Multimodal Language Models

Zhexiao Xiong

02/29/2024

Multimodal Large Language Models(MLLMs)

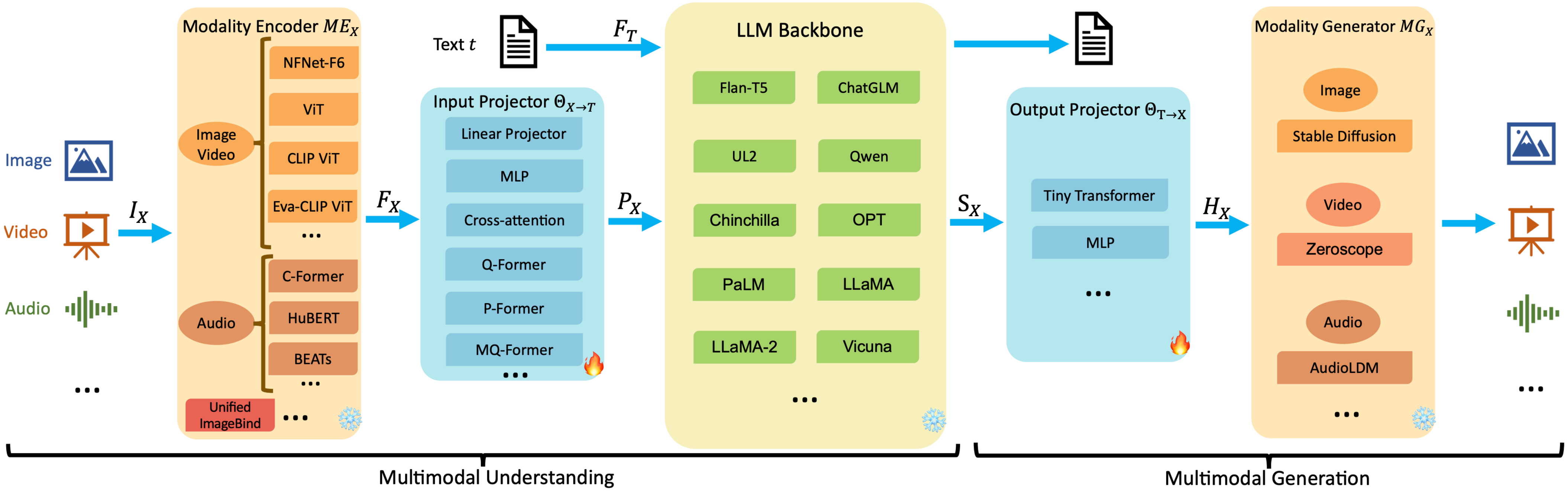
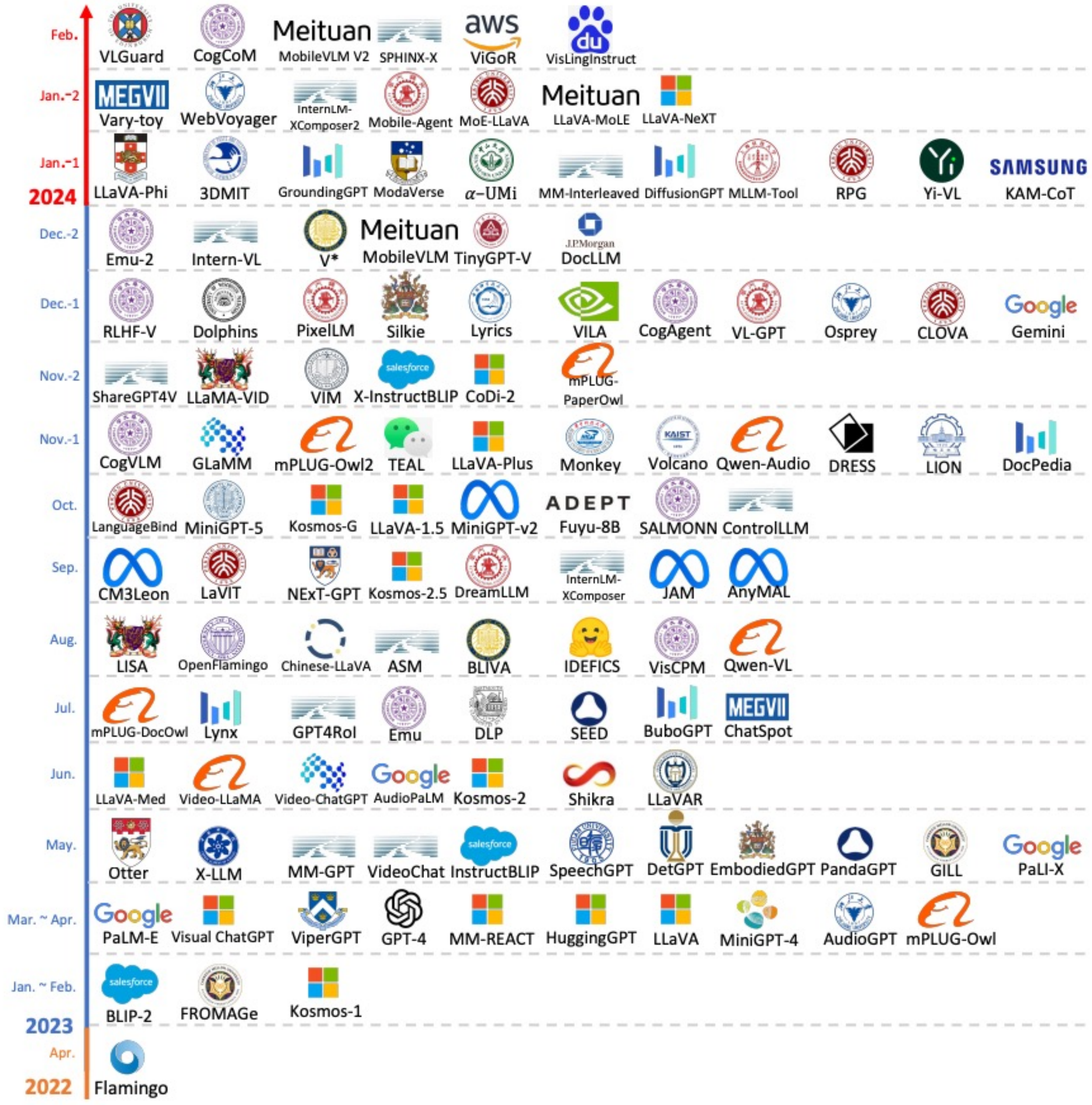
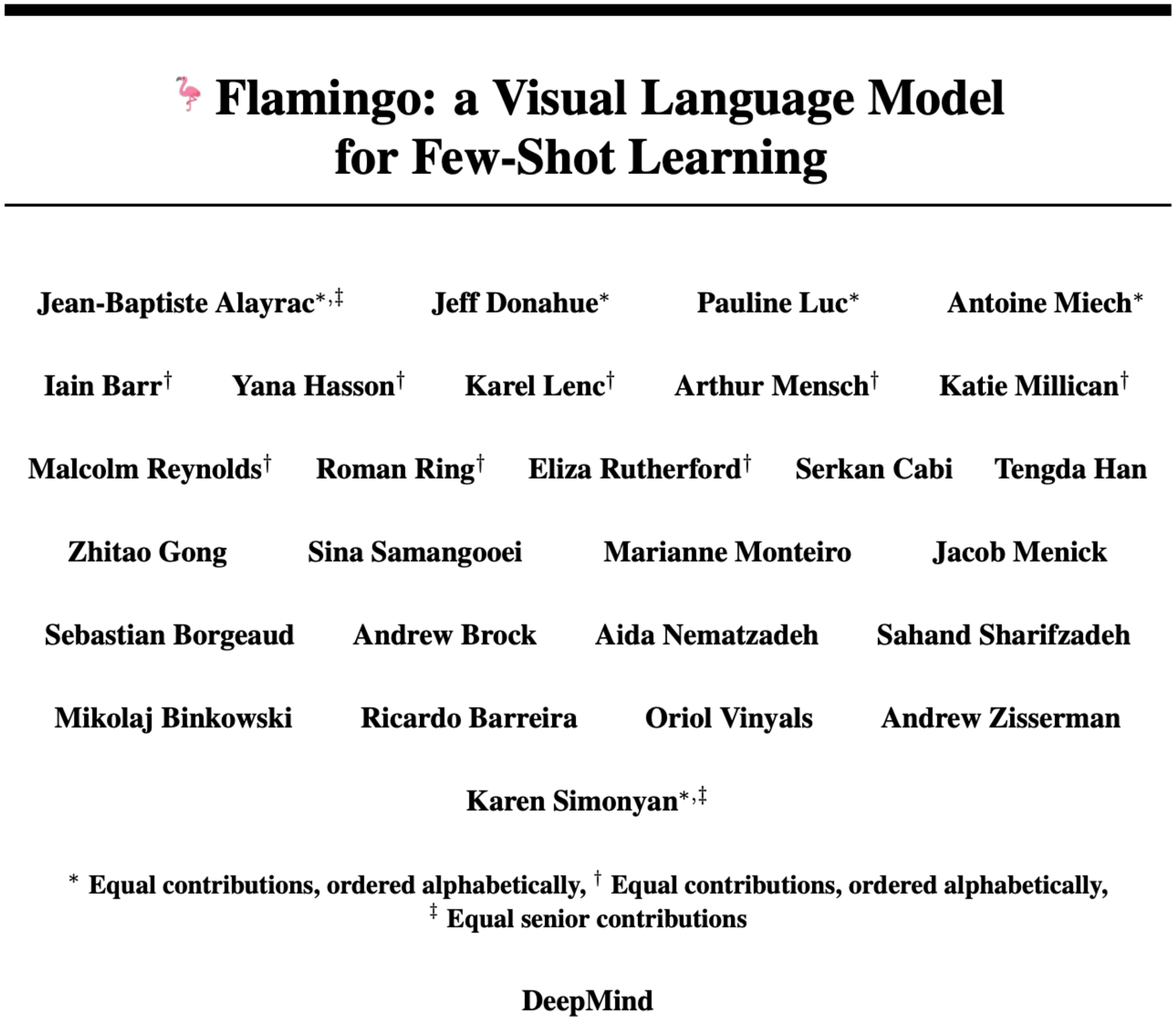


Figure 2: The general model architecture of MM-LLMs and the implementation choices for each component.

Timeline of Multimodal Large Language Models



Flamingo



🦩 Flamingo: a Visual Language Model for Few-Shot Learning

Jean-Baptiste Alayrac[*,‡] Jeff Donahue[*] Pauline Luc[*] Antoine Miech[*]

Iain Barr[†] Yana Hasson[†] Karel Lenc[†] Arthur Mensch[†] Katie Millican[†]

Malcolm Reynolds[†] Roman Ring[†] Eliza Rutherford[†] Serkan Cabi Tengda Han

Zhitao Gong Sina Samangooei Marianne Monteiro Jacob Menick

Sebastian Borgeaud Andrew Brock Aida Nematzadeh Sahand Sharifzadeh

Mikolaj Binkowski Ricardo Barreira Oriol Vinyals Andrew Zisserman

Karen Simonyan[*,‡]

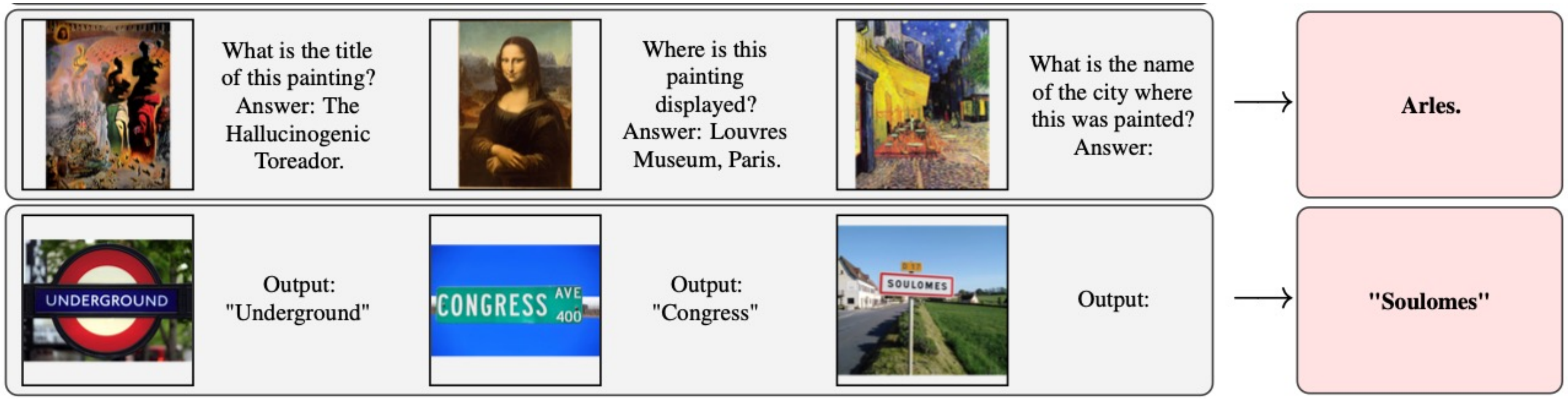
[*] Equal contributions, ordered alphabetically, [†] Equal contributions, ordered alphabetically,
[‡] Equal senior contributions

DeepMind

NeurIPS 2022

Task Description

- Input: text and visual data interleaved
- Output: Free-form text



What is the title of this painting? Answer: The Hallucinogenic Toreador.

Where is this painting displayed? Answer: Louvres Museum, Paris.

What is the name of the city where this was painted? Answer:

Arles.

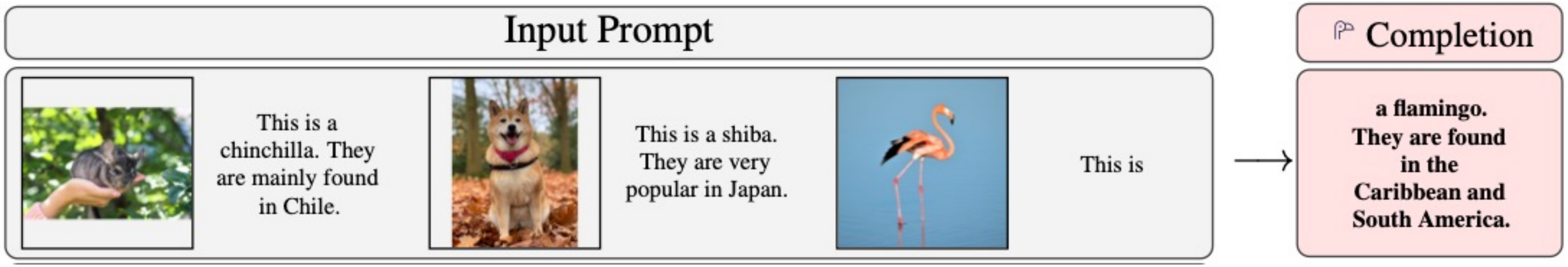
Output: "Underground"

Output: "Congress"

Output:

"Soulomes"

Task Description



Input Prompt

This is a chinchilla. They are mainly found in Chile.

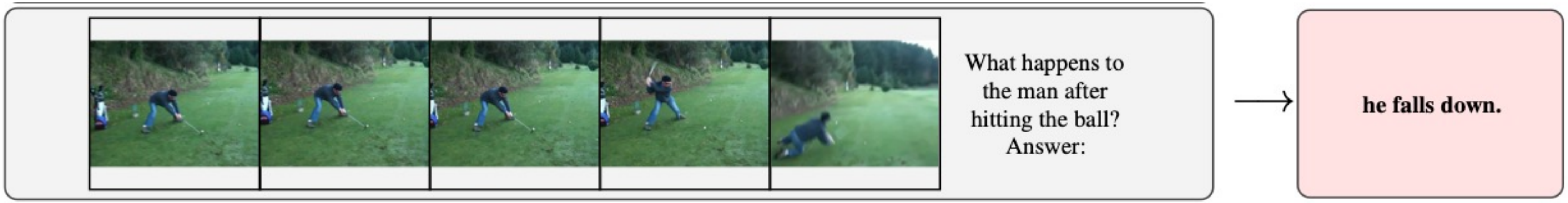
This is a shiba. They are very popular in Japan.

This is

Completion

a flamingo. They are found in the Caribbean and South America.

Image Understanding with few-shot prompting

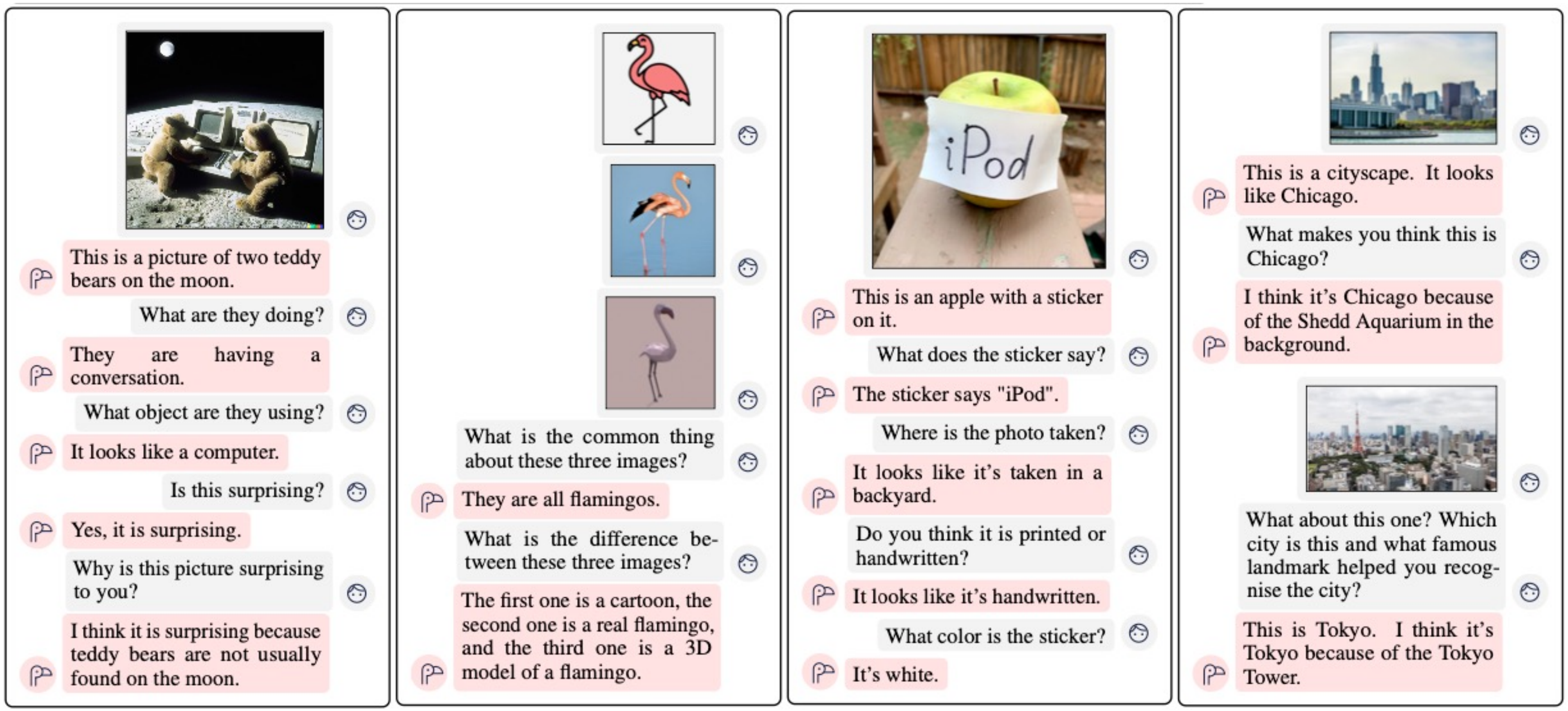


What happens to the man after hitting the ball? Answer:

he falls down.

Video Understanding/Multi-image Visual Dialogue

Task Description



This is a picture of two teddy bears on the moon.

What are they doing?

They are having a conversation.

What object are they using?

It looks like a computer.

Is this surprising?

Yes, it is surprising.

Why is this picture surprising to you?

I think it is surprising because teddy bears are not usually found on the moon.

What is the common thing about these three images?

They are all flamingos.

What is the difference between these three images?

The first one is a cartoon, the second one is a real flamingo, and the third one is a 3D model of a flamingo.

This is an apple with a sticker on it.

What does the sticker say?

The sticker says "iPod".

Where is the photo taken?

It looks like it's taken in a backyard.

Do you think it is printed or handwritten?

It looks like it's handwritten.

What color is the sticker?

It's white.

This is a cityscape. It looks like Chicago.

What makes you think this is Chicago?

I think it's Chicago because of the Shedd Aquarium in the background.

What about this one? Which city is this and what famous landmark helped you recognise the city?

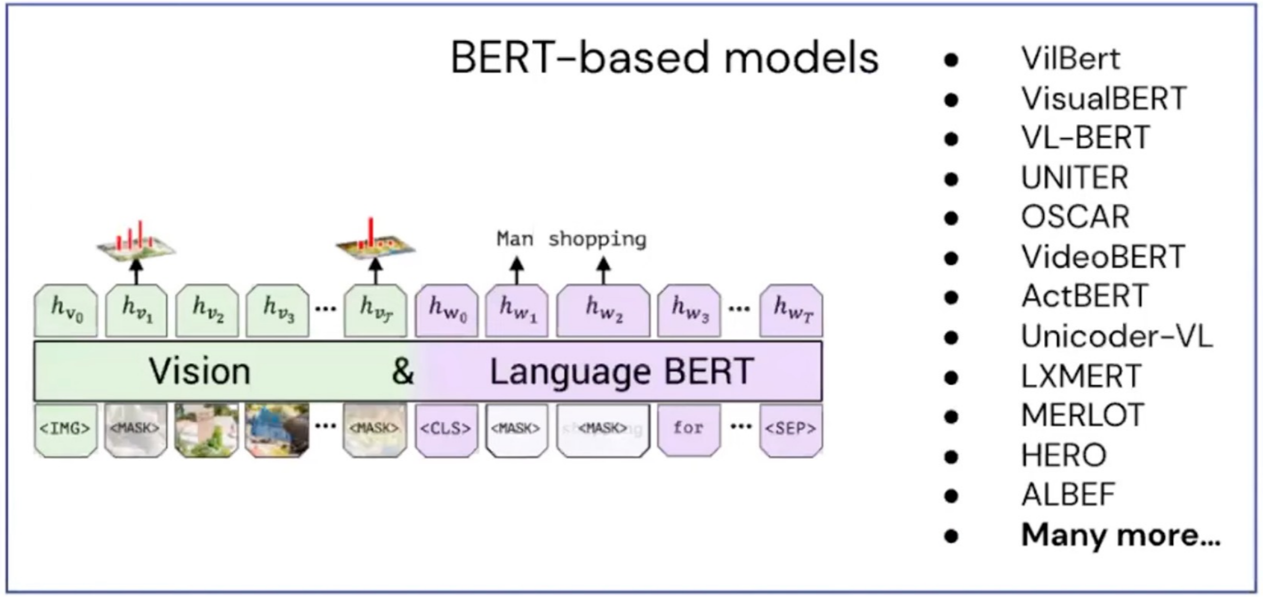
This is Tokyo. I think it's Tokyo because of the Tokyo Tower.

Visual Question Answering

Motivation

- An early attempt to build a state-of-the-art, generalist Visual Language Model that can be rapidly adapted to different multimodal tasks via few-shot learning.
  - Visual Language Model: ingest visual data(images/videos) along with a language input, produce language output.
  - Generalist: one model can address multiple task (captioning, visual dialogue, classification) with the same weights and without any post-hoc training.
  - Few-shot learning: condition the model to solve various tasks with only a few input-output examples.

Related work

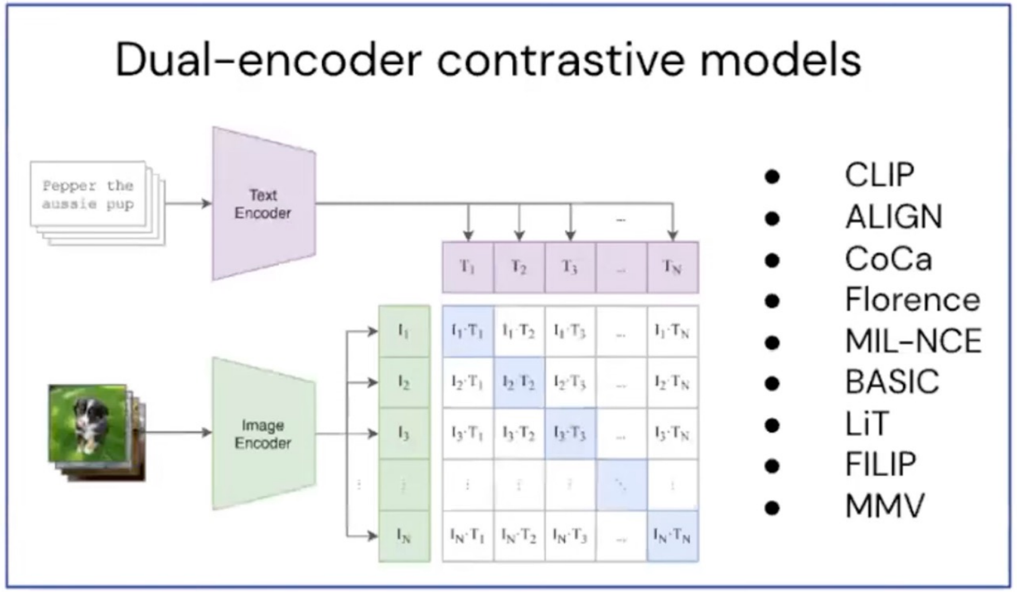


BERT-based models

- VilBert
- VisualBERT
- VL-BERT
- UNITER
- OSCAR
- VideoBERT
- ActBERT
- Unicoder-VL
- LXMERT
- MERLOT
- HERO
- ALBEF
- **Many more...**

Man shopping
h_{v_0} h_{v_1} h_{v_2} h_{v_3} ··· h_{v_T} h_{w_0} h_{w_1} h_{w_2} h_{w_3} ··· h_{w_T}
Vision & Language BERT
<IMG> <MASK> ··· <MASK> <CLS> <MASK> <MASK> for ··· <SEP>

BERT-Based Models



Dual-encoder contrastive models

- CLIP
- ALIGN
- CoCa
- Florence
- MIL-NCE
- BASIC
- LiT
- FILIP
- MMV

Pepper the aussie pup
Text Encoder
Image Encoder
T₁ T₂ T₃ ... T_N
I₁ I₂ I₃ ⋮ I_N
I₁·T₁ I₁·T₂ I₁·T₃ ... I₁·T_N
I₂·T₁ I₂·T₂ I₂·T₃ ... I₂·T_N
I₃·T₁ I₃·T₂ I₃·T₃ ... I₃·T_N
I_N·T₁ I_N·T₂ I_N·T₃ ... I_N·T_N

CLIP series

Architecture

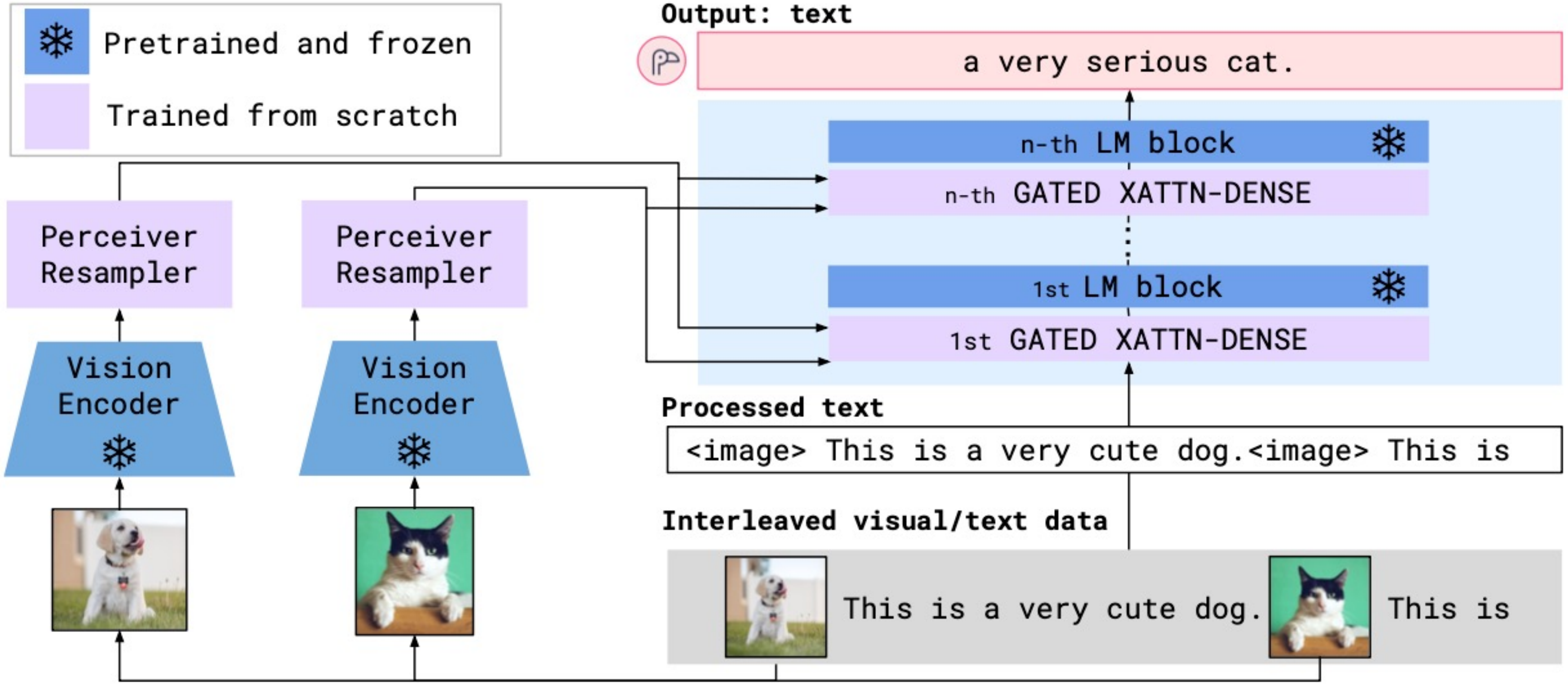
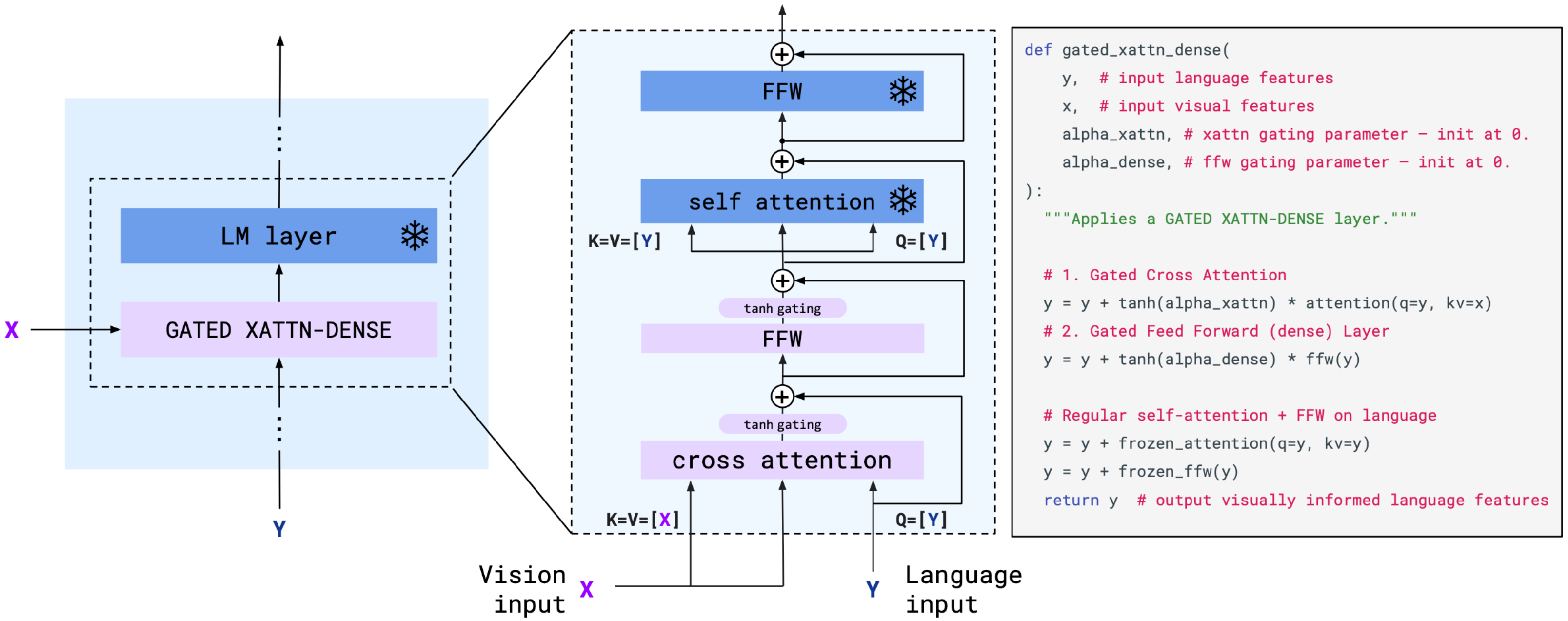


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Pretrained parts of the model are frozen: the Vision Encoder and the LLM

GATED XATTN-DENSE block



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

  # 1. Gated Cross Attention
  y = y + tanh(alpha_xattn) * attention(q=y, kv=x)
  # 2. Gated Feed Forward (dense) Layer
  y = y + tanh(alpha_dense) * ffw(y)

  # Regular self-attention + FFW on language
  y = y + frozen_attention(q=y, kv=y)
  y = y + frozen_ffw(y)
  return y  # output visually informed language features
```

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

Training datasets

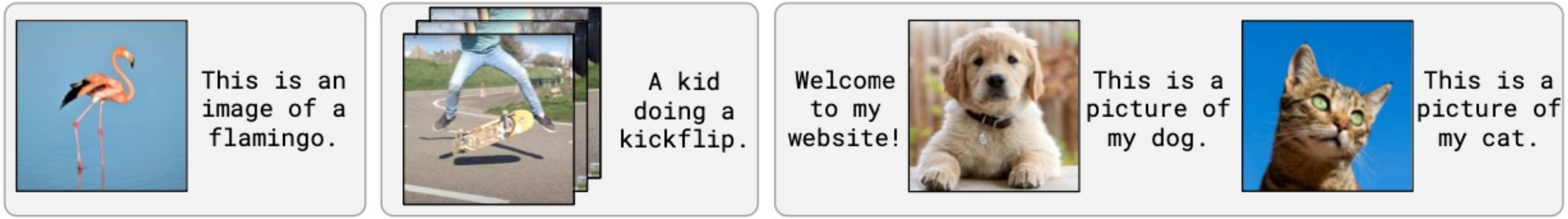

This is an image of a flamingo.
A kid doing a kickflip.
Welcome to my website!
This is a picture of my dog.
This is a picture of my cat.

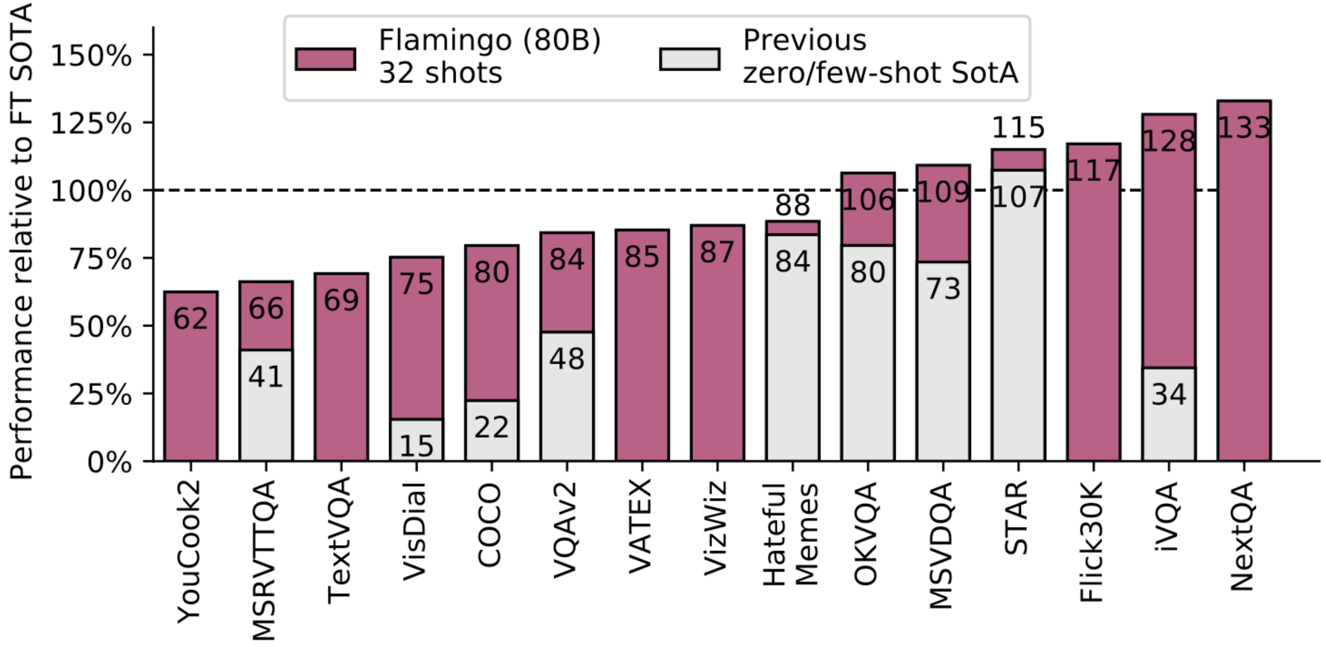
Image-Text Pairs dataset `[N=1, T=1, H, W, C]`

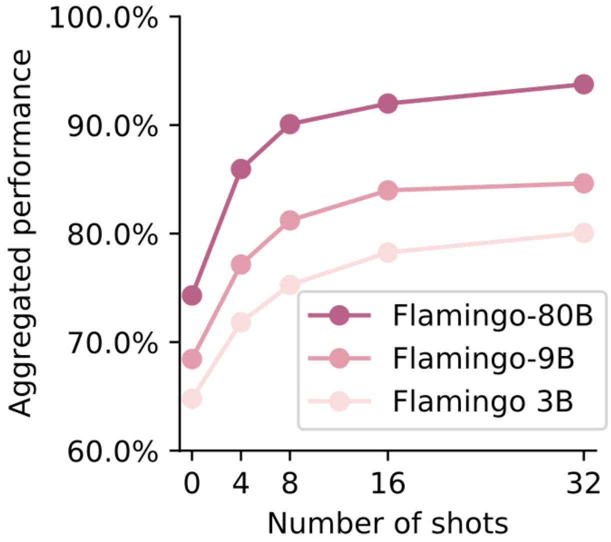
Video-Text Pairs dataset `[N=1, T>1, H, W, C]`

Multi-Modal Massive Web (M3W) dataset `[N>1, T=1, H, W, C]`

Figure 9: **Training datasets.** Mixture of training datasets of different formats. N corresponds to the number of visual inputs for a single example. For paired image (or video) and text datasets, $N = 1$. T is the number of video frames ($T = 1$ for images). H, W, and C are height, width and color channels.

Results


Performance relative to FT SOTA
150%
125%
100%
75%
50%
25%
0%
Flamingo (80B) 32 shots
Previous zero/few-shot SotA
62
66
41
69
75
15
80
22
84
48
85
87
88
84
106
80
109
73
115
107
117
128
34
133
YouCook2
MSRVTTQA
TextVQA
VisDial
COCO
VQAv2
VATEX
VizWiz
Hateful Memes
OKVQA
MSVDQA
STAR
Flick30K
iVQA
NextQA


Aggregated performance
100.0%
90.0%
80.0%
70.0%
60.0%
0
4
8
16
32
Number of shots
Flamingo-80B
Flamingo-9B
Flamingo 3B

Limitations

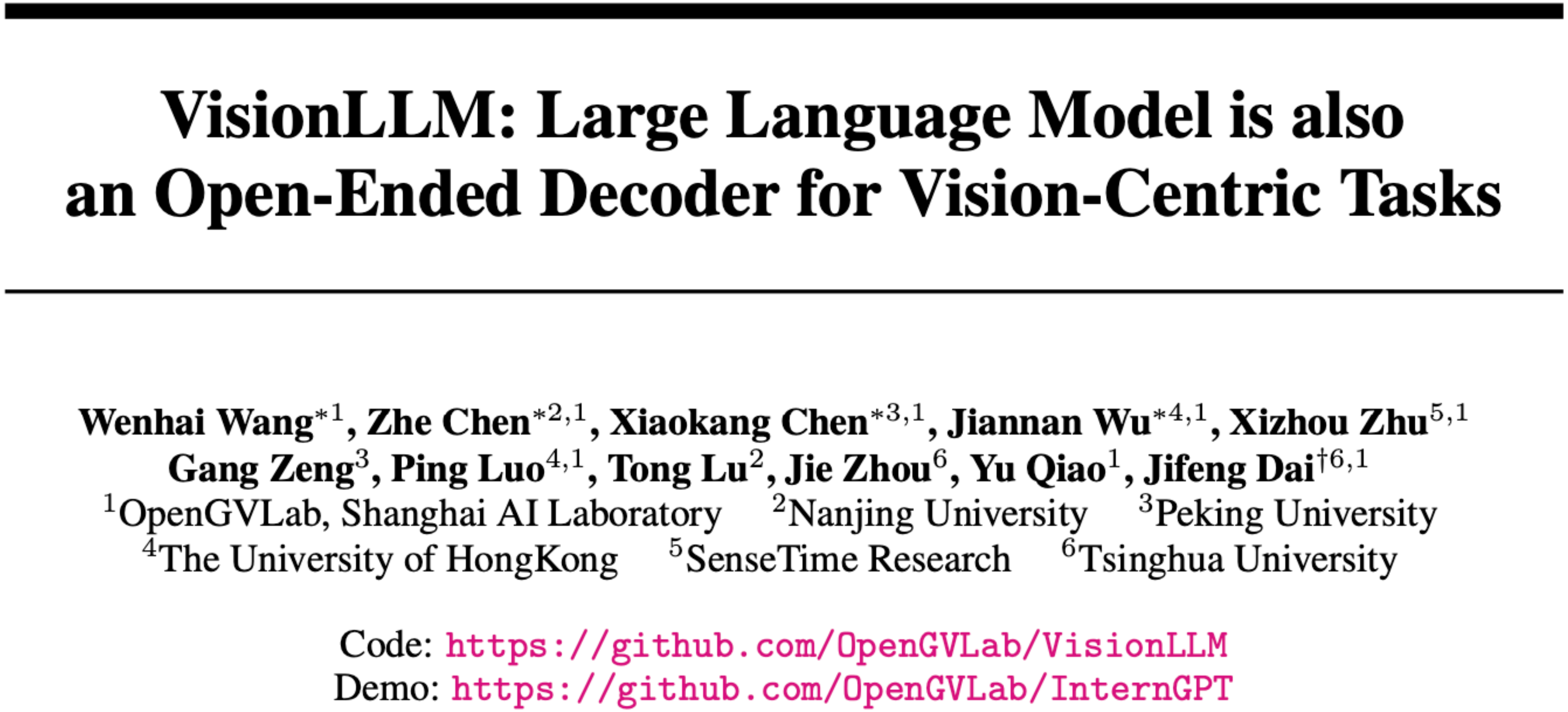
- Generalize poorly to sequences longer than the training ones.

- The classification performance of Flamingo lags behind that of state-of-the-art contrastive models.

- In-context learning has significant advantages over gradient-based few-shot learning methods, but also suffers from drawbacks depending on the characteristics of the application.

Conclusion

- Introduce the Flamingo family of VLMs which can perform various multimodal tasks (such as captioning, visual dialogue, or visual question-answering) from only a few input/output examples.
- Evaluate how Flamingo models can be adapted to various tasks via few-shot learning.
- *Flamingo* sets a new state of the art (at that time) in few-shot learning on a wide array of 16 multimodal language and image/video understanding tasks.



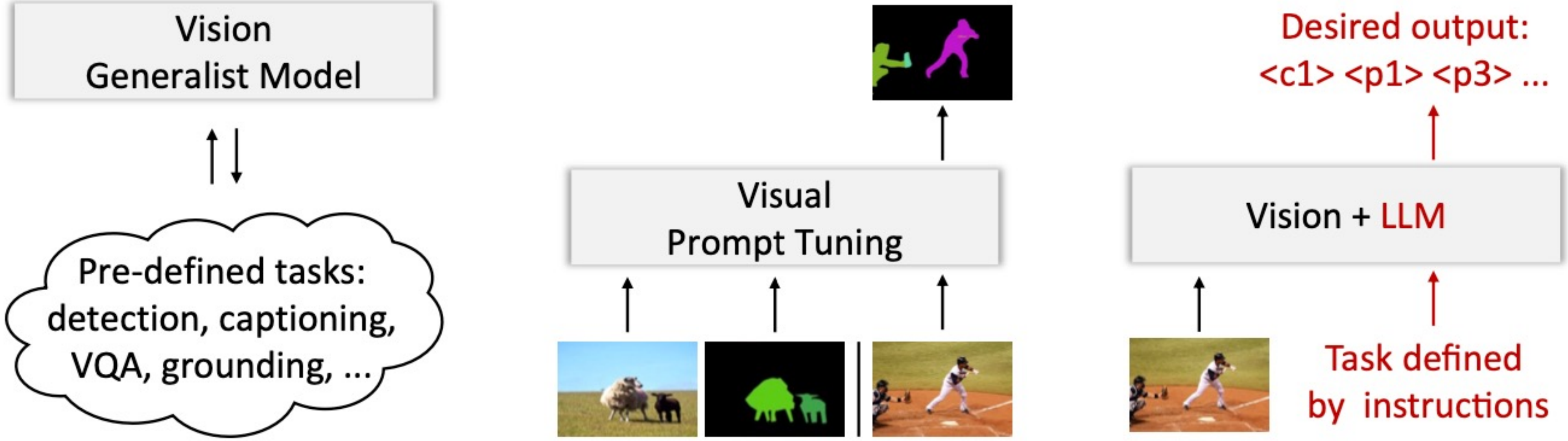
VisionLLM: Large Language Model is also an Open-Ended Decoder for Vision-Centric Tasks

Wenhai Wang*[1], Zhe Chen*[2,1], Xiaokang Chen*[3,1], Jiannan Wu*[4,1], Xizhou Zhu[5,1]
Gang Zeng[3], Ping Luo[4,1], Tong Lu[2], Jie Zhou[6], Yu Qiao[1], Jifeng Dai†[6,1]
[1]OpenGVLab, Shanghai AI Laboratory [2]Nanjing University [3]Peking University
[4]The University of HongKong [5]SenseTime Research [6]Tsinghua University

Code: https://github.com/OpenGVLab/VisionLLM
Demo: https://github.com/OpenGVLab/InternGPT

NeurIPS 2023

Background



(a) Vision generalist models [59, 61, 83] are constrained by the format of pre-defined tasks.

(b) Visual prompt tuning [26, 64, 62] are inconsistent with the format of LLMs.

(c) VisionLLM (ours) can *flexibly manage vision-centric tasks using language instructions like LLMs*.

Figure 1: **Comparison of our VisionLLM with popular paradigms.** Unlike current vision generalist models that depend on pre-defined task formats and visual prompt tuning models that are inconsistent with large language models (LLMs), VisionLLM leverages the power of LLMs for open-ended vision tasks by using language instructions.

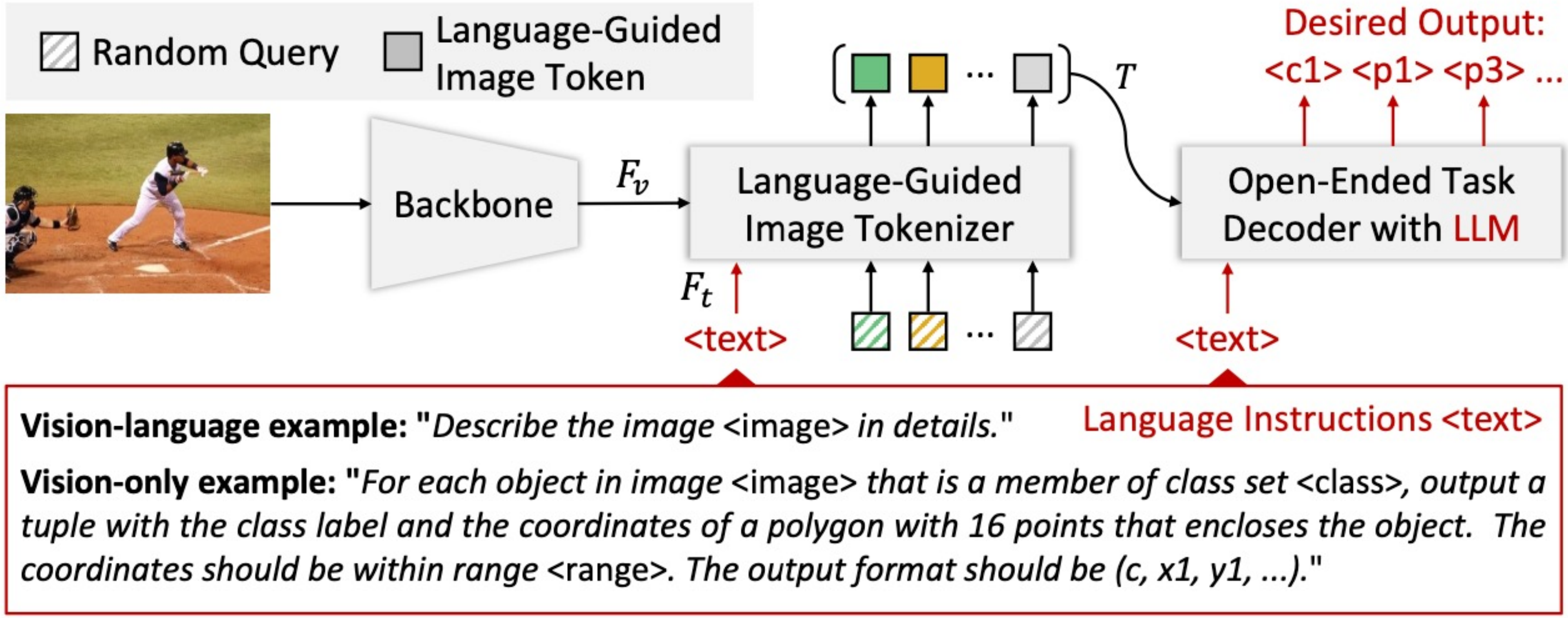
Contributions

- Propose VisionLLM, the first framework that leverages the power of LLMs to address vision- centric tasks in an open-ended and customizable manner.
- Design unified language instruction that matches the format of language models and covers various vision-centric tasks including visual perception.
    - language-guided image tokenizer
    - an LLM-based task decoder
- Showcase their ability to handle diverse scenarios, including random object categories, random output formats, and random task descriptions
- Construct a series of tasks to verify the effectiveness of the model, showcasing its ability to handle diverse scenarios, including random object categories, random output formats, and random task descriptions

VisionLLM

- Framework
- Unified Language Instruction
- Language-Guided Image Tokenizer
- LLM-based Open-Ended Task Decoder

Overall Architecture


▨ Random Query
■ Language-Guided Image Token
Backbone
F_v
Language-Guided Image Tokenizer
F_t
<text>
T
Desired Output:
<c1> <p1> <p3> ...
Open-Ended Task Decoder with LLM
<text>
Language Instructions <text>

Vision-language example: "*Describe the image* <image> *in details.*"

Vision-only example: "*For each object in image* <image> *that is a member of class set* <class>, *output a tuple with the class label and the coordinates of a polygon with 16 points that encloses the object. The coordinates should be within range* <range>. *The output format should be (c, x1, y1, ...).*"

Unified Language Instruction

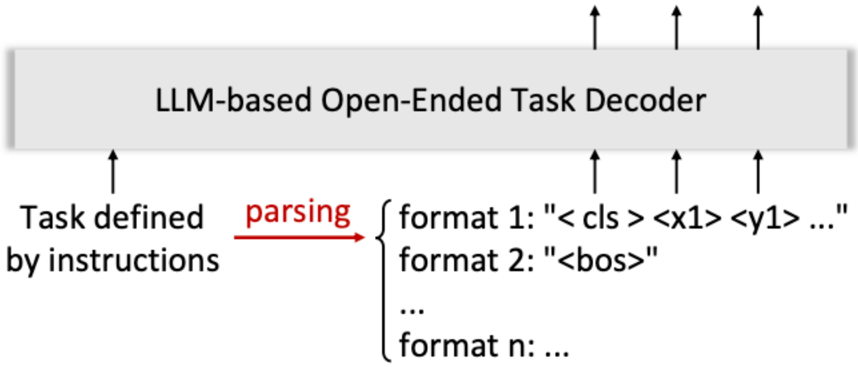
- Vision-Language Tasks: Straightforward, similar to NLP tasks.
  - Image caption task: *The image is* <image>. *Please generate a caption for the image:*
  - VQA task: *The image is* <image>. *Please generate an answer for the image according to the question:* <question>
- Vision-Only Tasks: describe vision tasks by providing a task description and specifying the desired output format via language instructions.
  - Take instance segmentation task as example: *Segment all the objects of category set* <class> *within the* <range> *of the image and generate a list of the format (c, x1, y1, x2, y2, ..., x8, y8). Here, c represents the index of the class label starting from 0, and (x1, y1, x2, y2, ..., x8, y8) correspond to the offsets of boundary points of the object relative to the center point. The image is:* <image>

Language Guided Image Tokenizer

- Images are treated as a kind of foreign language and are converted into token representations.
- Steps:
  - Image Backbone -> Visual features at 4 different scales.
  - Text Encoder -> Language Features
  - Language features injected into each scale of visual features using cross attention -> multi-scale language-aware visual features
  - Transformer-based network with M random-initialized queires Q to capture the high-level information of images on top of multi-scale language-aware visual features to extract M image tokens.

LLM-based Open-Ended Task Decoder

- Decoder based on Alpaca(adapted from LLaMA)
- Cons with Alpaca:
  - Few digit tokens (0-9), unable to locate objects by numbers
  - Multiple tokens for category name -> inefficient classification
  - Causal Model -> inefficient for visual perception tasks.
- Solutions:
  - Discretized Location tokens
  - Category name tokens
  - Output format is query decoding: Parsing of the structural output format by LLM.


LLM-based Open-Ended Task Decoder
Task defined by instructions
parsing
format 1: "< cls > <x1> <y1> ..."
format 2: "<bos>"
...
format n: ...

Experiments

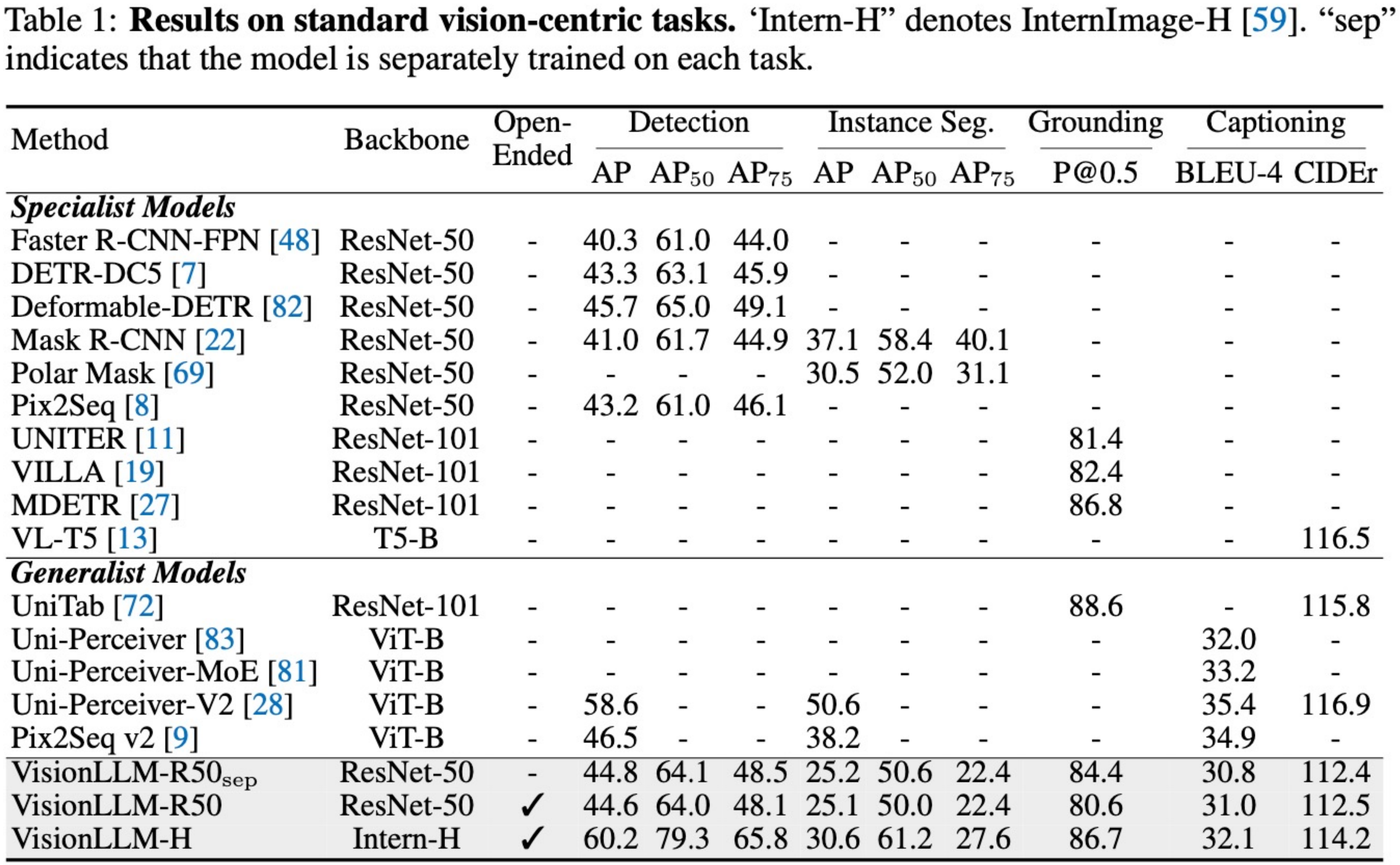


Table 1: **Results on standard vision-centric tasks.** ‘Intern-H” denotes InternImage-H [59]. “sep” indicates that the model is separately trained on each task.

Method	Backbone	Open-Ended	Detection			Instance Seg.			Grounding	Captioning	
			AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	P@0.5	BLEU-4	CIDEr
Specialist Models											
Faster R-CNN-FPN [48]	ResNet-50	-	40.3	61.0	44.0	-	-	-	-	-	-
DETR-DC5 [7]	ResNet-50	-	43.3	63.1	45.9	-	-	-	-	-	-
Deformable-DETR [82]	ResNet-50	-	45.7	65.0	49.1	-	-	-	-	-	-
Mask R-CNN [22]	ResNet-50	-	41.0	61.7	44.9	37.1	58.4	40.1	-	-	-
Polar Mask [69]	ResNet-50	-	-	-	-	30.5	52.0	31.1	-	-	-
Pix2Seq [8]	ResNet-50	-	43.2	61.0	46.1	-	-	-	-	-	-
UNITER [11]	ResNet-101	-	-	-	-	-	-	-	81.4	-	-
VILLA [19]	ResNet-101	-	-	-	-	-	-	-	82.4	-	-
MDETR [27]	ResNet-101	-	-	-	-	-	-	-	86.8	-	-
VL-T5 [13]	T5-B	-	-	-	-	-	-	-	-	-	116.5
Generalist Models											
UniTab [72]	ResNet-101	-	-	-	-	-	-	-	88.6	-	115.8
Uni-Perceiver [83]	ViT-B	-	-	-	-	-	-	-	-	32.0	-
Uni-Perceiver-MoE [81]	ViT-B	-	-	-	-	-	-	-	-	33.2	-
Uni-Perceiver-V2 [28]	ViT-B	-	58.6	-	-	50.6	-	-	-	35.4	116.9
Pix2Seq v2 [9]	ViT-B	-	46.5	-	-	38.2	-	-	-	34.9	-
VisionLLM-$R50_{sep}$	ResNet-50	-	44.8	64.1	48.5	25.2	50.6	22.4	84.4	30.8	112.4
VisionLLM-R50	ResNet-50	✓	44.6	64.0	48.1	25.1	50.0	22.4	80.6	31.0	112.5
VisionLLM-H	Intern-H	✓	60.2	79.3	65.8	30.6	61.2	27.6	86.7	32.1	114.2

Experiments

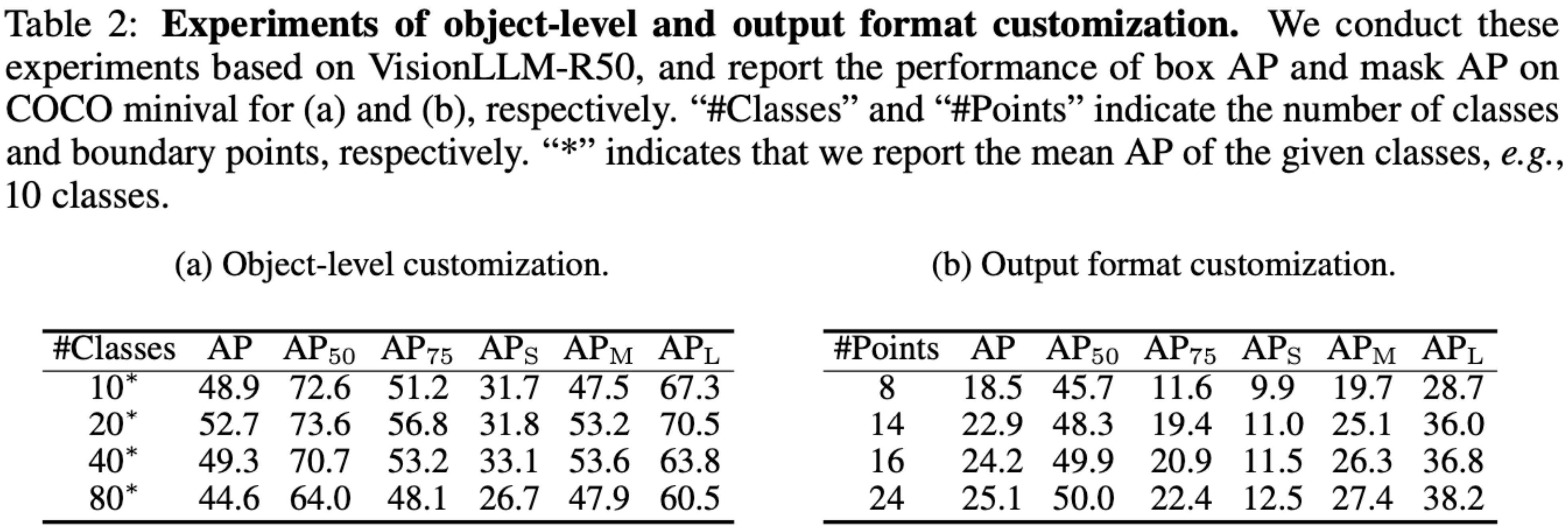


Table 2: **Experiments of object-level and output format customization.** We conduct these experiments based on VisionLLM-R50, and report the performance of box AP and mask AP on COCO minival for (a) and (b), respectively. “#Classes” and “#Points” indicate the number of classes and boundary points, respectively. “*” indicates that we report the mean AP of the given classes, *e.g.*, 10 classes.

(a) Object-level customization.

#Classes	AP	AP_{50}	AP_{75}	AP_{S}	AP_{M}	AP_{L}
10^*	48.9	72.6	51.2	31.7	47.5	67.3
20^*	52.7	73.6	56.8	31.8	53.2	70.5
40^*	49.3	70.7	53.2	33.1	53.6	63.8
80^*	44.6	64.0	48.1	26.7	47.9	60.5

(b) Output format customization.

#Points	AP	AP_{50}	AP_{75}	AP_{S}	AP_{M}	AP_{L}
8	18.5	45.7	11.6	9.9	19.7	28.7
14	22.9	48.3	19.4	11.0	25.1	36.0
16	24.2	49.9	20.9	11.5	26.3	36.8
24	25.1	50.0	22.4	12.5	27.4	38.2

Conclusion

- Designed unified language instruction that matches the format of language models and covers various vision-centric tasks including visual perception.
- Developed a language-guided image tokenizer and an LLM-based task decoder that can handle open-ended tasks according to the given language instructions.
- Verified the effectiveness of the models on a series of tasks with different granularities, demonstrating remarkable generality and flexibility.

Thank you