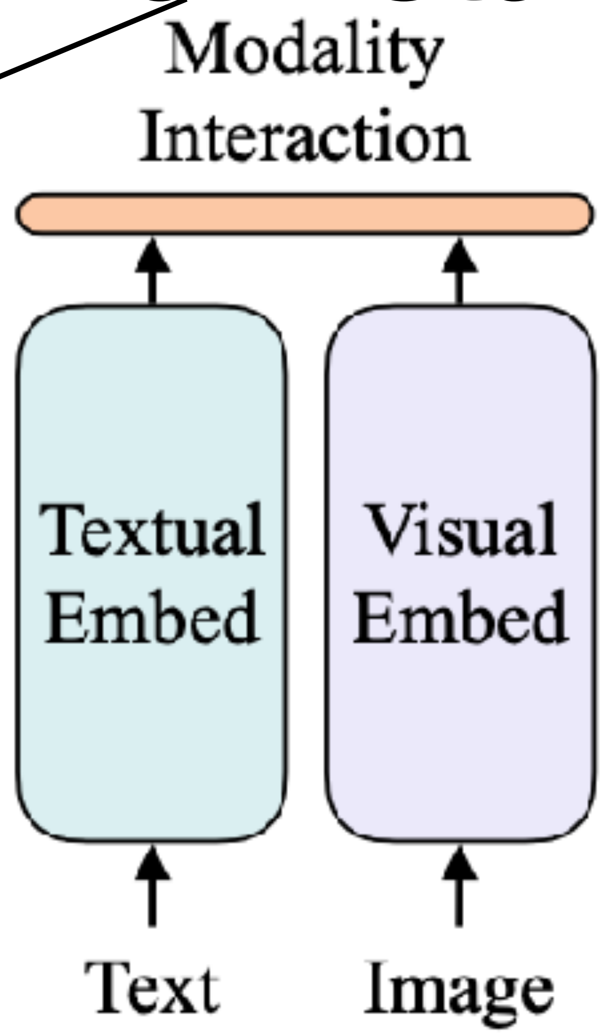
Image and Text Understanding



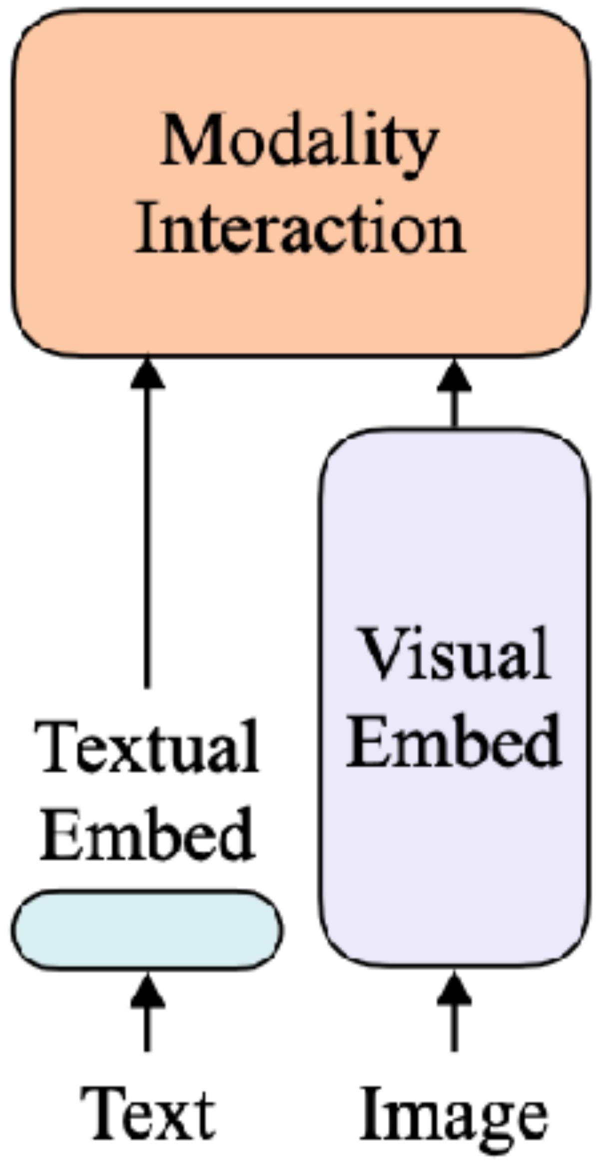
Embedding vs Fusion Trade Offs

Cosine Distance

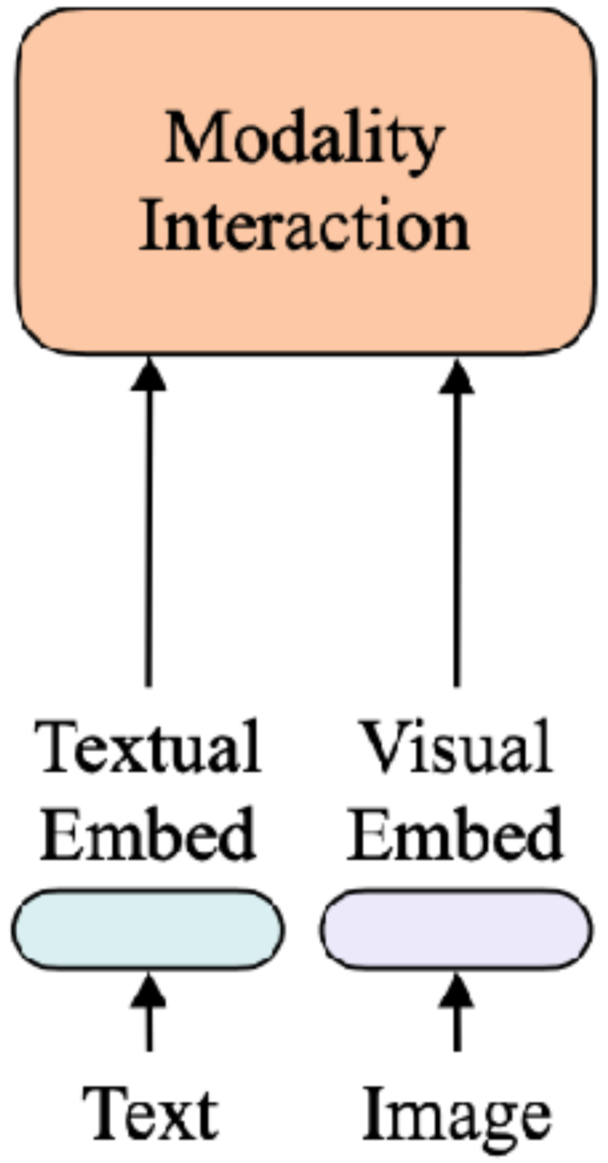
Enough for image-text matching



CLIP



Perhaps, need stronger fusion mechanism for complex reasoning tasks



Vision and Language Fusion

- Is there a good model that can efficiently encode interactions among the sequence?
- **Hint:** What models have been covered in this class?

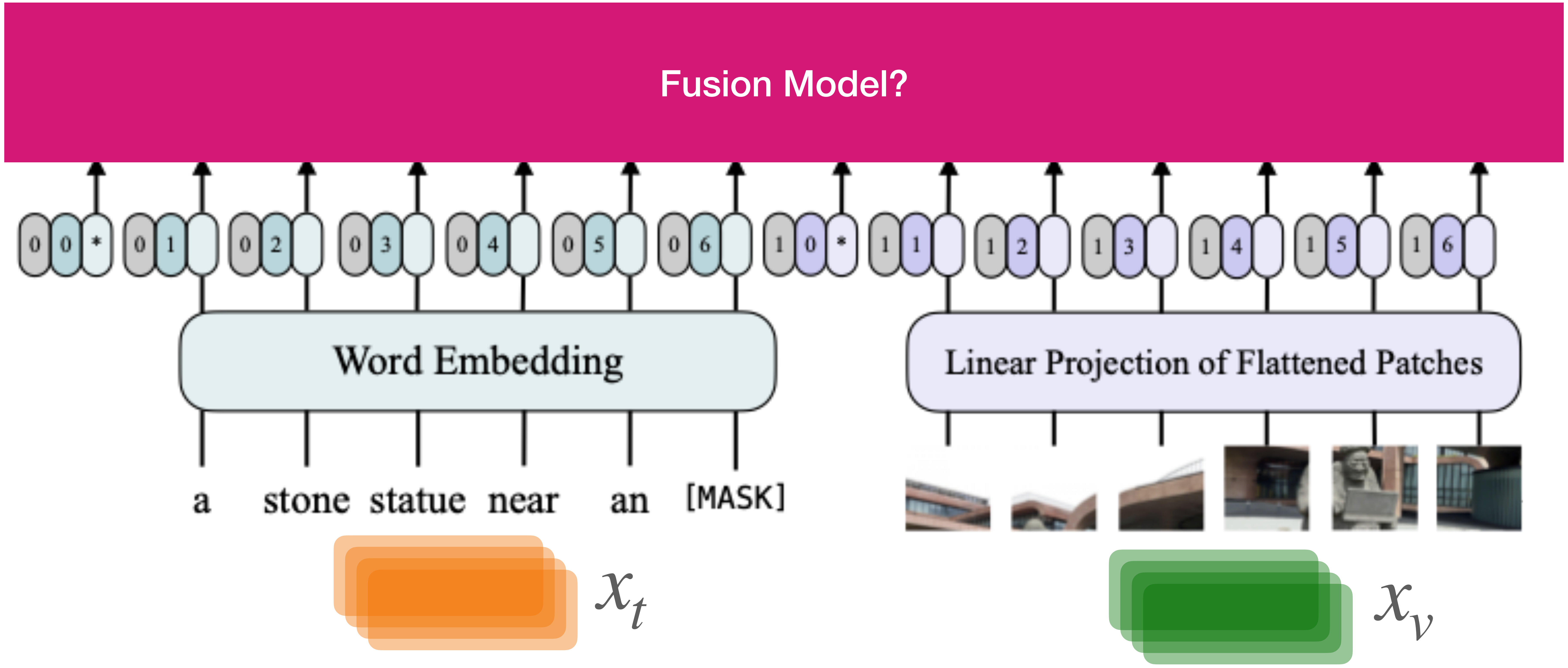


Image-to-Text Generative Models

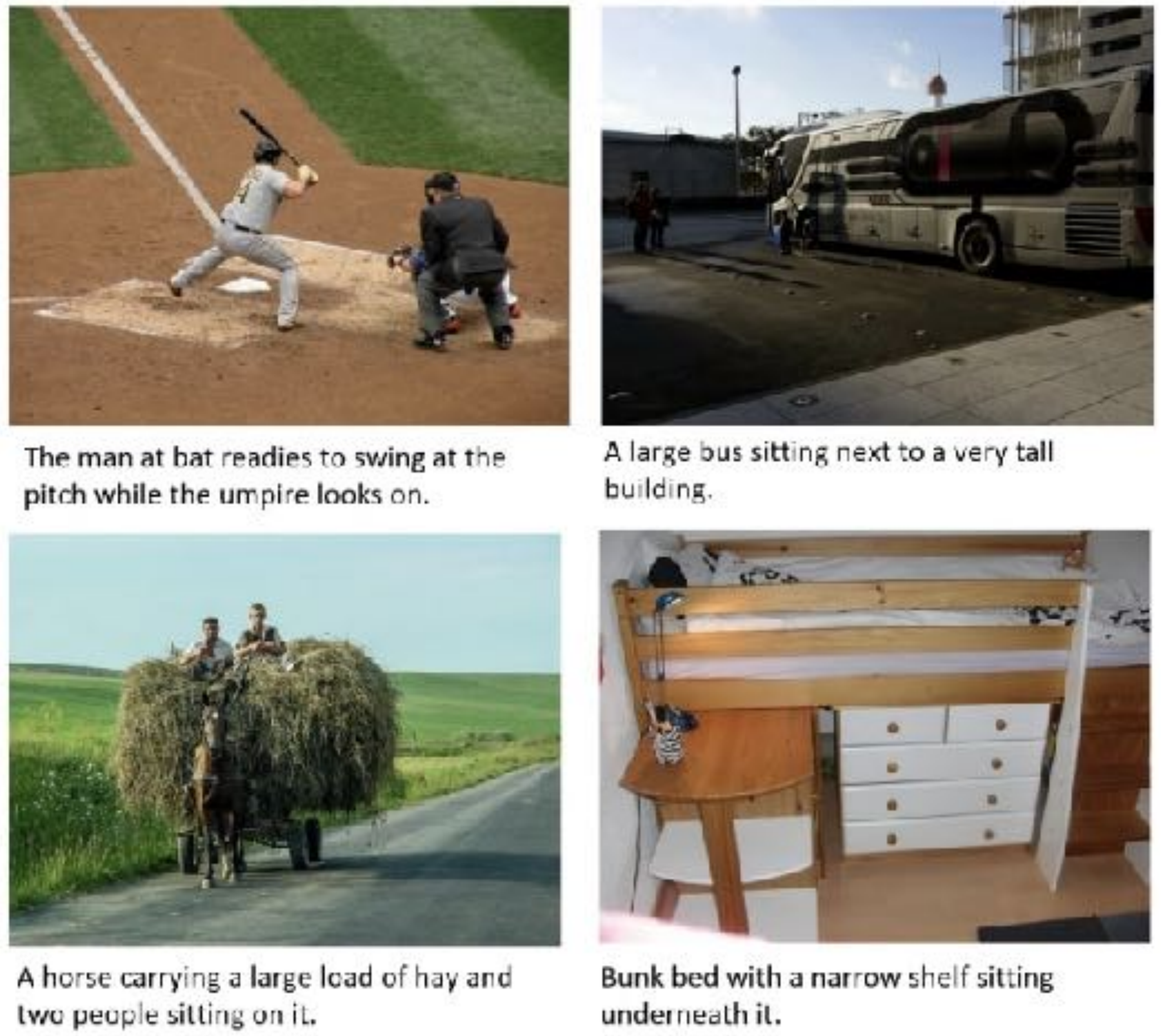
- Image Encoder – Trainable/Frozen
- Language Encoder/Decoder – Trainable/Frozen
- Connecting/Aligning Modules – Trainable

Vision Model | Aligning Model | Language Model

A woman throwing a frisbee

Frozen: Multimodal Few-Shot Learning with Frozen Language Models (Deepmind 2021)

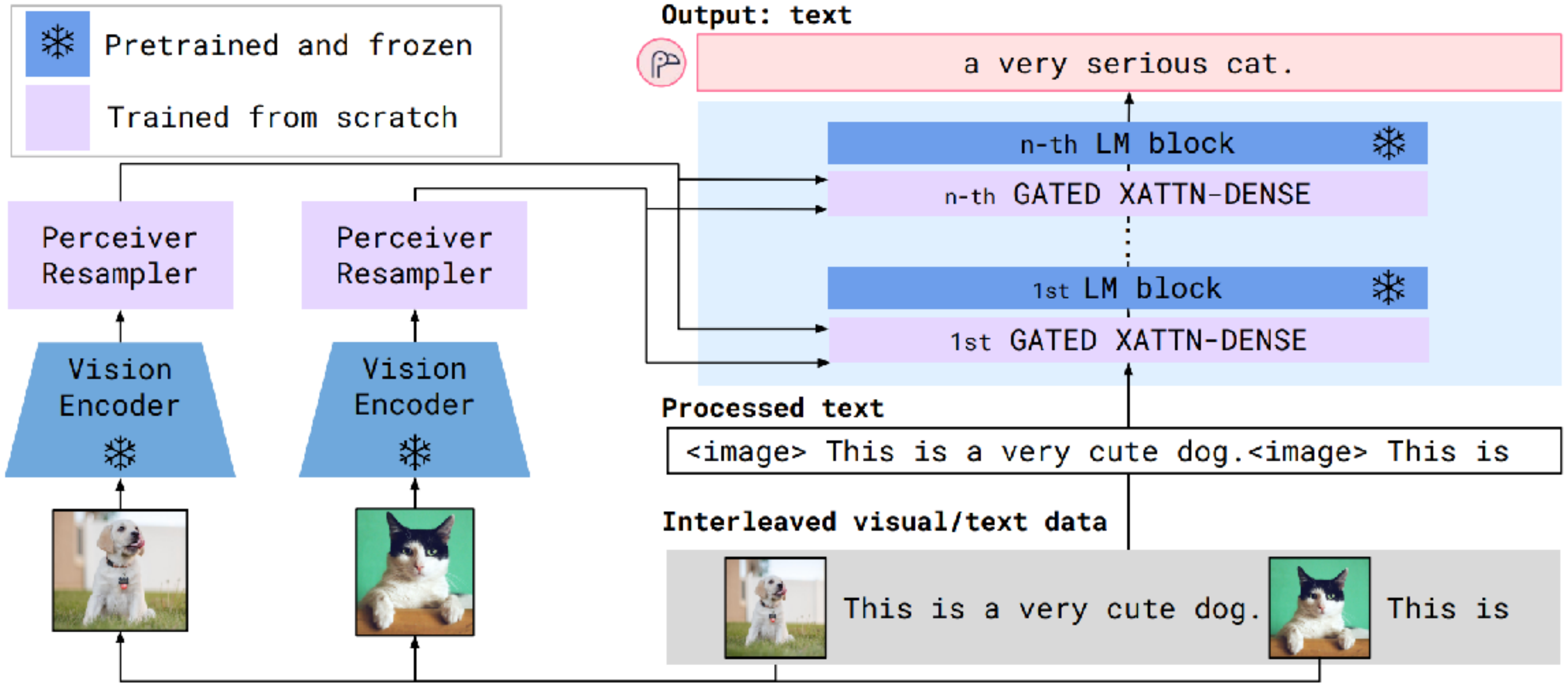
- Image Captioning: describe an image using a sentence.



Microsoft COCO, 2014

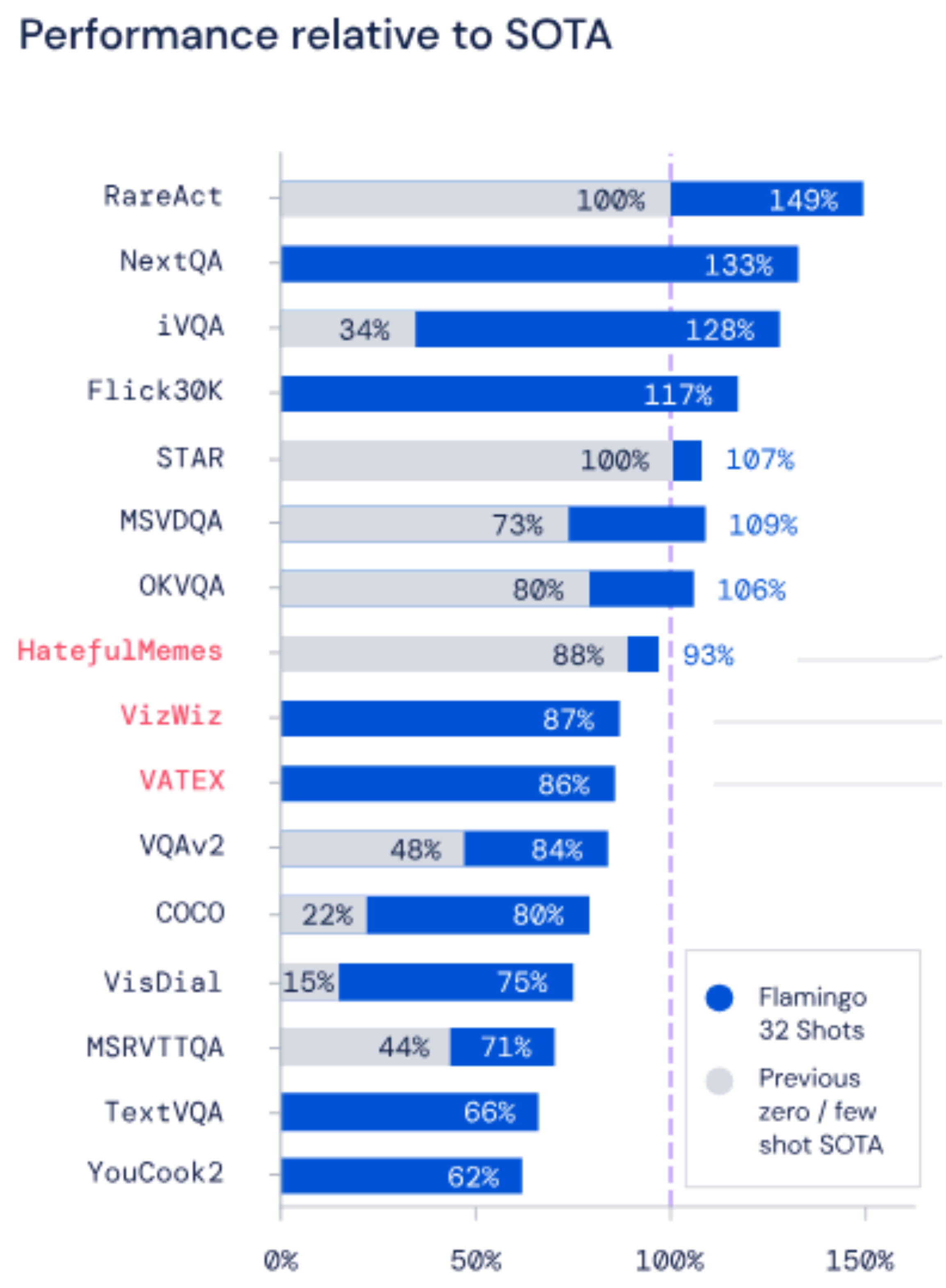
Frozen: Prefix Tuning of Image Embeddings (Deepmind, 2021)

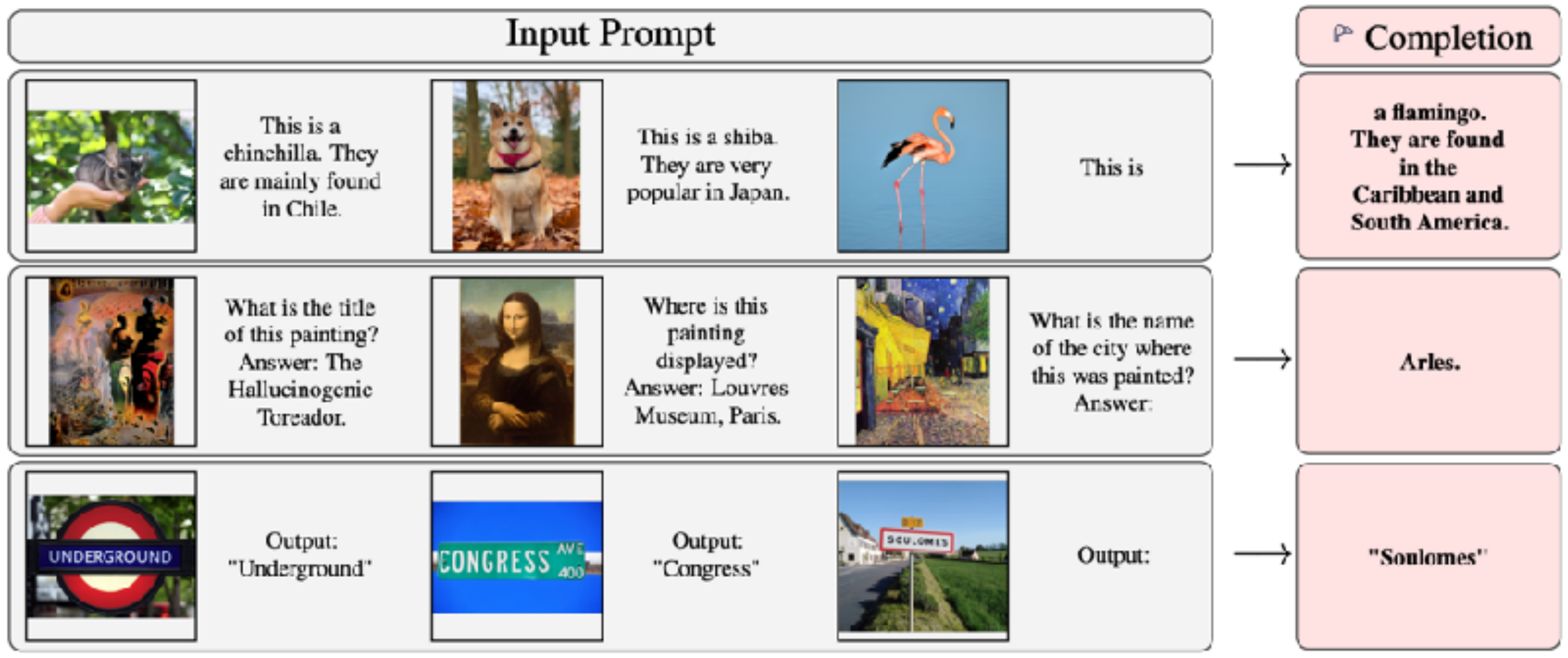
Flamingo: a Visual Language Model for Few-Shot Learning (Deepmind, 2022)

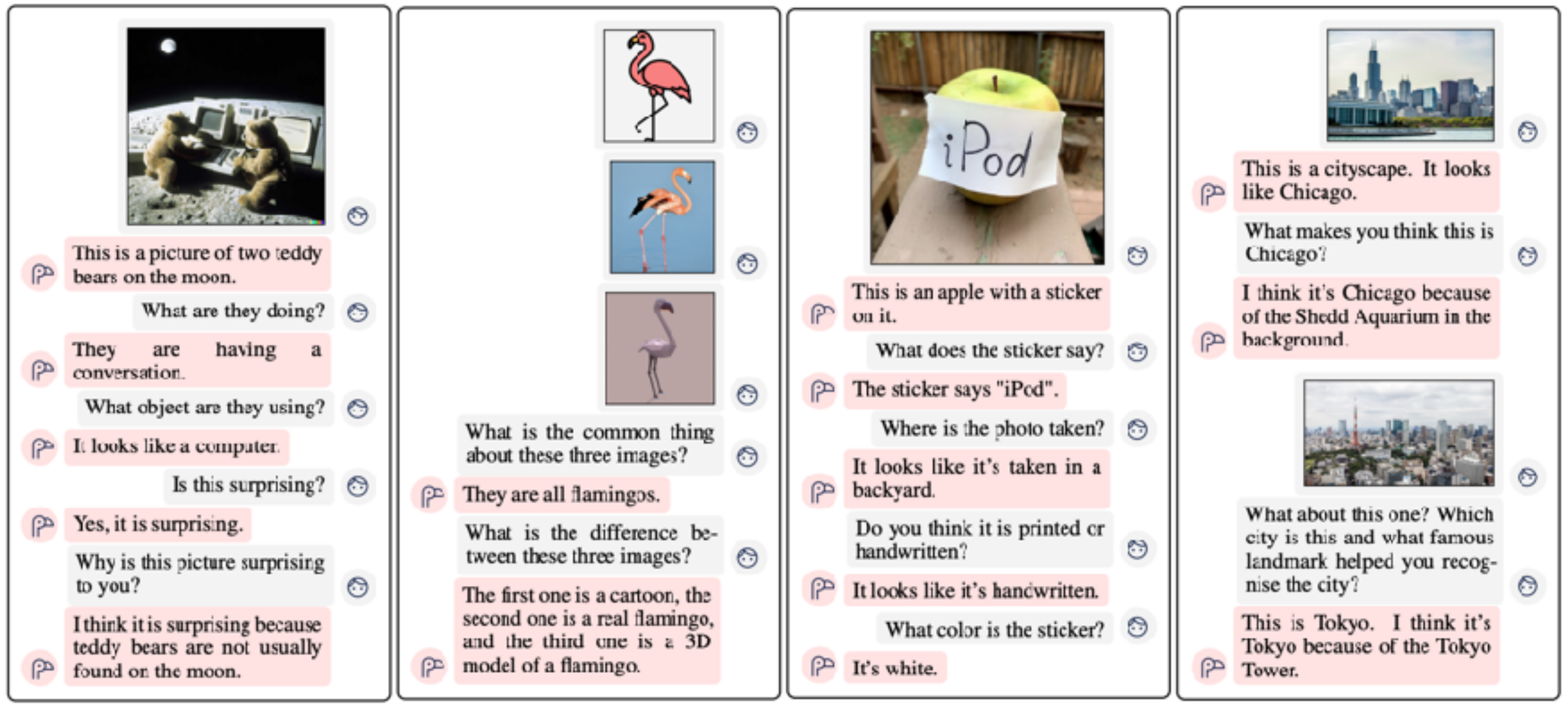


Flamingo: a Visual Language Model for Few-Shot Learning (Deepmind, 2022)

- Strong zero/few-shot performance







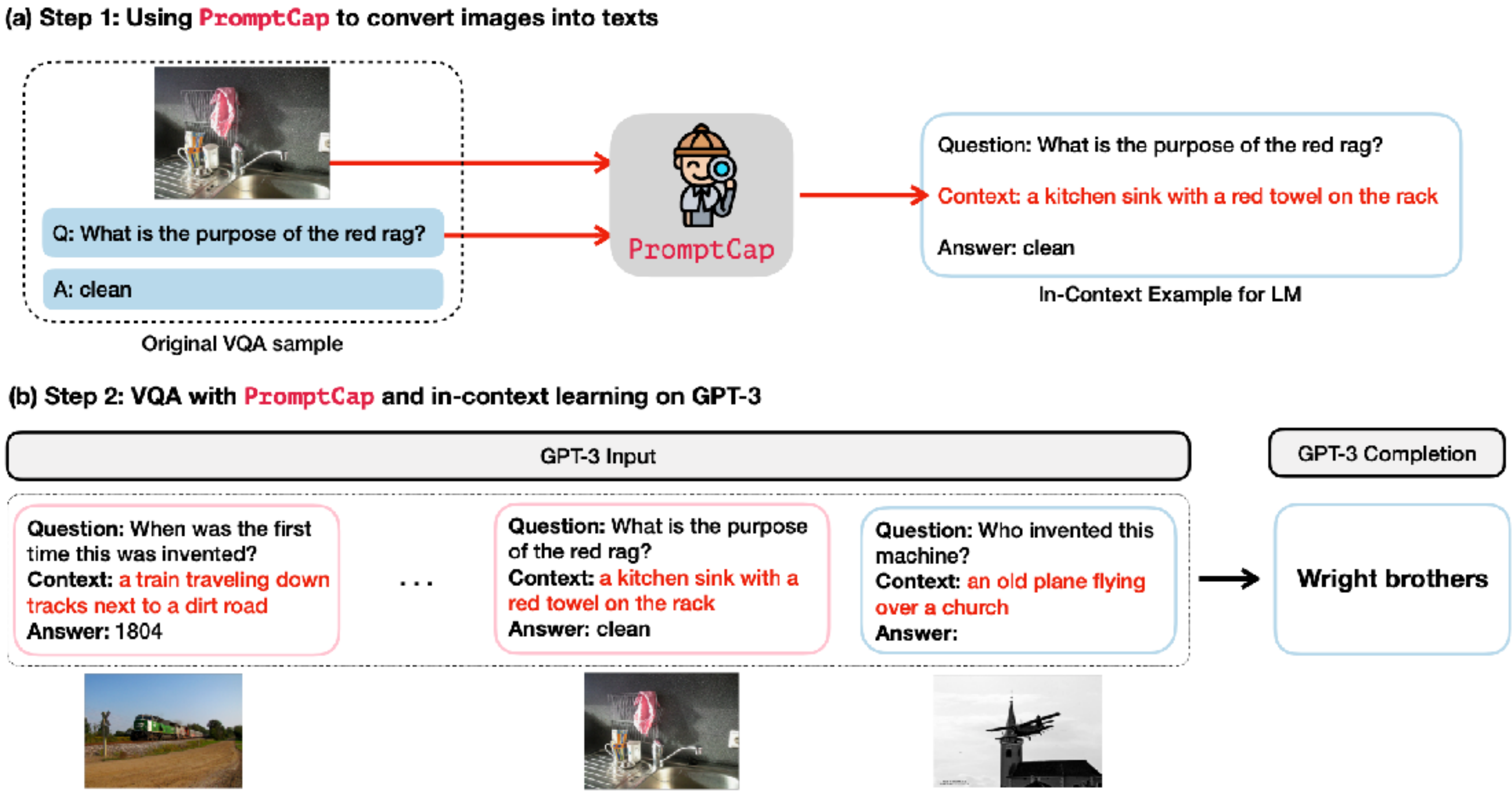
Trends of VL Models

- Race of Scaling Model Size / Dataset / # of Tasks?

Model	Model Size				PT dataset size	PT Tasks
	Image Enc.	Text Enc.†	Fusion†	Total		
CLIP ViT-L/14 (Radford et al., 2021)	302M	123M	0	425M	400M	ITC
ALIGN (Jia et al., 2021)	480M	340M	0	820M	1.8B	ITC
Florence (Yuan et al., 2021)	637M	256M	0	893M	900M	ITC
SimVLM-huge (Wang et al., 2022k)	300M	39M	600M	939M	1.8B	PrefixLM
METER-huge (Dou et al., 2022b)	637M	125M	220M	982M	900M+20M[1]	MLM+ITM
LEMON (Hu et al., 2022)	147M[2]	39M	636M	822M	200M	MLM
Flamingo (Alayrac et al., 2022)	200M	70B	10B	80.2B	2.1B+27M[3]	LM
GIT (Wang et al., 2022d)	637M	40M	70M	747M	800M	LM
GIT2 (Wang et al., 2022d)	4.8B	40M	260M	5.1B	12.9B	LM
CoCa (Yu et al., 2022a)	1B	477M	623M	2.1B	1.8B+3B[4]	ITC+LM
BEiT-3 (Wang et al., 2022g)	692M[5]	692M[5]	52M[5]	1.9B	21M+14M[6]	MIM+MLM +MVLM
PaLI (Chen et al., 2022e)	3.9B	40M	13B	16.9B	1.6B	LM+VQA[7] +OCR+OD

Side note: Language as a common interface?

- Image captioning is all we need?



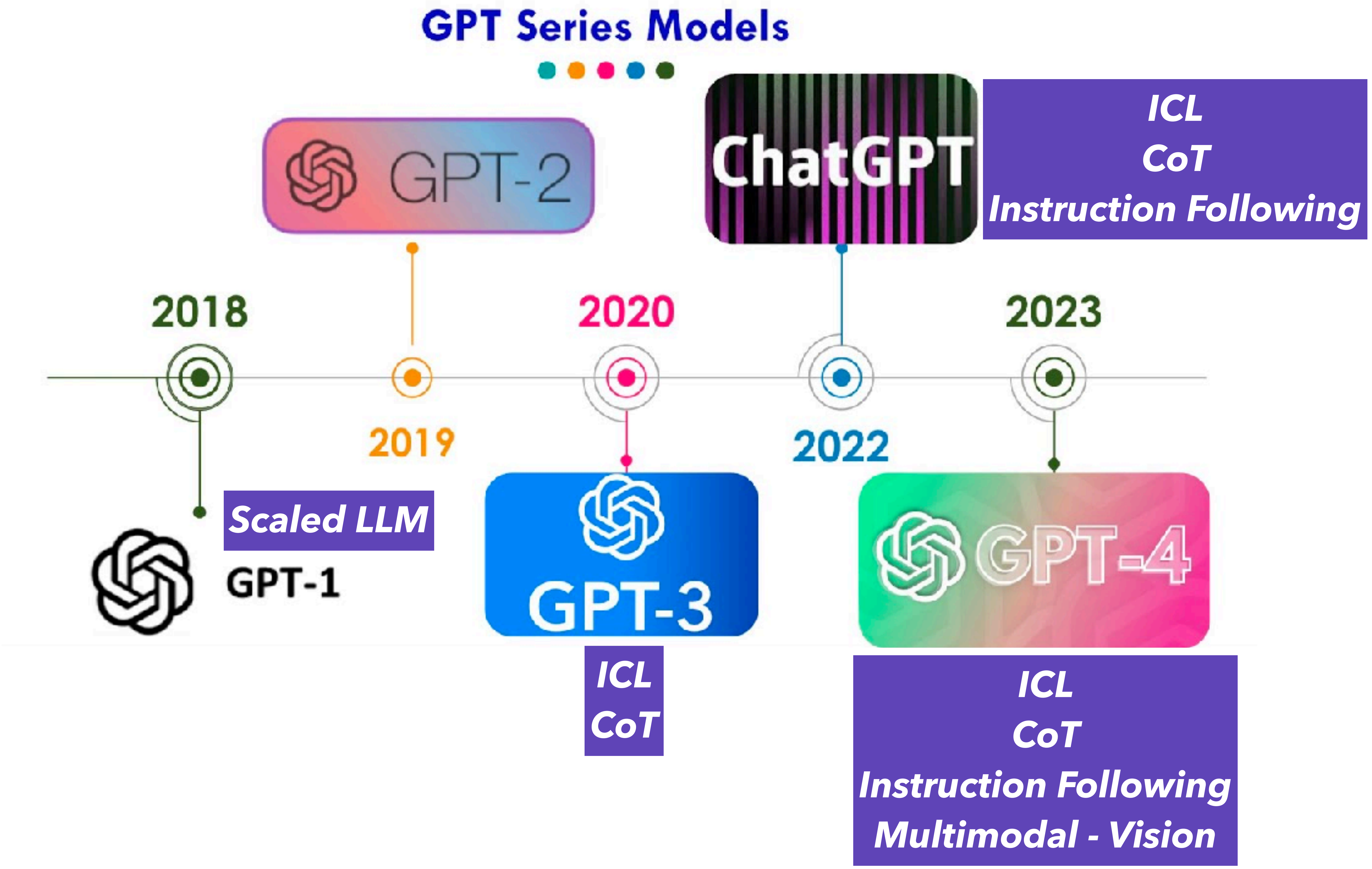
[An Empirical Study of GPT-3 for Few-Shot Knowledge-Based VQA. AAAI 2022]

[PromptCap: Prompt-Guided Image Captioning for VQA with GPT-3. ICCV 2023]

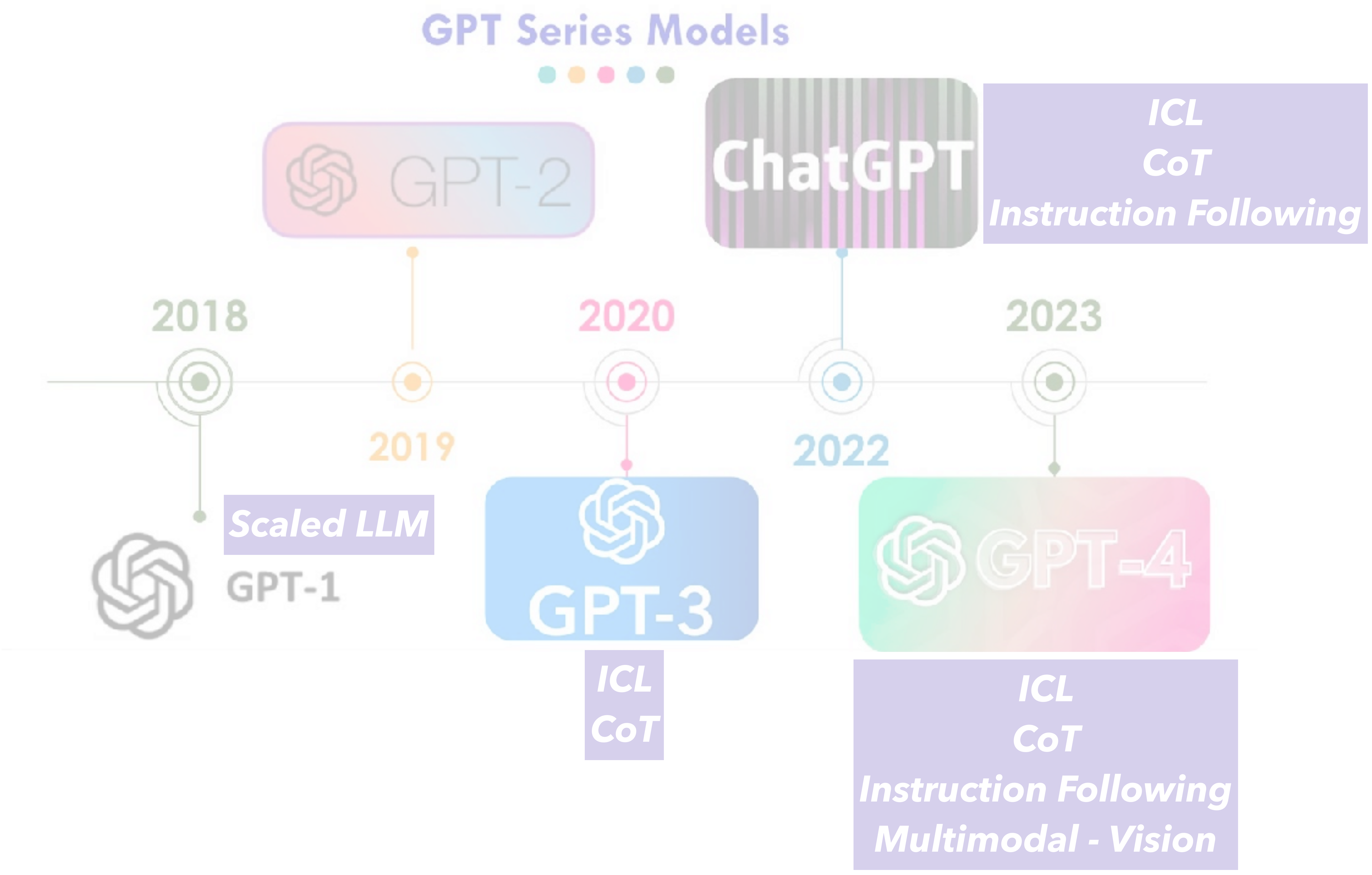
Multimodality

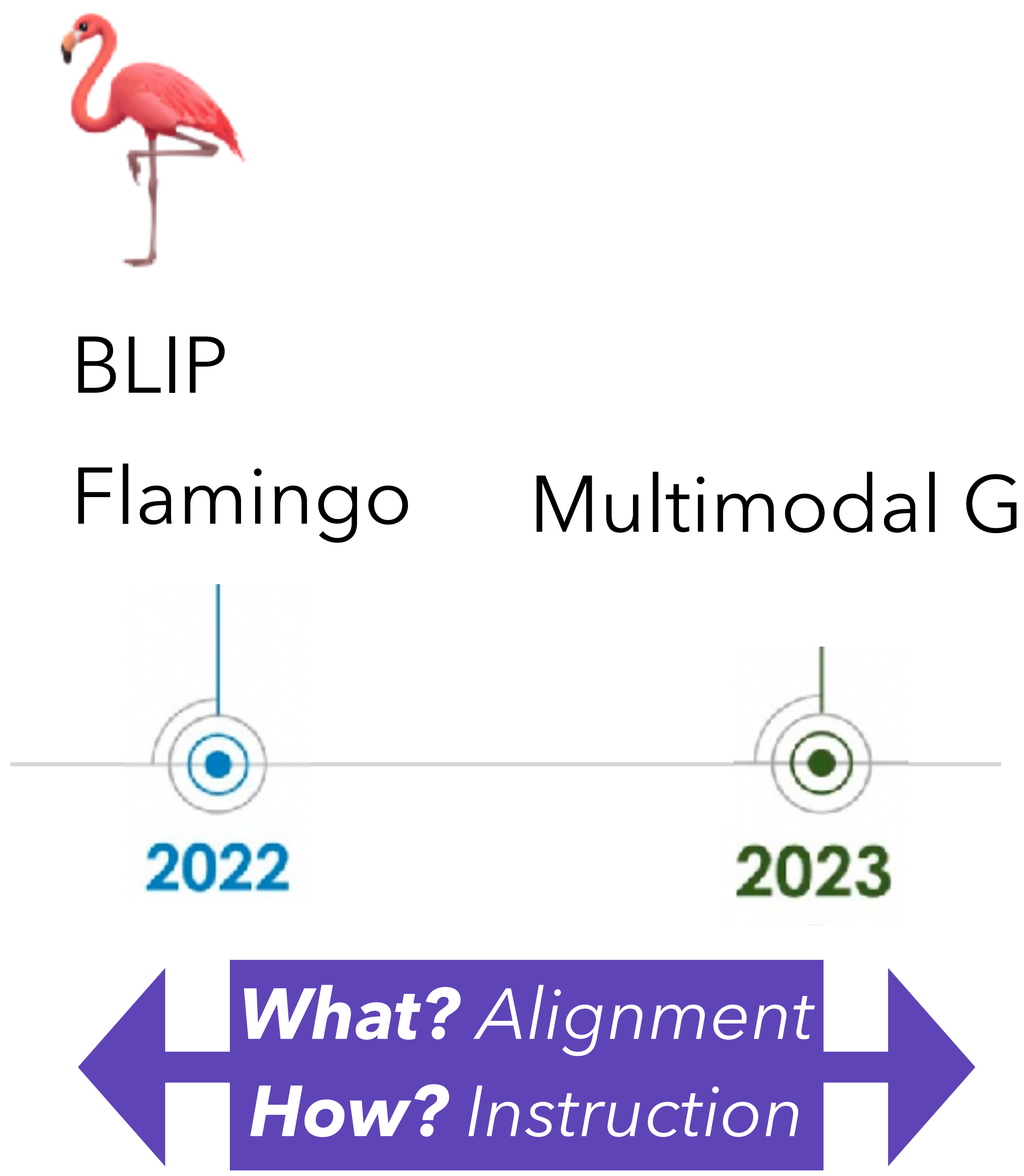
Instruction tuning,
In-context Learning,
T2I Generation

Gaps in Multimodal LMs



Gaps in Multimodal LMs





Instruction Tuning in LLMs

- Easily 0-shot **generalizable** (previously **hard** to generalize)
- Task instructions are given **explicitly** (previously **implicit**)
- Multi-tasking with specific **instructions** (previously **objectives**)
    - Easy to mix tasks with instructions

Instruction	Input	Output

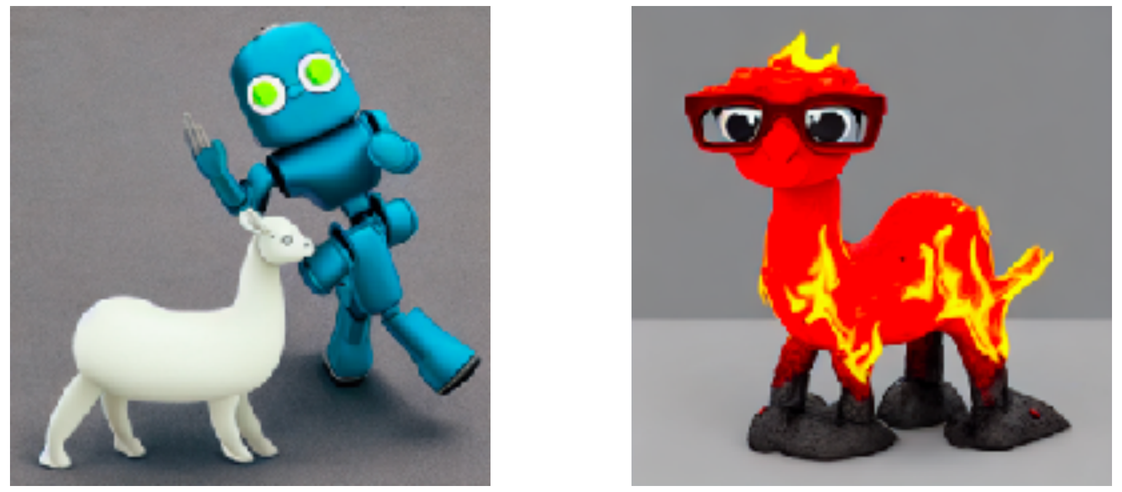
How about MM Instruction-tuning?

- Data?
- Model?
- Improvements?
- Open Questions?

Self-Instruct – strong (V)LM teachers

- LLM Teachers:
  - GPT 3.5
  - ShareGPT
- Resulting IF models

- Challenge for VLMs:
  - VLM Teachers?
  - Can we use LLM teachers?

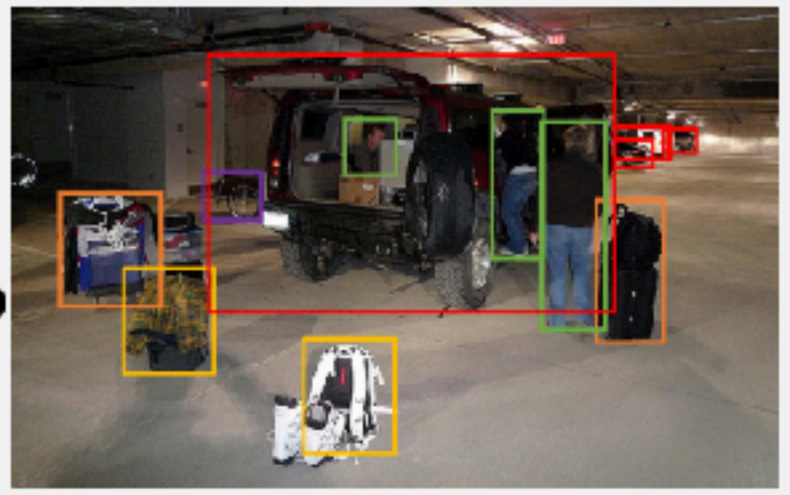


Use text-only teachers to bootstrap

Data for Visual Instruction-tuning

- Symbolic representations of images from GPT4
  - Captions
  - Bounding boxes
- GPT-assisted self-instruct tuning example generation

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>

Data for Visual Instruction-tuning

- 3 types of instruction-following questions and responses *(text-only with symbolic repr.)*
  - Conversation (sequential QA pairs)
  - Detailed Description
  - Complex Reasoning (very important)

```
messages = [ {"role":"system", "content": f"""You are an AI visual assistant, and you are
seeing a single image. What you see are provided with five sentences, describing the same image you
are looking at. Answer all questions as you are seeing the image.

Design a conversation between you and a person asking about this photo. The answers should be in a
tone that a visual AI assistant is seeing the image and answering the question. Ask diverse questions
and give corresponding answers.

Include questions asking about the visual content of the image, including the object types, counting
the objects, object actions, object locations, relative positions between objects, etc. Only include
questions that have definite answers:
(1) one can see the content in the image that the question asks about and can answer confidently;
(2) one can determine confidently from the image that it is not in the image. Do not ask any question
that cannot be answered confidently.

Also include complex questions that are relevant to the content in the image, for example, asking
about background knowledge of the objects in the image, asking to discuss about events happening in
the image, etc. Again, do not ask about uncertain details. Provide detailed answers when answering
complex questions. For example, give detailed examples or reasoning steps to make the content more
convincing and well-organized. You can include multiple paragraphs if necessary."""}
]
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']} )
messages.append({"role":"user", "content":'\n'.join(query)})
```

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.

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.

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.

Data for Visual Instruction-tuning

- 3 types of instruction-following questions and responses *(text-only with symbolic repr.)*
    - Conversation (sequential QA pairs)
    - Detailed Description
    - Complex Reasoning (very important)

- Use COCO images and captions
- GPT-4 Language only model to prompt
- Few-shot prompting with manual examples

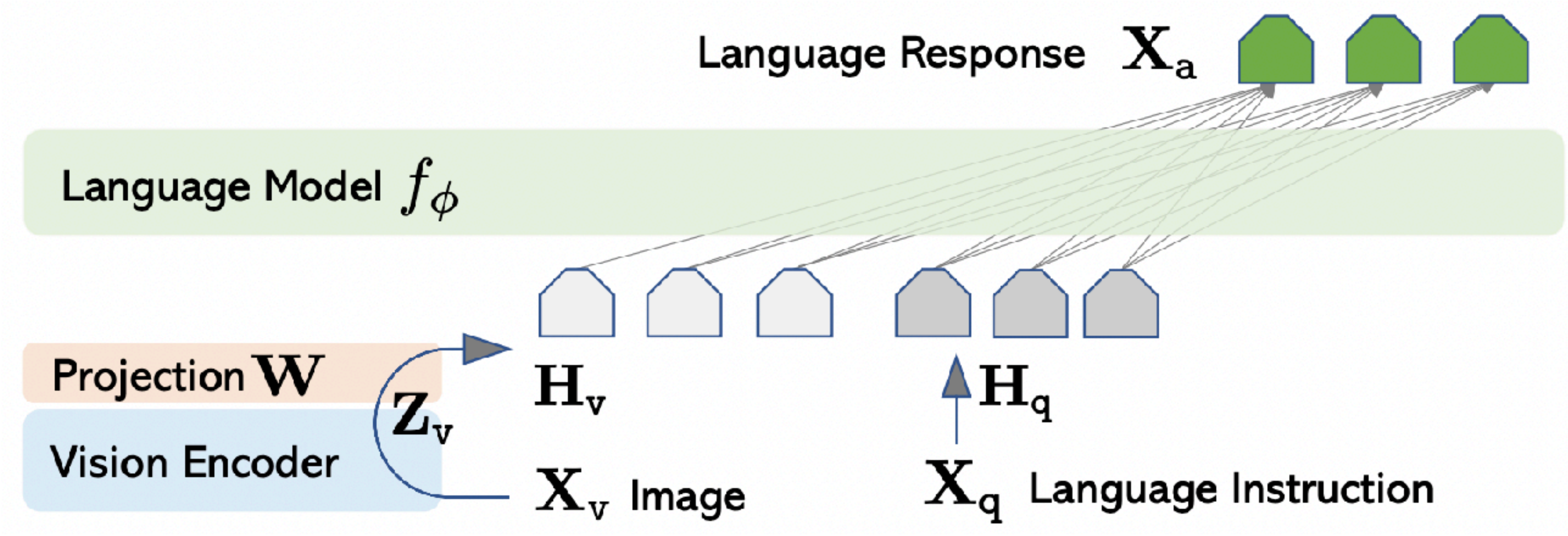
- 158k instruction following samples
    - 58k conversations
    - 23k detailed descriptions
    - 77k complex reasoning

LLaVA: Large Lang and Vis Assistant

$$p(\mathbf{X}_a|\mathbf{X}_v, \mathbf{X}_{\texttt{instruct}}) = \prod_{i=1}^{L} p_{\theta}(\boldsymbol{x}_i|\mathbf{X}_v, \mathbf{X}_{\texttt{instruct},<i}, \mathbf{X}_{a,<i}),$$

LLAMA

CLIP VE
VIT-L/14



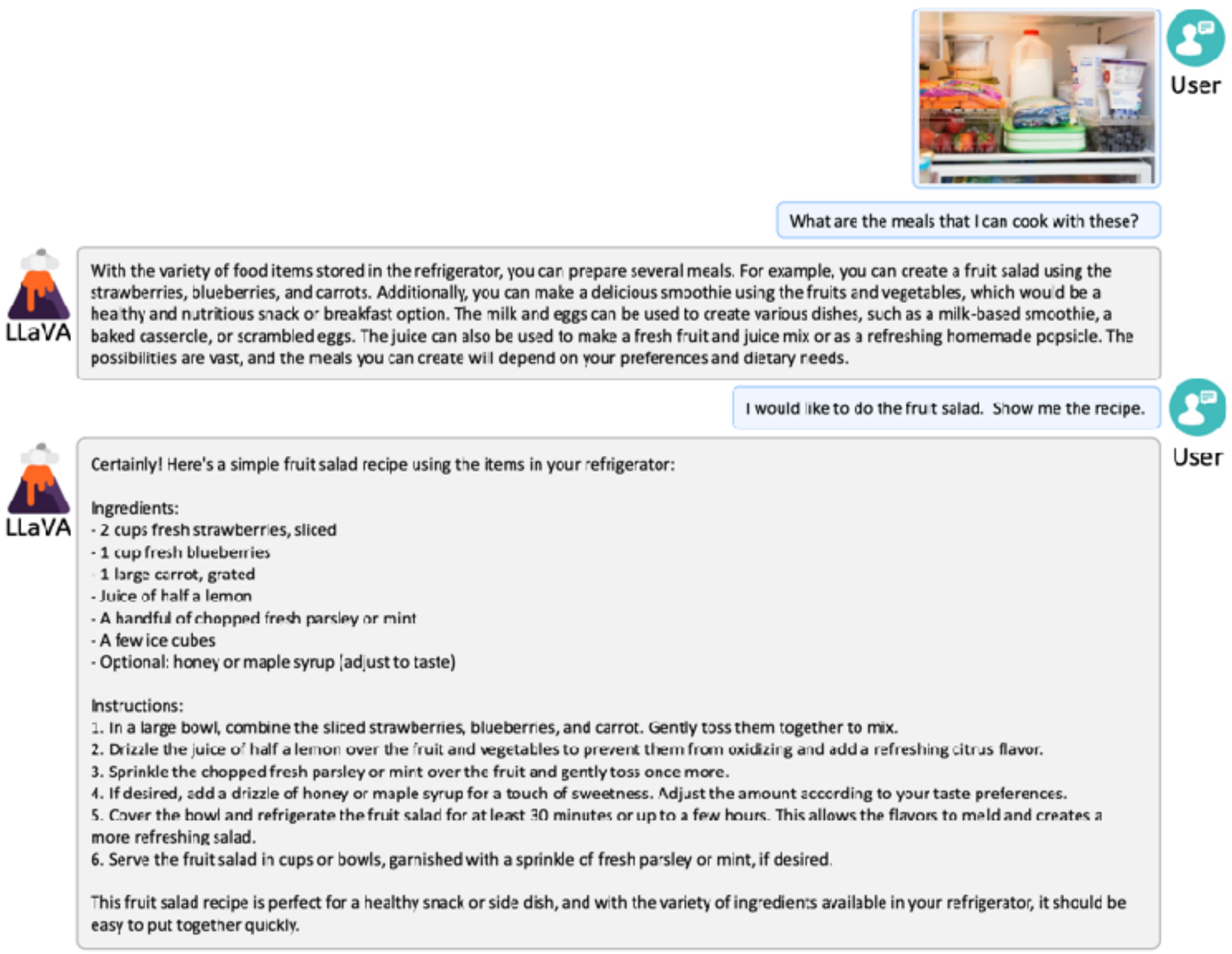
$\mathbf{X}_{\texttt{system-message}}$ <STOP>
Human : $\mathbf{X}^1_{\texttt{instruct}}$ <STOP> Assistant: $\mathbf{X}^1_a$ <STOP>
Human : $\mathbf{X}^2_{\texttt{instruct}}$ <STOP> Assistant: $\mathbf{X}^2_a$ <STOP> $\cdots$

$$\mathbf{X}^t_{\texttt{instruct}} = \begin{cases} \text{Randomly choose } [\mathbf{X}^1_q, \mathbf{X}_v] \text{ or } [\mathbf{X}_v, \mathbf{X}^1_q], & \text{the first turn } t = 1 \\ \mathbf{X}^t_q, & \text{the remaining turns } t > 1 \end{cases}$$

Dual stage training

- Stage 1: **Pre-training for feature alignment**
  - Only projection matrix is updated
  - Trained on a subset of CC3M (595k IT pairs)
- Stage 2: **Fine-tuning for user and task orientation**
  - Both projection matrix and LLM are updated
  - Tuned on Visual chat (user chat-like orientation 158k) & Science QA (complex science reasoning)

LLaVA demos



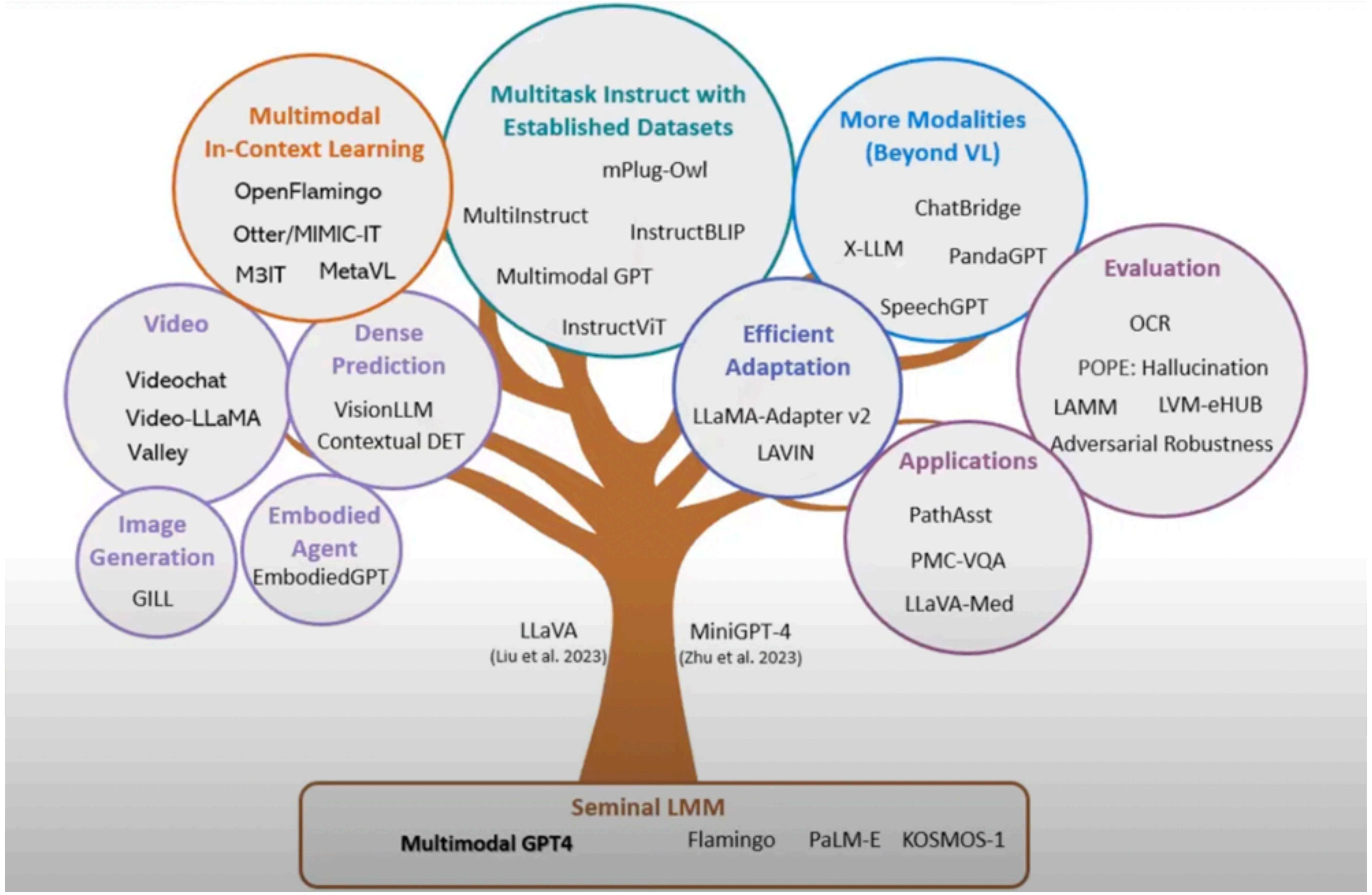
User
What are the meals that I can cook with these?
LLaVA
With the variety of food items stored in the refrigerator, you can prepare several meals. For example, you can create a fruit salad using the strawberries, blueberries, and carrots. Additionally, you can make a delicious smoothie using the fruits and vegetables, which would be a healthy and nutritious snack or breakfast option. The milk and eggs can be used to create various dishes, such as a milk-based smoothie, a baked casserole, or scrambled eggs. The juice can also be used to make a fresh fruit and juice mix or as a refreshing homemade popsicle. The possibilities are vast, and the meals you can create will depend on your preferences and dietary needs.
User
I would like to do the fruit salad. Show me the recipe.
LLaVA
Certainly! Here's a simple fruit salad recipe using the items in your refrigerator:
Ingredients:
- 2 cups fresh strawberries, sliced
- 1 cup fresh blueberries
- 1 large carrot, grated
- Juice of half a lemon
- A handful of chopped fresh parsley or mint
- A few ice cubes
- Optional: honey or maple syrup (adjust to taste)
Instructions:
1. In a large bowl, combine the sliced strawberries, blueberries, and carrot. Gently toss them together to mix.
2. Drizzle the juice of half a lemon over the fruit and vegetables to prevent them from oxidizing and add a refreshing citrus flavor.
3. Sprinkle the chopped fresh parsley or mint over the fruit and gently toss once more.
4. If desired, add a drizzle of honey or maple syrup for a touch of sweetness. Adjust the amount according to your taste preferences.
5. Cover the bowl and refrigerate the fruit salad for at least 30 minutes or up to a few hours. This allows the flavors to meld and creates a more refreshing salad.
6. Serve the fruit salad in cups or bowls, garnished with a sprinkle of fresh parsley or mint, if desired.
This fruit salad recipe is perfect for a healthy snack or side dish, and with the variety of ingredients available in your refrigerator, it should be easy to put together quickly.

My Joke Website
[Funny Joke]
[push to reveal punchline]
User
Turn this mock-up into an interactive website using html/js, replace the joke with a real joke.
LLaVA
<!DOCTYPE html>
<html>
<head>
<title>My Joke Website</title>
<style>
body { background-color: white; font-family: Arial, sans-serif; }
.container { display: flex; flex-direction: column; align-items: center; justify-content: center; height: 100vh; }
h1 { font-size: 36px; margin-bottom: 20px; }
p { font-size: 20px; }
button { background-color: blue; color: white; border: none; padding: 10px 20px; font-size: 18px; cursor: pointer; }
</style>
</head>
<body>
<div class="container">
<h1>My Joke Website</h1>
<p id="punchline">Welcome to my joke website. Push the button to reveal the punchline!</p>
<button onclick="revealPunchline()">Push me!</button>
</div>
<script>
function revealPunchline() {
document.getElementById("punchline").innerHTML = "The punchline is: 'Why was the math book sad? Because it had too many problems.'";
}
</script>
</body>
</html>
Place the generated HTML code into an HTML renderer to render the web page.
My Joke Website
Welcome to my joke website. Push the button to reveal the punchline!
Push me!
Rendered website
My Joke Website
The punchline is: 'Why was the math book sad? Because it had too many problems.'
Push me!
Rendered website (clicked the button)

Open Questions

- Human preference in Alignment
- Retrieval Augmentation for In-Context Learning

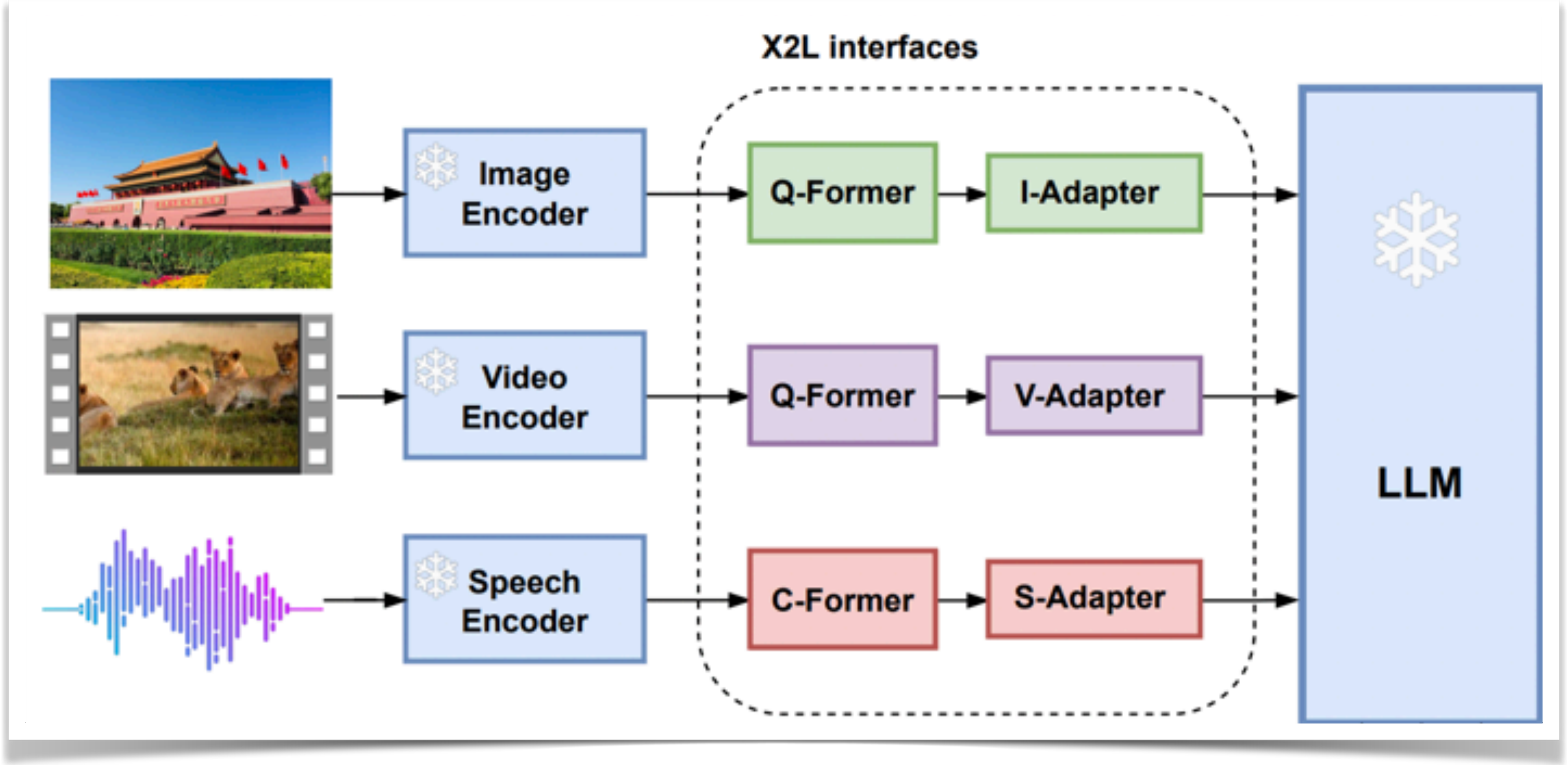
Map of other topics



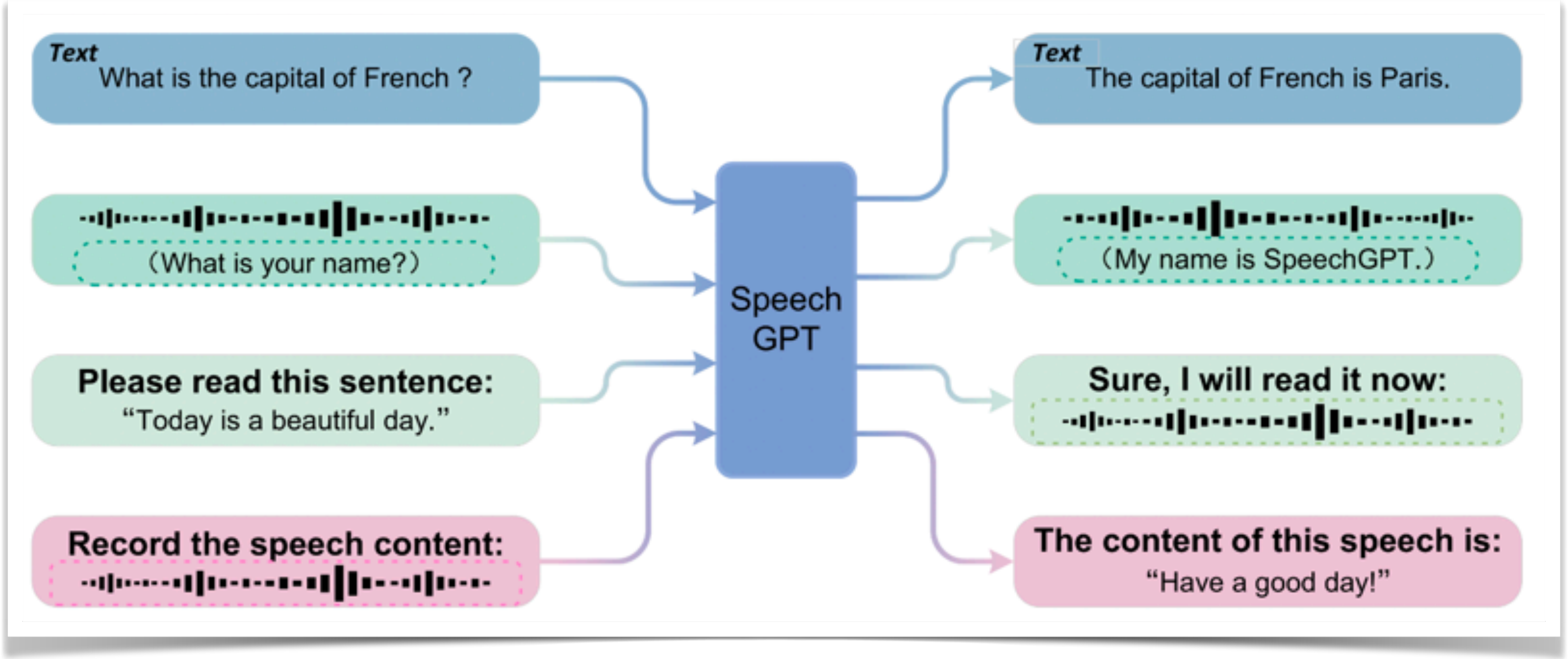
- Credit: VLP Tutorial @CVPR'23

More Modalities (Beyond VL)

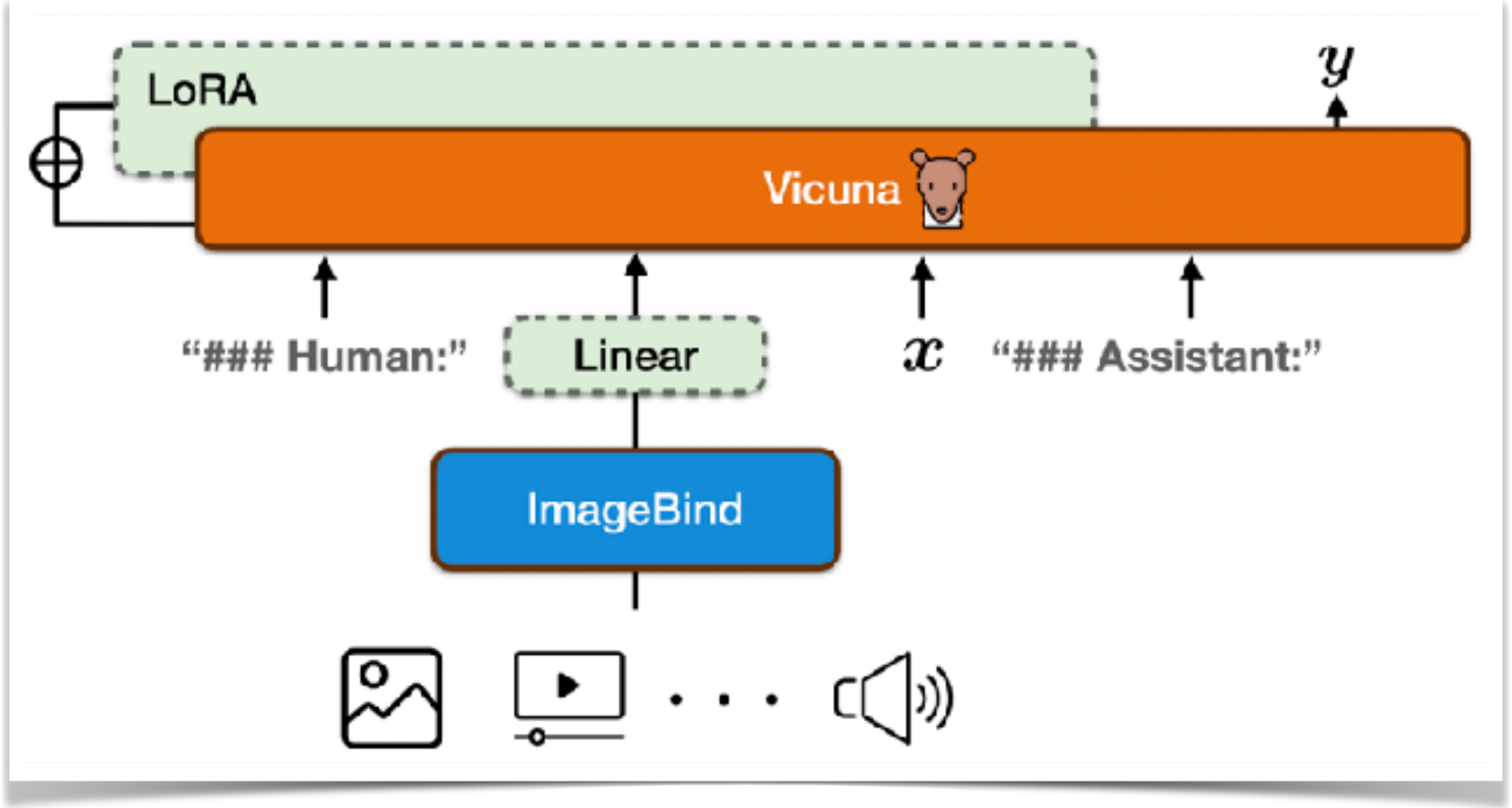
X-LLM



Speech-GPT



Panda-GPT



Evaluations

Evaluating Object Hallucination in Large Vision-Language Models

Yifan Li[1,3]*, Yifan Du[1,3]*, Kun Zhou[2]*, Jinpeng Wang [4],
Wayne Xin Zhao[2,3†] and Ji-Rong Wen[1,2,3]

LAMM: Language-Assisted Multi-Modal Instruction-Tuning Dataset, Framework, and Benchmark

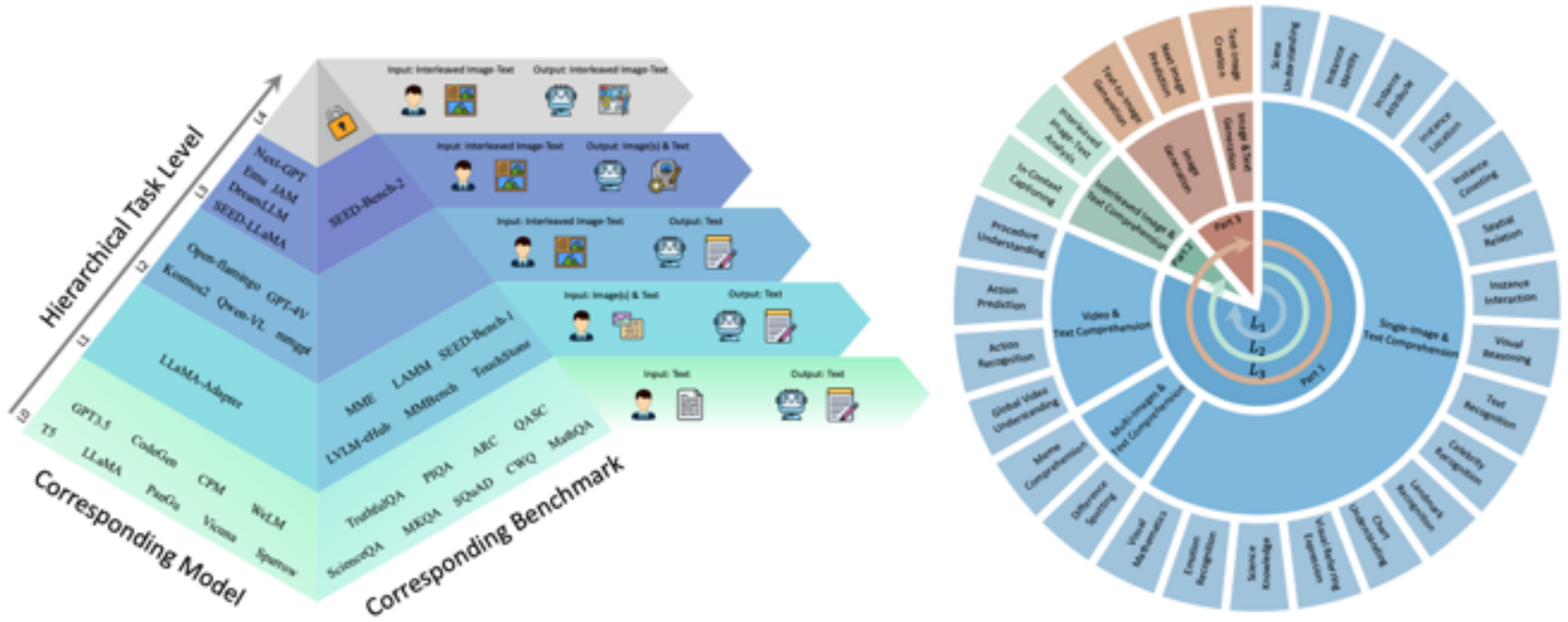
Zhenfei Yin[*1,3] Jiong Wang[*1,4] Jianjian Cao[*1,4] Zhelun Shi[*1,2] Dingning Liu[1,5] Mukai Li[1]
Xiaoshui Huang[1] Zhiyong Wang[3] Lu Sheng[2] Lei Bai[†,1] Jing Shao[†,1] Wanli Ouyang[1]
[1]Shanghai Artificial Intelligence Laboratory [2]Beihang University [3]The University of Sydney
[4]Fudan University [5]Dalian University of Technology
* Equal Contribution † Corresponding Authors

MMBench: Is Your Multi-modal Model an All-around Player?

Yuan Liu[1,*], Haodong Duan[1,*], Yuanhan Zhang[2,*], Bo Li[2,*], Songyang Zhang[1,*],
Wangbo Zhao[4], Yike Yuan[5], Jiaqi Wang[1], Conghui He[1], Ziwei Liu[2,†], Kai Chen[1,†]
Dahua Lin[1,3,†]
[1]Shanghai AI Laboratory [2]Nanyang Technological University
[3] The Chinese University of Hong Kong [4] National University of Singapore
[5] Zhejiang University
* Contribute equally in random order † Corresponding author

SEED-Bench-2: Benchmarking Multimodal Large Language Models

Bohao Li[3,1*] Yuying Ge[1*] Yixiao Ge[1,2†] Guangzhi Wang[2] Rui Wang[1]
Ruimao Zhang[3†] Ying Shan[1,2]
[1]Tencent AI Lab
[2]ARC Lab, Tencent PCG
[3]School of Data Science, The Chinese University of HongKong, Shenzhen



Applications to Domains/Tasks

Medical:

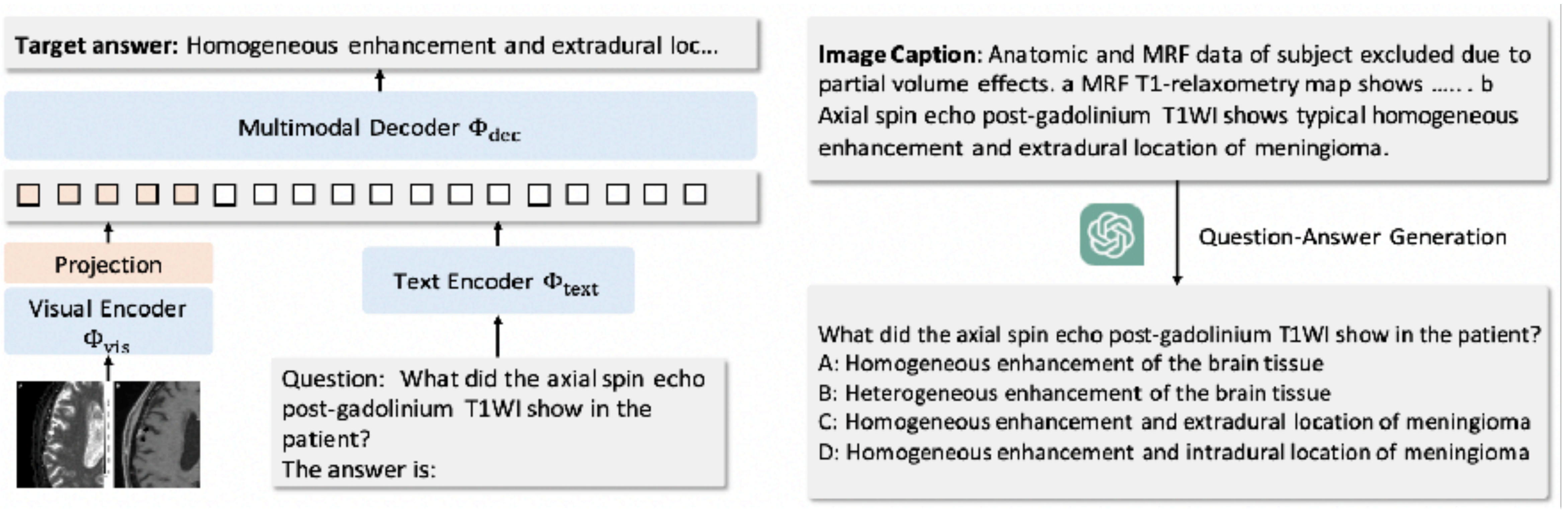
Med-LLaVA

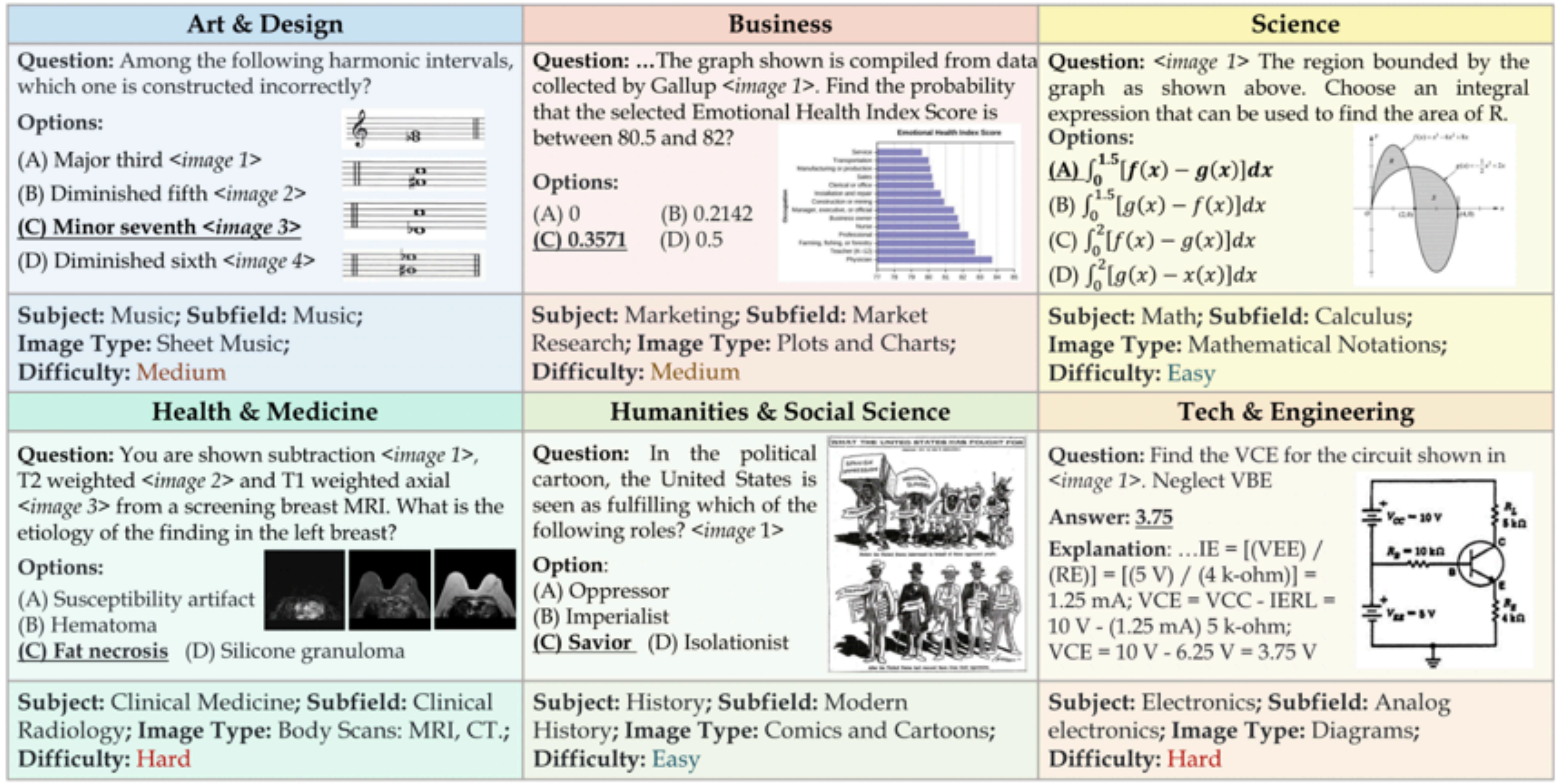
PMC-VQA

Domains - pathology, geometry, art and design

Image types - diagrams, tables, plots, chemical structures

Expert skill - Mathematical equations, science formula





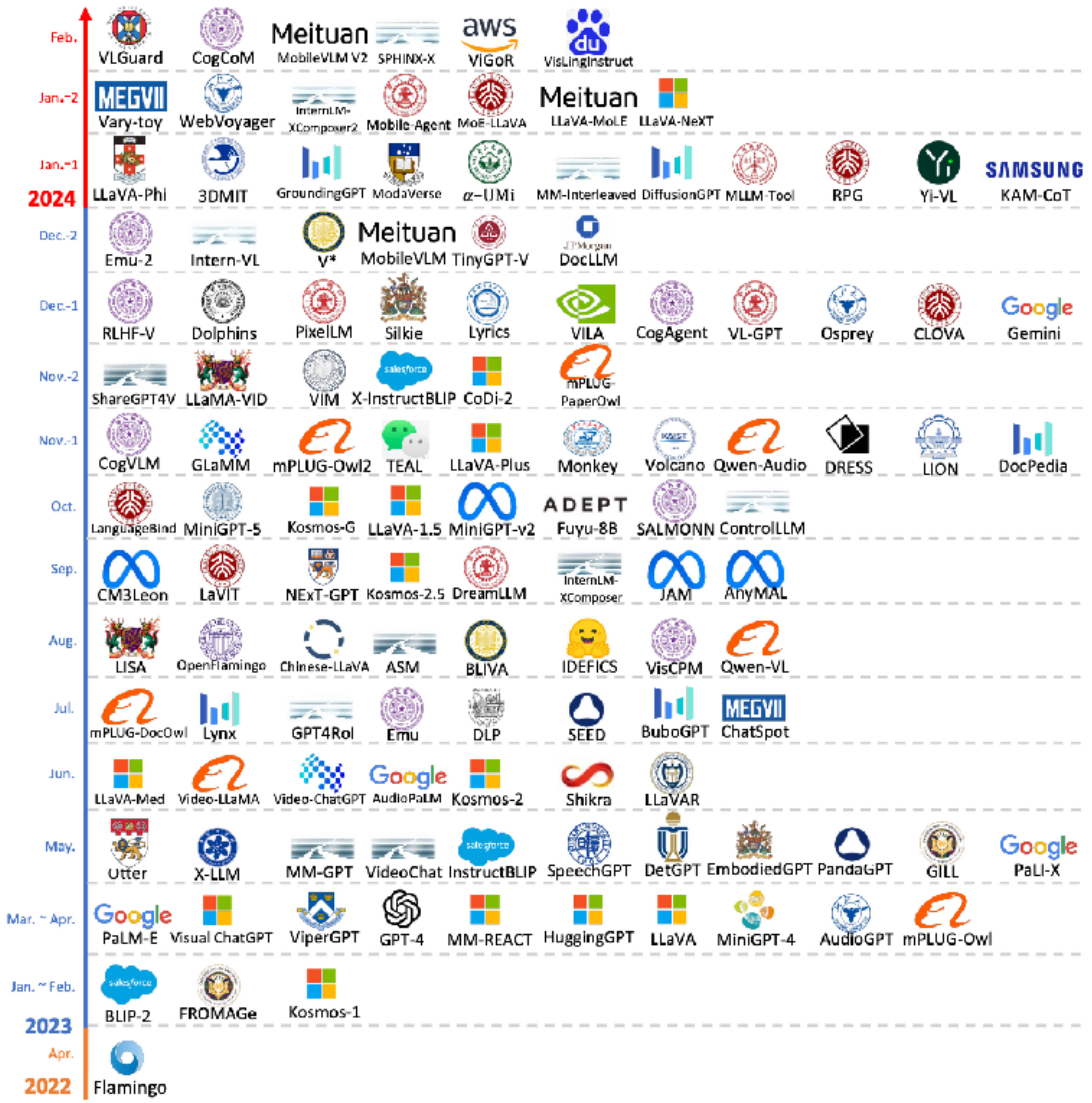
mmmu-benchmark.github.io/

BLINK: Multimodal LLMs Can See but Not Perceive

- 14 tasks that humans can solve "within a blink", but extremely challenging for multimodal LLMs
- Even the best multimodal LLMs lacks visual perception ability

🥇 Human: 96%
🥈 GPT4V: 51%
🥉 Gemini Pro: 45%
4️⃣ Claude OPUS: 43%
5️⃣ Random guess: 38%

A fast developing field



- **Big gaps remain**

1. Visual Perception Gaps
2. Multiple Images
3. 3D
4. Video reasoning
5. Encoding high resolution images
6. Encoding long sequences
7. Multimodal hallucinations

…

Q & A

Thank you for listening!

Slides adapted from Prof. Yejin Choi's CSE 517 winter 2024 slides