# Multimodal Language Models

Ji Hyun Kim, Amir Shariatmadari

March 18th, 2024

University of Virginia | ENGINEERING
Department of Computer Science

# Papers

1. Flamingo: a Visual Language Model for Few-Shot Learning
2. VisionLLM: Large Language Model is also an Open-Ended Decoder for Vision-Centric Tasks
3. Visual Instruction Tuning (LLaVA)
4. NExT-GPT: Any-to-Any Multimodal LLM

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

Alayrac et al., 2022

DeepMind

University of Virginia | ENGINEERING
Department of Computer Science

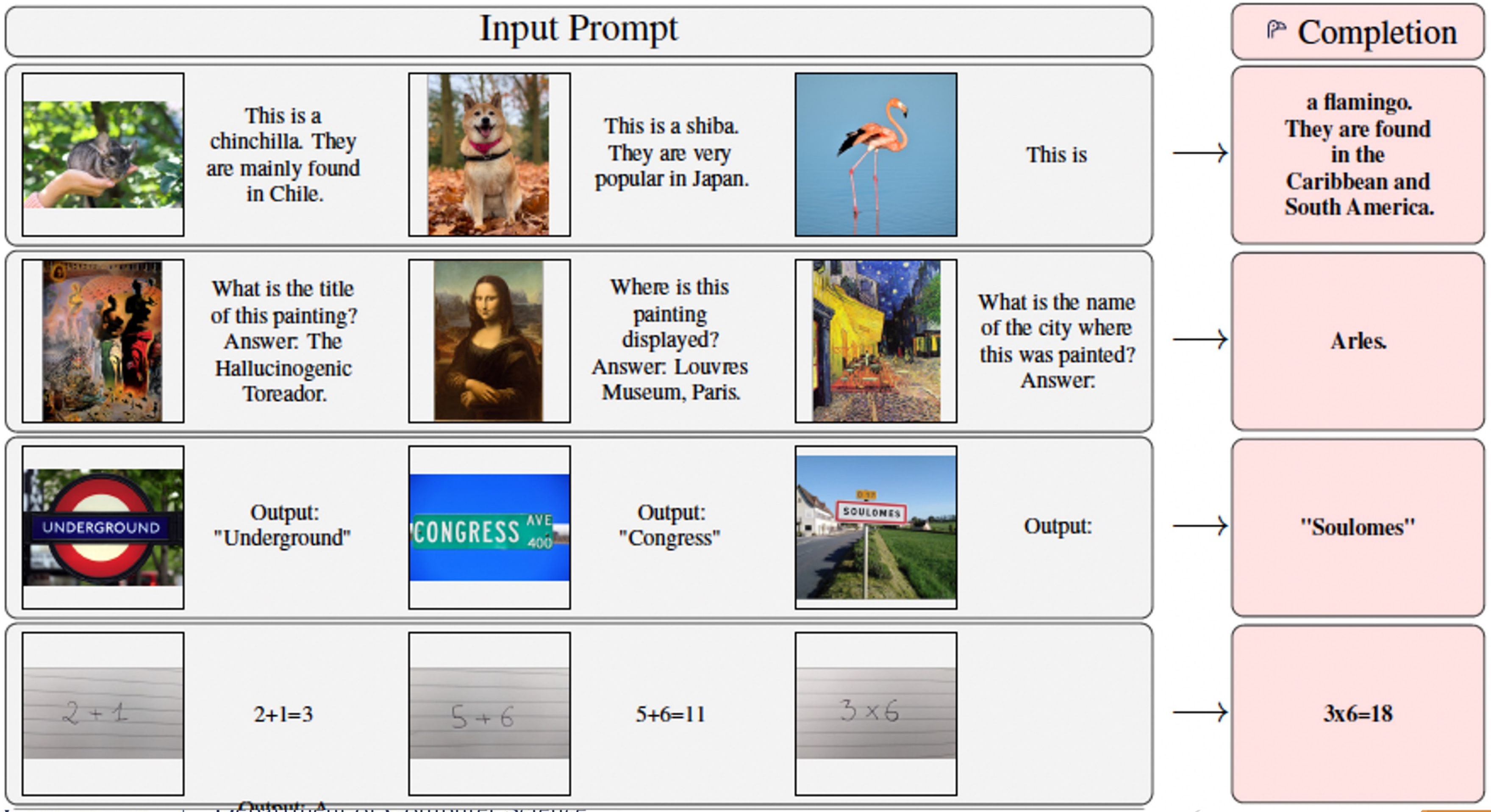
# Motivation

- “One key aspect of intelligence is the ability to quickly learn to perform a new task given a short instruction”
- GPT-3 demonstrates cutting edge performance with few-shot learning.
- Most vision models follow pre-training and fine-tuning paradigm:
    - Needs lots of data
    - Domain/task specific hyperparameter tuning and optimization.
- Multi-modal models trained using contrastive learning demonstrate zero-shot learning capabilities, but their architecture confines them to limited tasks such as classification.

# Motivation

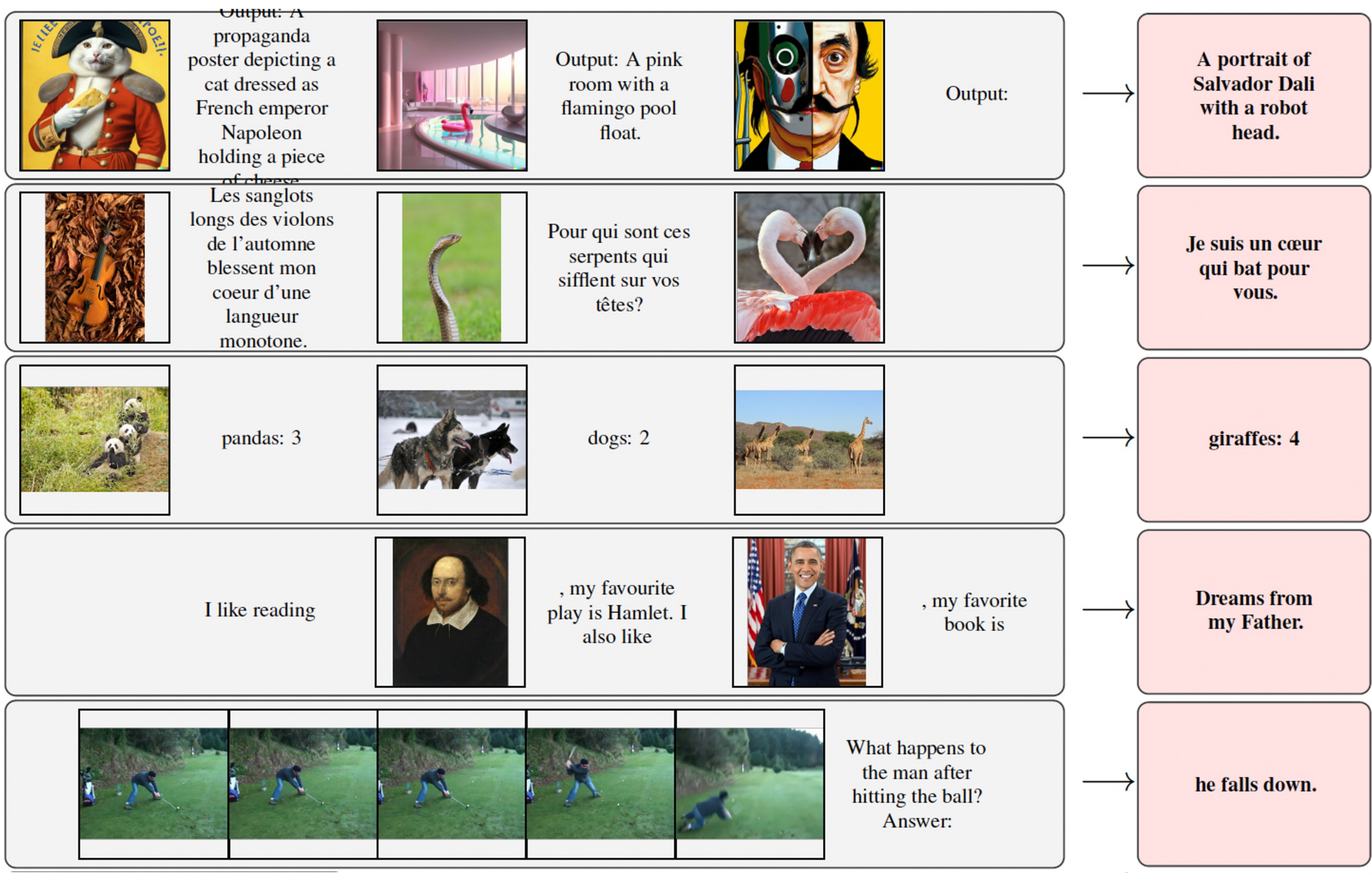
- How can we adopt the few-shot learning capabilities of GPT-3 to multi-modal models?
- How can we make a multi-modal model that is flexible enough to input interleaved text, images, and video and output generated, open ended text?

# What is Flamingo?



| Input Prompt | | | Completion |
|---|---|---|---|
| This is a chinchilla. They are mainly found in Chile. | This is a shiba. They are very popular in Japan. | This is | a flamingo. They are found in the Caribbean and South America. |
| 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" |
| 2+1=3 | 5+6=11 | | 3x6=18 |

# What is Flamingo?



Output: A propaganda poster depicting a cat dressed as French emperor Napoleon holding a piece of cheese. | Output: A pink room with a flamingo pool float. | Output: → **A portrait of Salvador Dali with a robot head.**

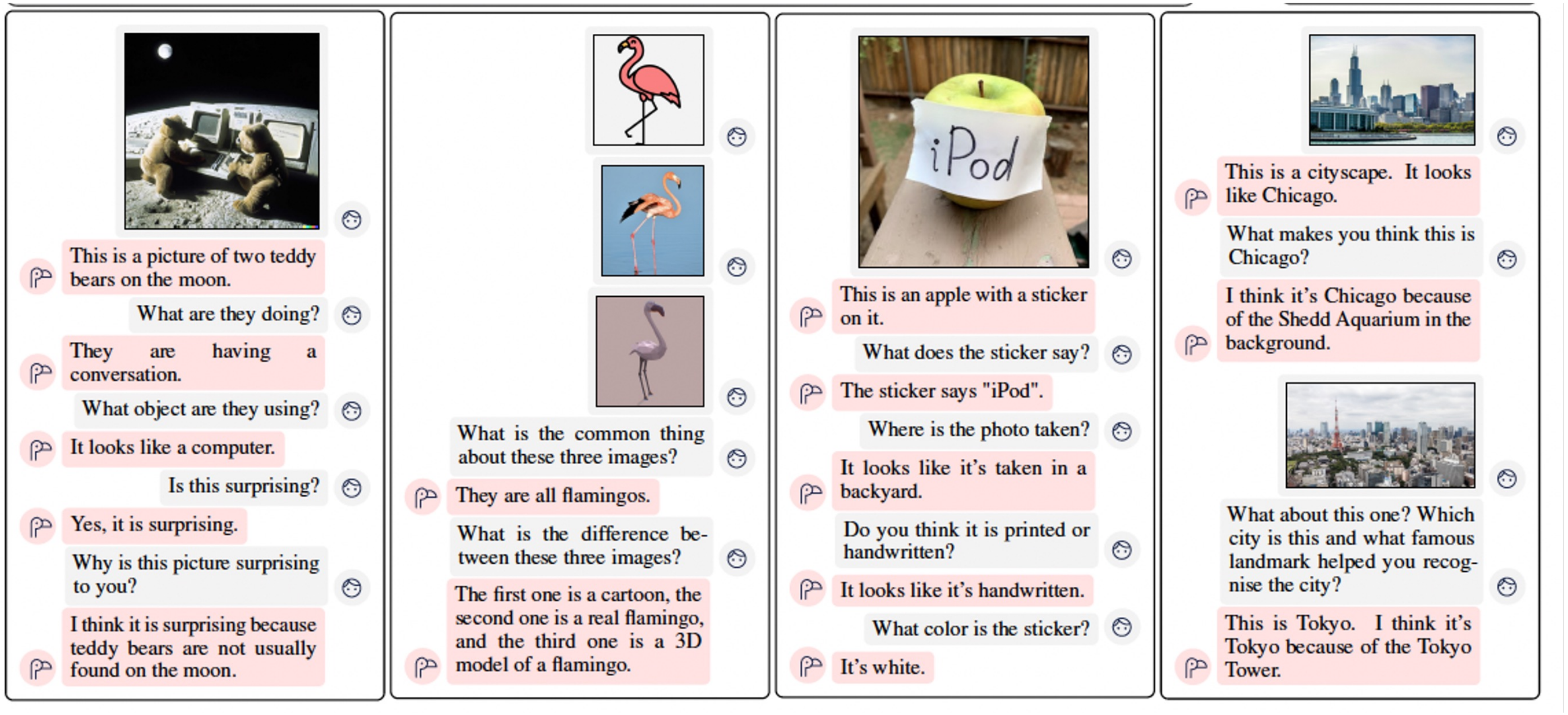
Les sanglots longs des violons de l'automne blessent mon coeur d'une langueur monotone. | Pour qui sont ces serpents qui sifflent sur vos têtes? → **Je suis un cœur qui bat pour vous.**

pandas: 3 | dogs: 2 | → **giraffes: 4**

I like reading | , my favourite play is Hamlet. I also like | , my favorite book is → **Dreams from my Father.**

What happens to the man after hitting the ball? Answer: → **he falls down.**

# What is Flamingo?



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.

# Overview of How Flamingo Works

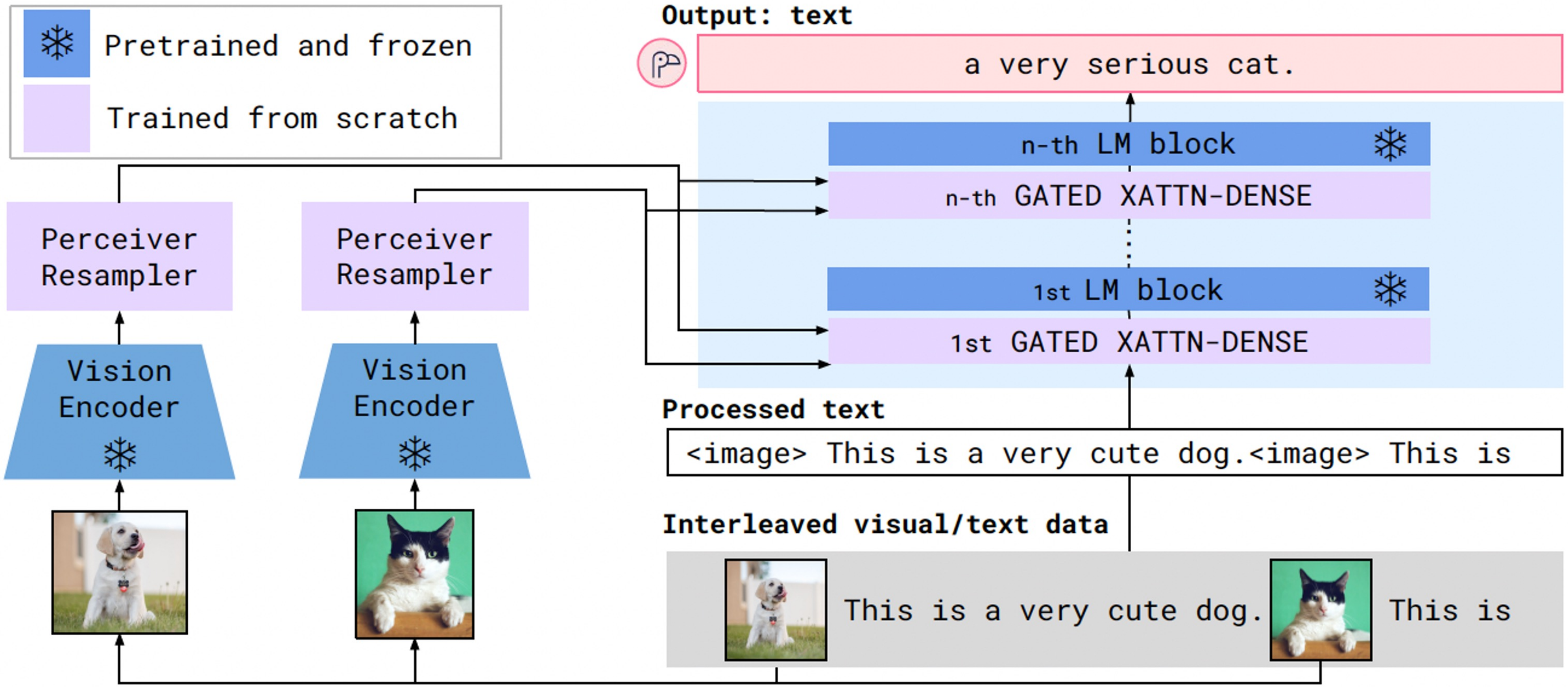


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

# What does Flamingo Model?

- Next token prediction
- What is the likelihood of predicting the text y where “$y\ell$ is the $\ell$-th language token of the input text, $y{<}\ell$ is the set of preceding tokens, $x{\leq}\ell$ is the set of images/videos preceding token $y\ell$ in the interleaved sequence”?

$$p(y|x) = \prod_{\ell=1}^{L} p(y_\ell | y_{<\ell}, x_{\leq \ell}),$$

University of Virginia | ENGINEERING Department of Computer Science

# Vision Encoder

- Inputs text/video
- Outputs 2D image features flattened into 1D sequence of image features or
- 3D Spatial-temporal sequence of video features flattened into 1D
- Pre-trained Normalizer-Free ResNet (frozen)
- pre-trained on contrastive loss objective between text and image pairs.

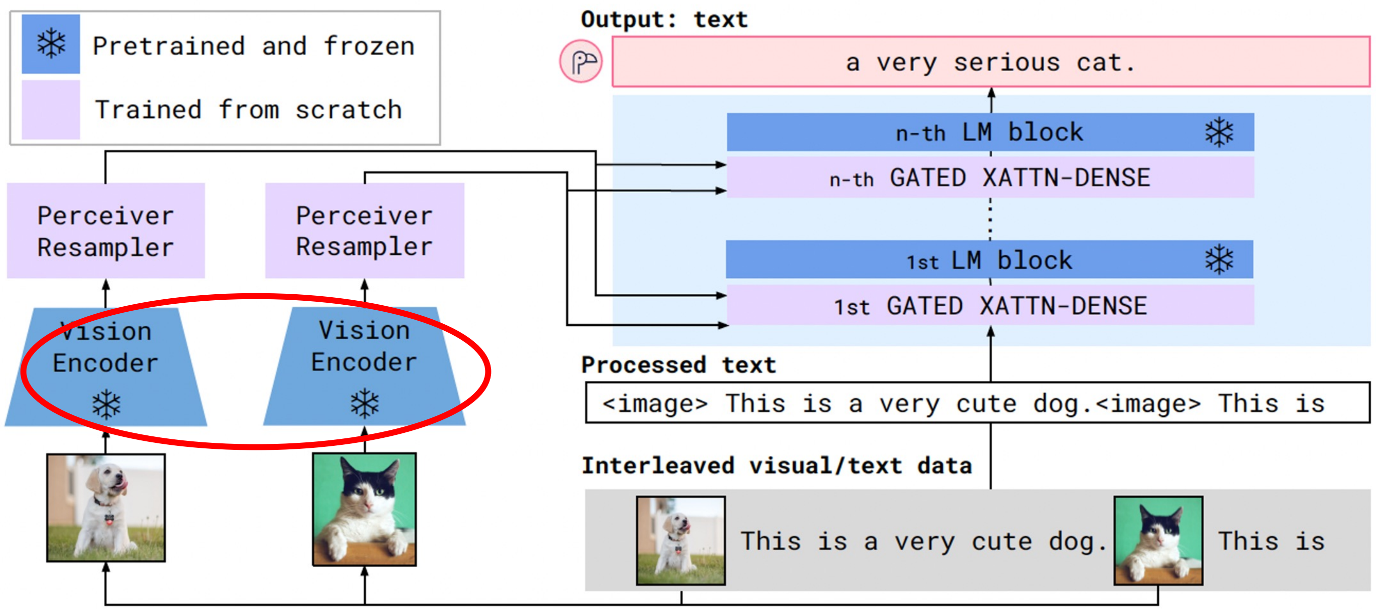


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

# Perceiver Resampler

- Difficult for frozen LM and gated cross-attention dense blocks to take variable length image vectors.
- Bridge between vision encoder and frozen LLM
- Inputs variable length image/video features produced by vision encoder
- Outputs fixed number of visual tokens

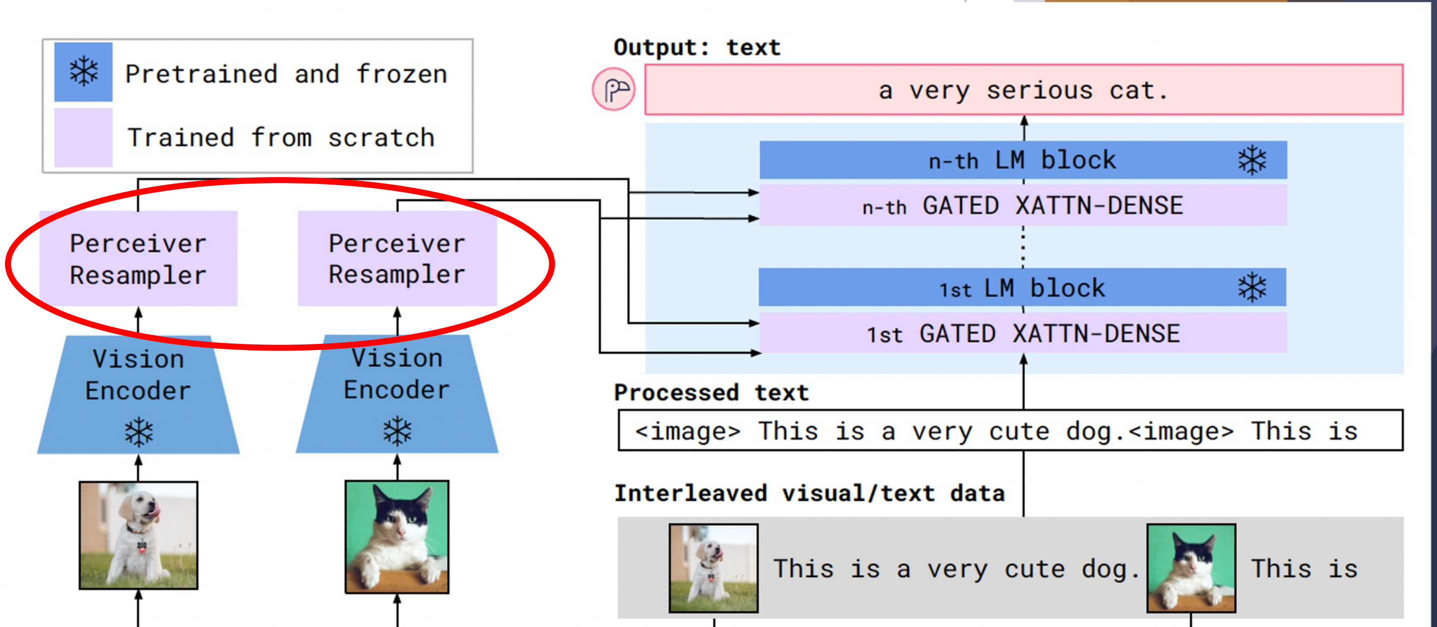
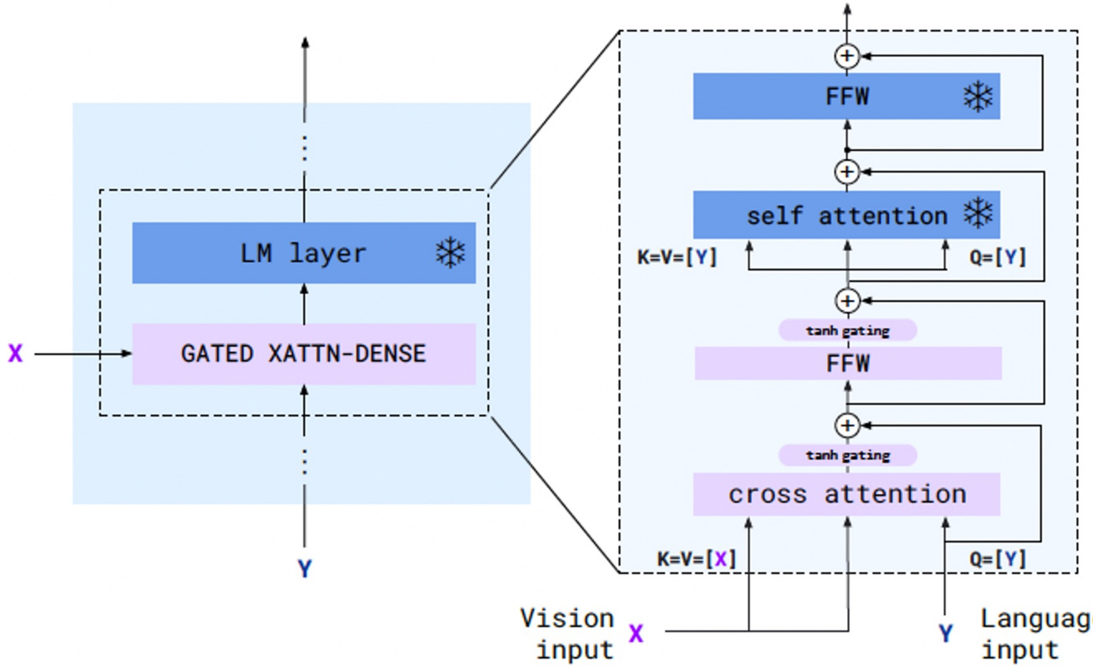
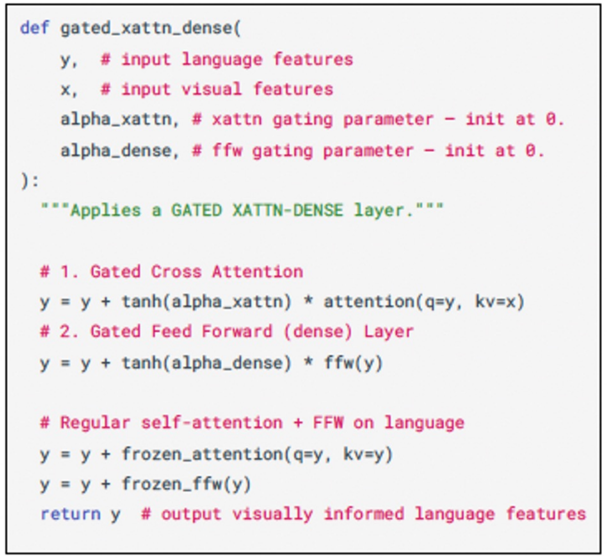


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

# Gated Cross-attention Dense Blocks

- Flamingo adds Gated Cross-attention Dense Blocks before the transformer's (Chinchilla LM) self-attention blocks.

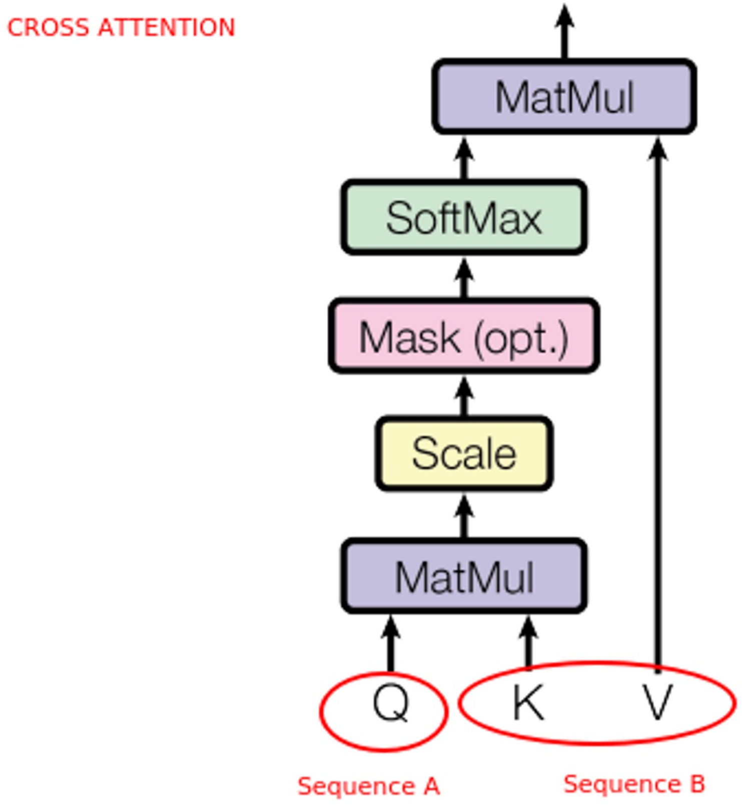




```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter - init at 0.
    alpha_dense, # ffw gating parameter - init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 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
```

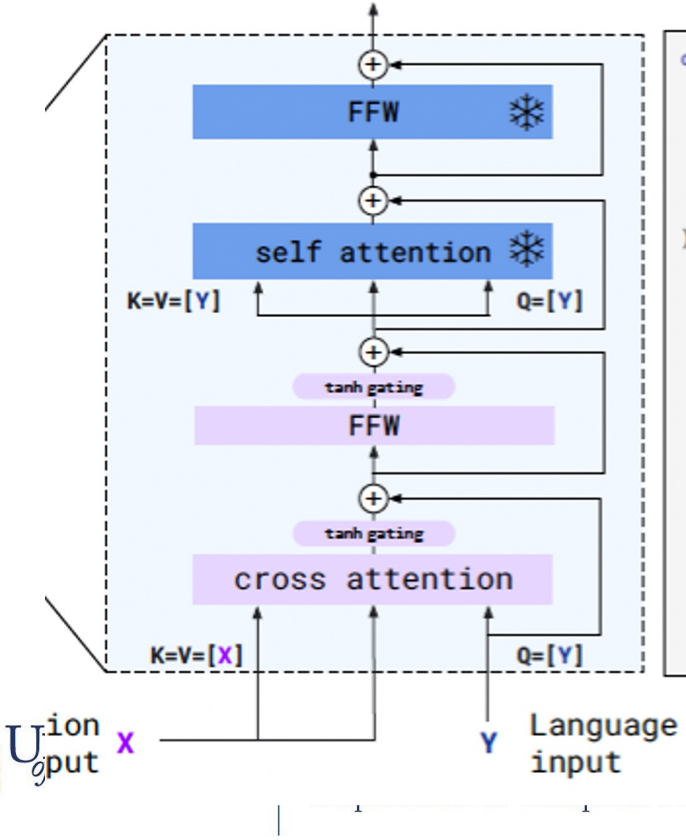
# What is cross-attention?

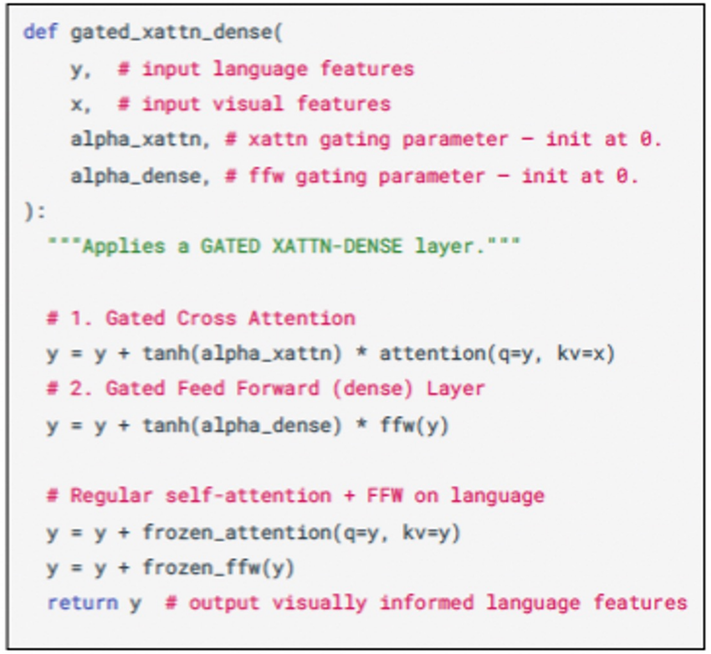
- Queries input text modality.
- Keys and values input vision modality



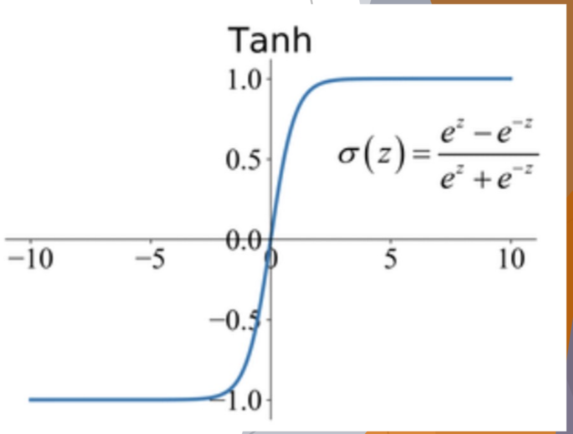
# Gated Cross-attention Dense Blocks

- tanh gating allows for stability in training.
- As tanh gate’s parameter increases, the more cross-attention block as an effect.



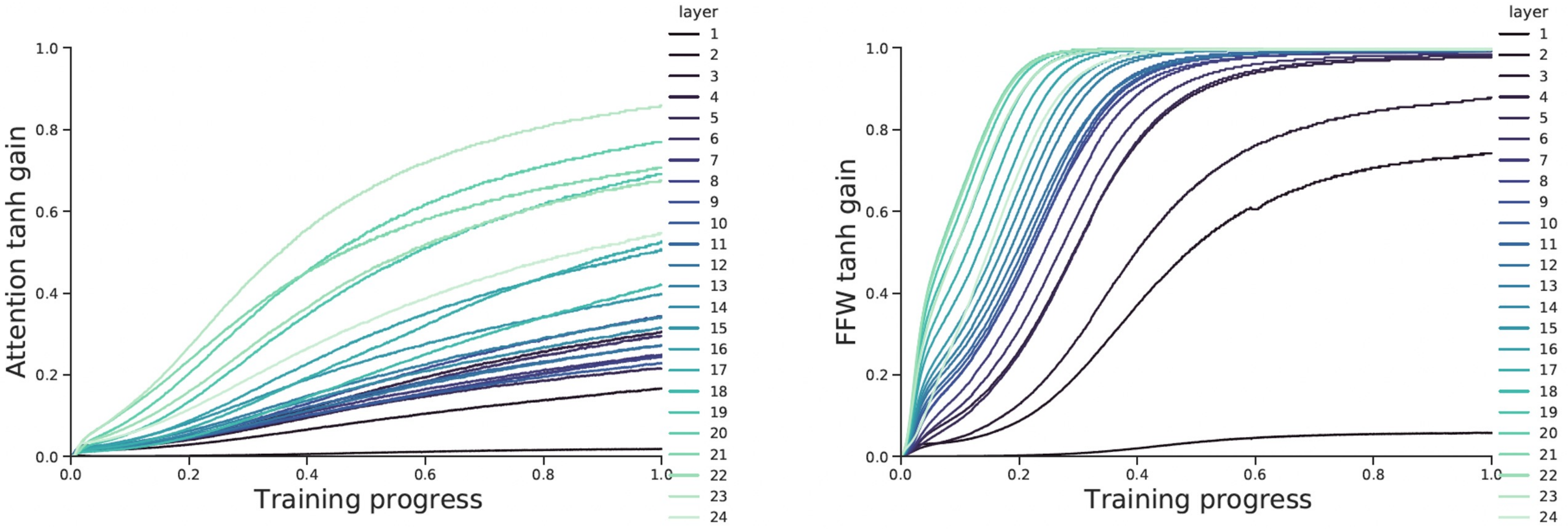


```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter – init at 0.
    alpha_dense, # ffw gating parameter – init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 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
```



# Gated Cross-attention Dense Blocks

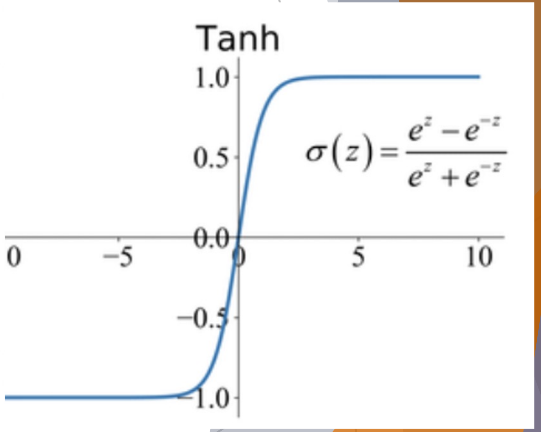
- tanh gating allows for stability in training.



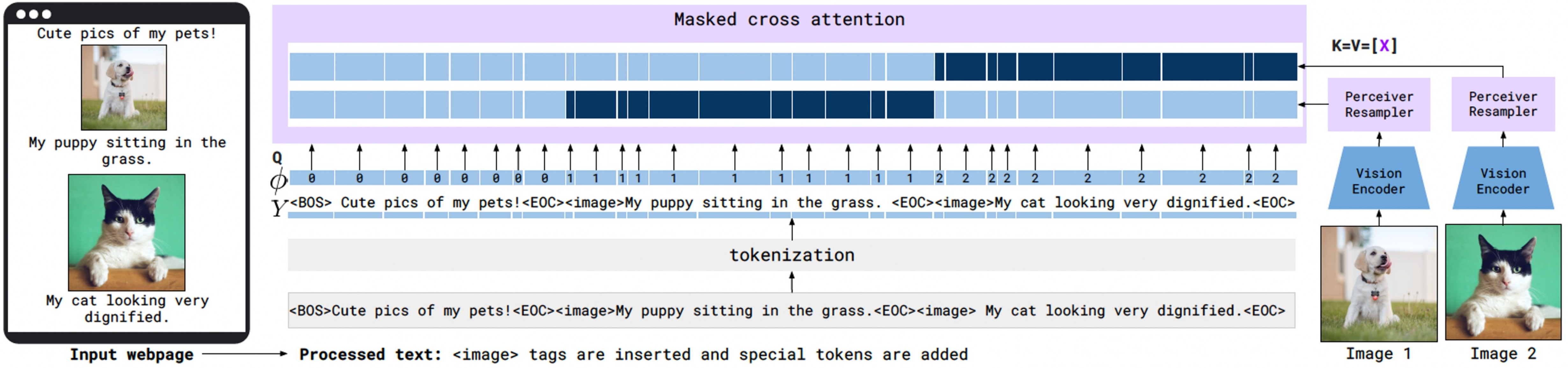
(a) Attention tanh gating

(b) FFW tanh gating.

Figure 6: Evolution of the absolute value of the tanh gating at different layers of *Flamingo*-3B.



# Per-image/video Masking


Cute pics of my pets!
My puppy sitting in the grass.
My cat looking very dignified.
Input webpage
Masked cross attention
K=V=[X]
Perceiver Resampler
Perceiver Resampler
Vision Encoder
Vision Encoder
Image 1
Image 2
Q
ϕ
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
Y
<BOS> Cute pics of my pets!<EOC><image>My puppy sitting in the grass. <EOC><image>My cat looking very dignified.<EOC>
tokenization
<BOS>Cute pics of my pets!<EOC><image>My puppy sitting in the grass.<EOC><image> My cat looking very dignified.<EOC>

**Processed text:** <image> tags are inserted and special tokens are added

# Training Details and Objective

- MultiModal MassiveWeb (M3W) dataset
  - Text and image pairs extracted from HTML of ~43 million webpages
- ALIGN Dataset
  - 1.8 billion images paired with alt-text
  - Complemented with in house dataset of Long Text & Image Pairs which consists of 312 million image and text pairs
- In house dataset of 27 million short videos paired with sentence descriptions
- Multi-objective loss:
  - Next token prediction loss
  - Accumulate weighed sum of losses among the different datasets
  - Datasets are trained together instead of one after another.

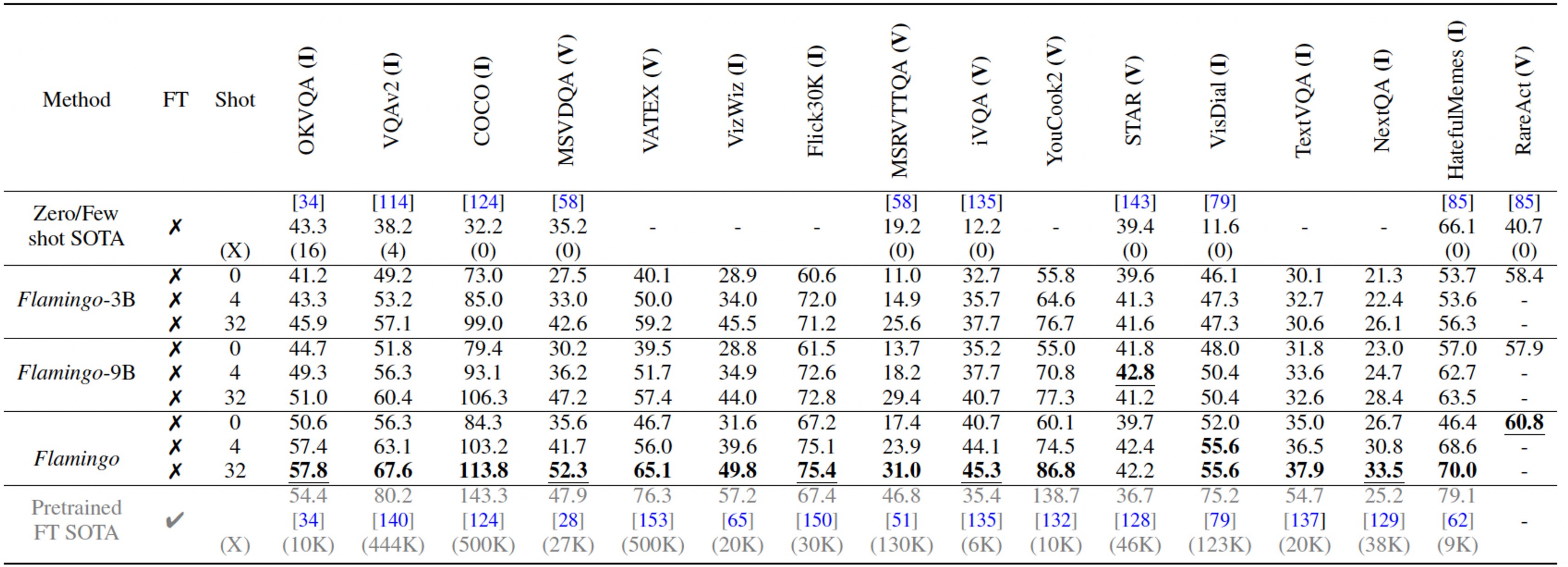
$$\sum_{m=1}^{M} \lambda_m \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}_m}\left[-\sum_{\ell=1}^{L}\log p(y_\ell|y_{<\ell}, x_{\leq\ell})\right],$$

# Experiments: Benchmarks

- 16 Benchmarks
  - 9 image benchmarks
  - 7 video benchmarks

# Experiments: Zero and Few Shot Results

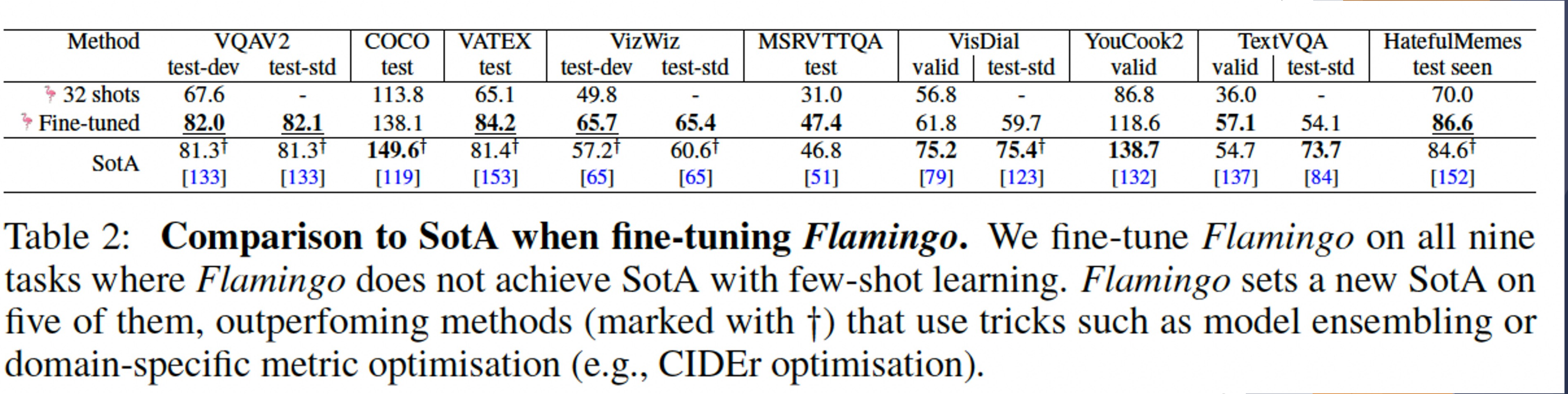
- 16 Benchmarks
  - 9 image benchmarks
  - 7 video benchmarks
- Beats SOTA with zero or few shots on many benchmarks.



| Method | FT | Shot | OKVQA (I) | VQAv2 (I) | COCO (I) | MSVDQA (V) | VATEX (V) | VizWiz (I) | Flick30K (I) | MSRVTTQA (V) | iVQA (V) | YouCook2 (V) | STAR (V) | VisDial (I) | TextVQA (I) | NextQA (I) | HatefulMemes (I) | RareAct (V) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Zero/Few shot SOTA | ✗ | (X) | [34]<br>43.3<br>(16) | [114]<br>38.2<br>(4) | [124]<br>32.2<br>(0) | [58]<br>35.2<br>(0) | - | - | - | [58]<br>19.2<br>(0) | [135]<br>12.2<br>(0) | - | [143]<br>39.4<br>(0) | [79]<br>11.6<br>(0) | - | - | [85]<br>66.1<br>(0) | [85]<br>40.7<br>(0) |
| *Flamingo*-3B | ✗ | 0 | 41.2 | 49.2 | 73.0 | 27.5 | 40.1 | 28.9 | 60.6 | 11.0 | 32.7 | 55.8 | 39.6 | 46.1 | 30.1 | 21.3 | 53.7 | 58.4 |
| | ✗ | 4 | 43.3 | 53.2 | 85.0 | 33.0 | 50.0 | 34.0 | 72.0 | 14.9 | 35.7 | 64.6 | 41.3 | 47.3 | 32.7 | 22.4 | 53.6 | - |
| | ✗ | 32 | 45.9 | 57.1 | 99.0 | 42.6 | 59.2 | 45.5 | 71.2 | 25.6 | 37.7 | 76.7 | 41.6 | 47.3 | 30.6 | 26.1 | 56.3 | - |
| *Flamingo*-9B | ✗ | 0 | 44.7 | 51.8 | 79.4 | 30.2 | 39.5 | 28.8 | 61.5 | 13.7 | 35.2 | 55.0 | 41.8 | 48.0 | 31.8 | 23.0 | 57.0 | 57.9 |
| | ✗ | 4 | 49.3 | 56.3 | 93.1 | 36.2 | 51.7 | 34.9 | 72.6 | 18.2 | 37.7 | 70.8 | **42.8** | 50.4 | 33.6 | 24.7 | 62.7 | - |
| | ✗ | 32 | 51.0 | 60.4 | 106.3 | 47.2 | 57.4 | 44.0 | 72.8 | 29.4 | 40.7 | 77.3 | 41.2 | 50.4 | 32.6 | 28.4 | 63.5 | - |
| *Flamingo* | ✗ | 0 | 50.6 | 56.3 | 84.3 | 35.6 | 46.7 | 31.6 | 67.2 | 17.4 | 40.7 | 60.1 | 39.7 | 52.0 | 35.0 | 26.7 | 46.4 | **60.8** |
| | ✗ | 4 | 57.4 | 63.1 | 103.2 | 41.7 | 56.0 | 39.6 | 75.1 | 23.9 | 44.1 | 74.5 | 42.4 | **55.6** | 36.5 | 30.8 | 68.6 | - |
| | ✗ | 32 | **57.8** | **67.6** | **113.8** | **52.3** | **65.1** | **49.8** | **75.4** | **31.0** | **45.3** | **86.8** | 42.2 | **55.6** | **37.9** | **33.5** | **70.0** | - |
| Pretrained FT SOTA | ✔ | (X) | 54.4<br>[34]<br>(10K) | 80.2<br>[140]<br>(444K) | 143.3<br>[124]<br>(500K) | 47.9<br>[28]<br>(27K) | 76.3<br>[153]<br>(500K) | 57.2<br>[65]<br>(20K) | 67.4<br>[150]<br>(30K) | 46.8<br>[51]<br>(130K) | 35.4<br>[135]<br>(6K) | 138.7<br>[132]<br>(10K) | 36.7<br>[128]<br>(46K) | 75.2<br>[79]<br>(123K) | 54.7<br>[137]<br>(20K) | 25.2<br>[129]<br>(38K) | 79.1<br>[62]<br>(9K) | - |

# Experiments: Fine-tuning results

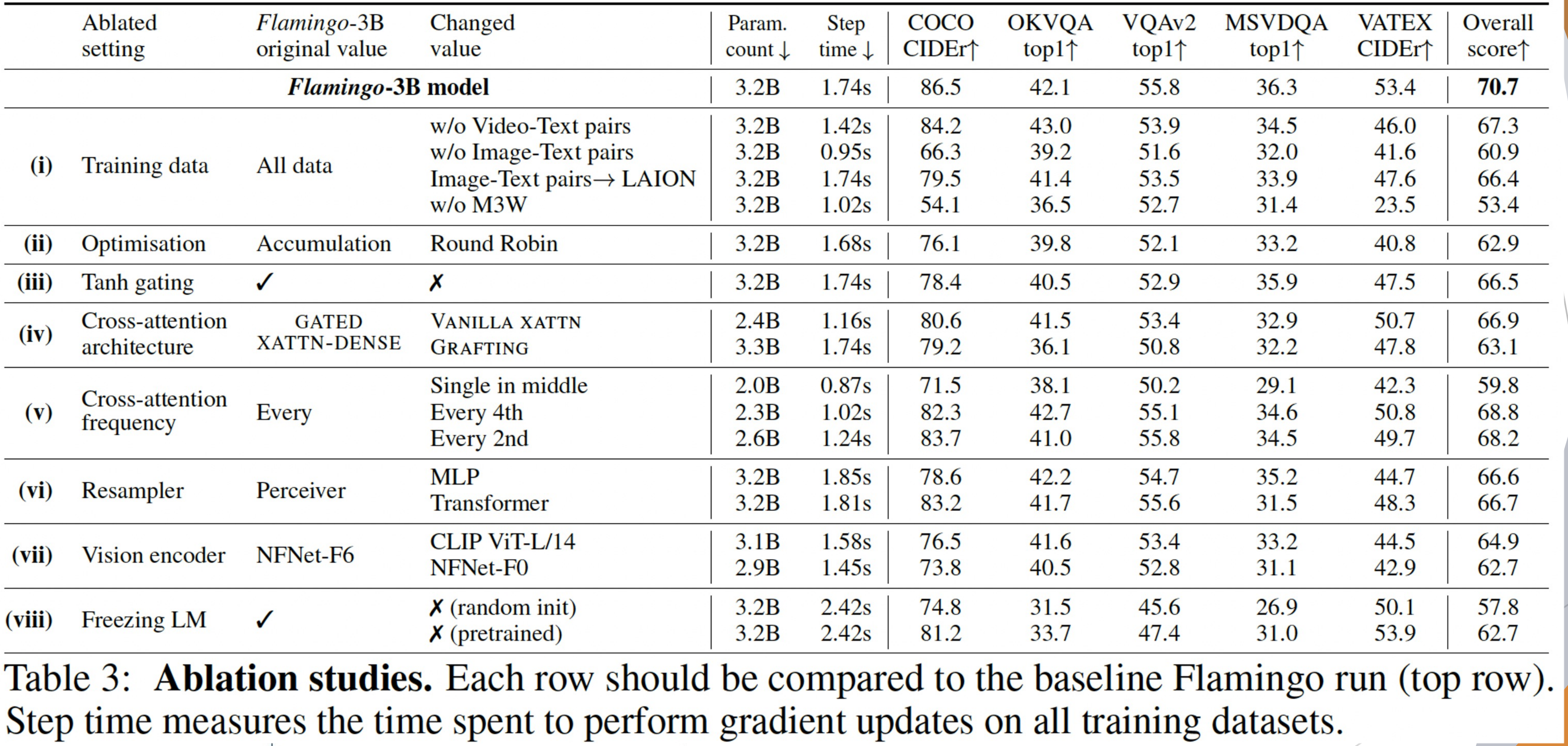
- Fine-tuning can improve Flamingo performance compared to few-shot learning.
- Some SOTA models still slightly perform better than Flamingo on a few benchmarks.



| Method | VQAV2 | | COCO | VATEX | VizWiz | | MSRVTTQA | VisDial | | YouCook2 | TextVQA | | HatefulMemes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | test-dev | test-std | test | test | test-dev | test-std | test | valid | test-std | valid | valid | test-std | test seen |
| 🦩 32 shots | 67.6 | - | 113.8 | 65.1 | 49.8 | - | 31.0 | 56.8 | - | 86.8 | 36.0 | - | 70.0 |
| 🦩 Fine-tuned | **82.0** | **82.1** | 138.1 | **84.2** | **65.7** | **65.4** | **47.4** | 61.8 | 59.7 | 118.6 | **57.1** | 54.1 | **86.6** |
| SotA | 81.3†<br>[133] | 81.3†<br>[133] | **149.6**†<br>[119] | 81.4†<br>[153] | 57.2†<br>[65] | 60.6†<br>[65] | 46.8<br>[51] | **75.2**<br>[79] | **75.4**†<br>[123] | **138.7**<br>[132] | 54.7<br>[137] | **73.7**<br>[84] | 84.6†<br>[152] |

Table 2: **Comparison to SotA when fine-tuning *Flamingo*.** We fine-tune *Flamingo* on all nine tasks where *Flamingo* does not achieve SotA with few-shot learning. *Flamingo* sets a new SotA on five of them, outperfoming methods (marked with †) that use tricks such as model ensembling or domain-specific metric optimisation (e.g., CIDEr optimisation).

# Ablation Study:



| | Ablated setting | *Flamingo*-3B original value | Changed value | Param. count ↓ | Step time ↓ | COCO CIDEr↑ | OKVQA top1↑ | VQAv2 top1↑ | MSVDQA top1↑ | VATEX CIDEr↑ | Overall score↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **_Flamingo_-3B model** | | 3.2B | 1.74s | 86.5 | 42.1 | 55.8 | 36.3 | 53.4 | **70.7** |
| **(i)** | Training data | All data | w/o Video-Text pairs | 3.2B | 1.42s | 84.2 | 43.0 | 53.9 | 34.5 | 46.0 | 67.3 |
| | | | w/o Image-Text pairs | 3.2B | 0.95s | 66.3 | 39.2 | 51.6 | 32.0 | 41.6 | 60.9 |
| | | | Image-Text pairs→ LAION | 3.2B | 1.74s | 79.5 | 41.4 | 53.5 | 33.9 | 47.6 | 66.4 |
| | | | w/o M3W | 3.2B | 1.02s | 54.1 | 36.5 | 52.7 | 31.4 | 23.5 | 53.4 |
| **(ii)** | Optimisation | Accumulation | Round Robin | 3.2B | 1.68s | 76.1 | 39.8 | 52.1 | 33.2 | 40.8 | 62.9 |
| **(iii)** | Tanh gating | ✓ | ✗ | 3.2B | 1.74s | 78.4 | 40.5 | 52.9 | 35.9 | 47.5 | 66.5 |
| **(iv)** | Cross-attention architecture | GATED XATTN-DENSE | VANILLA XATTN | 2.4B | 1.16s | 80.6 | 41.5 | 53.4 | 32.9 | 50.7 | 66.9 |
| | | | GRAFTING | 3.3B | 1.74s | 79.2 | 36.1 | 50.8 | 32.2 | 47.8 | 63.1 |
| **(v)** | Cross-attention frequency | Every | Single in middle | 2.0B | 0.87s | 71.5 | 38.1 | 50.2 | 29.1 | 42.3 | 59.8 |
| | | | Every 4th | 2.3B | 1.02s | 82.3 | 42.7 | 55.1 | 34.6 | 50.8 | 68.8 |
| | | | Every 2nd | 2.6B | 1.24s | 83.7 | 41.0 | 55.8 | 34.5 | 49.7 | 68.2 |
| **(vi)** | Resampler | Perceiver | MLP | 3.2B | 1.85s | 78.6 | 42.2 | 54.7 | 35.2 | 44.7 | 66.6 |
| | | | Transformer | 3.2B | 1.81s | 83.2 | 41.7 | 55.6 | 31.5 | 48.3 | 66.7 |
| **(vii)** | Vision encoder | NFNet-F6 | CLIP ViT-L/14 | 3.1B | 1.58s | 76.5 | 41.6 | 53.4 | 33.2 | 44.5 | 64.9 |
| | | | NFNet-F0 | 2.9B | 1.45s | 73.8 | 40.5 | 52.8 | 31.1 | 42.9 | 62.7 |
| **(viii)** | Freezing LM | ✓ | ✗ (random init) | 3.2B | 2.42s | 74.8 | 31.5 | 45.6 | 26.9 | 50.1 | 57.8 |
| | | | ✗ (pretrained) | 3.2B | 2.42s | 81.2 | 33.7 | 47.4 | 31.0 | 53.9 | 62.7 |

Table 3: **Ablation studies.** Each row should be compared to the baseline Flamingo run (top row). Step time measures the time spent to perform gradient updates on all training datasets.

# Ablation Study:

- Training data as an interleaved mixture is important.
  - 17% performance increase
- Tahn gate helps get rid of training instability.
- Inserting gated cross-attention dense blocks only every 4th layer increases computational efficiency by 66% with only a performance loss of 1.9%
- Keeping LLM frozen prevents catastrophic forgetting.

# Limitations

- Flamingo inherits issues from pretrained LLM
  - Hallucination.
  - Poor generalization to sequences longer than training data.
  - Sample inefficient during training (needs lots of examples to learn)
- Flamingo doesn’t perform as well as SOTA on classification tasks
- Flamingo inherits the flaws of in-context learning:
  - Highly sensitive to certain aspects of examples
  - Cost of inference and performance scale poorly with the number of shots.

# Conclusion

- Flamingo is a general purpose, open-ended, multi-modal model meant for image-language and video-language tasks.

- Flamingo can beat SOTA performance on a variety of tasks with few-shots of data.

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

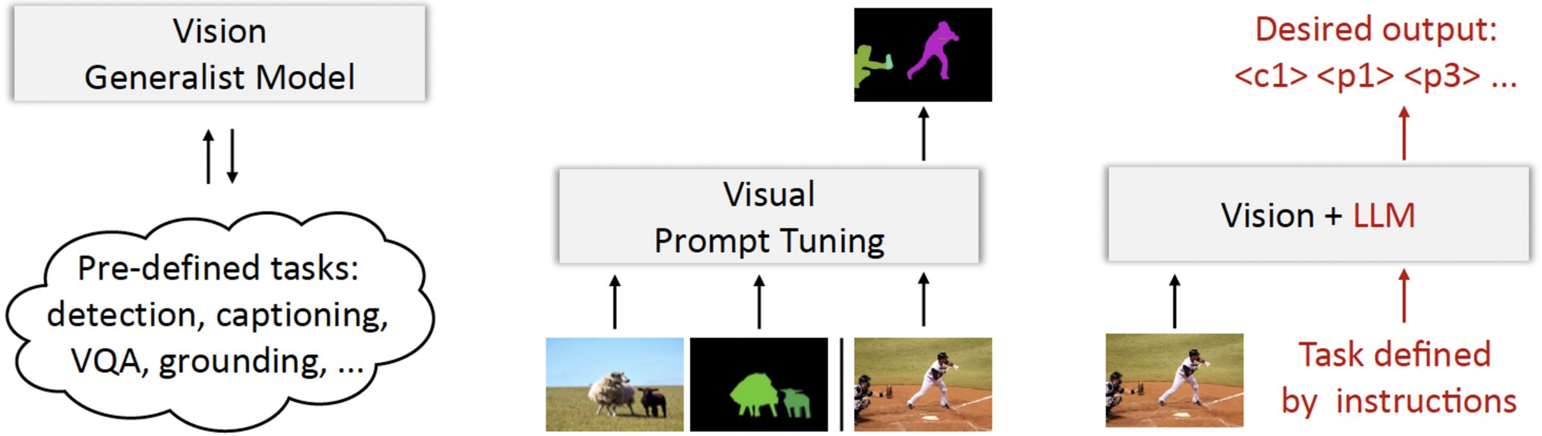
Wang et. al, 2023

OpenGVLab, Shanghai AI Laboratory

University of Virginia | ENGINEERING Department of Computer Science

# Motivation

- How can we adopt the versatility and flexibility of LLMs like GPT-3 to the vision domain?
- Can we have an open ended language vision model that can also perform on vision-centric tasks?



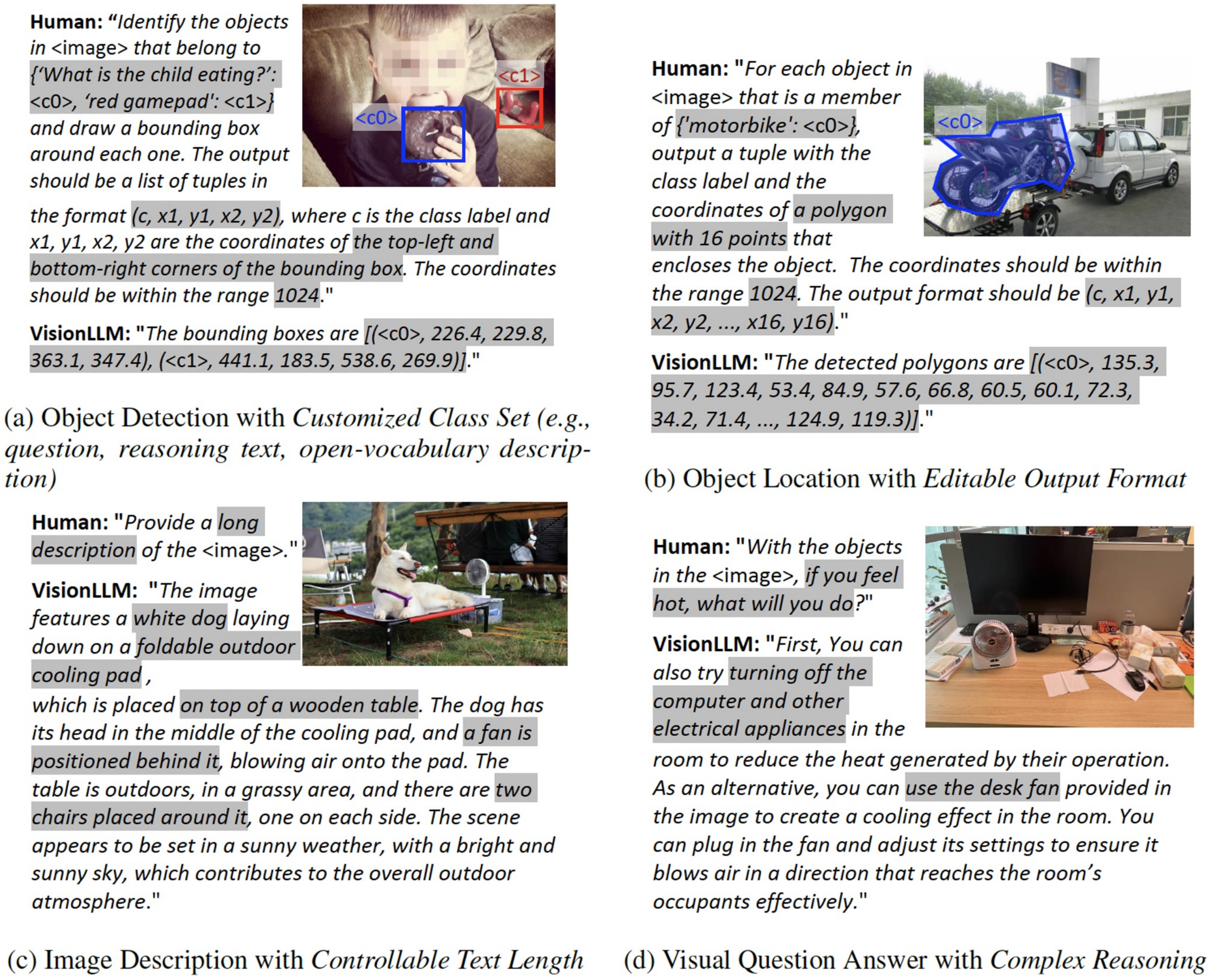
(a) Vision generalist models [59, 61, 83] are constrained by the format of pre-defined tasks.

(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.

# What is VisionLLM?



**Human:** *"Identify the objects in <image> that belong to {'What is the child eating?': <c0>, 'red gamepad': <c1>} and draw a bounding box around each one. The output should be a list of tuples in the format (c, x1, y1, x2, y2), where c is the class label and x1, y1, x2, y2 are the coordinates of the top-left and bottom-right corners of the bounding box. The coordinates should be within the range 1024."*

**VisionLLM:** *"The bounding boxes are [(<c0>, 226.4, 229.8, 363.1, 347.4), (<c1>, 441.1, 183.5, 538.6, 269.9)]."*

(a) Object Detection with *Customized Class Set (e.g., question, reasoning text, open-vocabulary description)*

**Human:** *"For each object in <image> that is a member of {'motorbike': <c0>}, 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 the range 1024. The output format should be (c, x1, y1, x2, y2, ..., x16, y16)."*

**VisionLLM:** *"The detected polygons are [(<c0>, 135.3, 95.7, 123.4, 53.4, 84.9, 57.6, 66.8, 60.5, 60.1, 72.3, 34.2, 71.4, ..., 124.9, 119.3)]."*

(b) Object Location with *Editable Output Format*

**Human:** *"Provide a long description of the <image>."*

**VisionLLM:** *"The image features a white dog laying down on a foldable outdoor cooling pad , which is placed on top of a wooden table. The dog has its head in the middle of the cooling pad, and a fan is positioned behind it, blowing air onto the pad. The table is outdoors, in a grassy area, and there are two chairs placed around it, one on each side. The scene appears to be set in a sunny weather, with a bright and sunny sky, which contributes to the overall outdoor atmosphere."*

(c) Image Description with *Controllable Text Length*

**Human:** *"With the objects in the <image>, if you feel hot, what will you do?"*

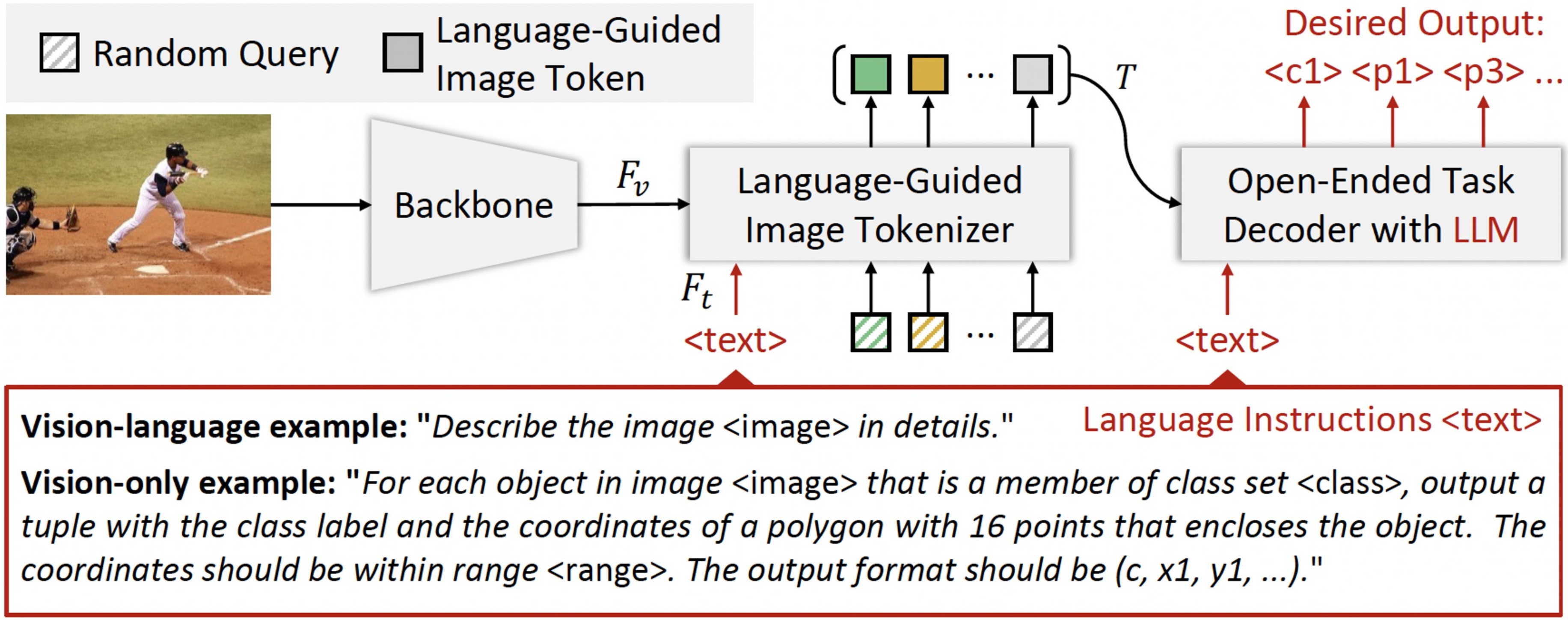
**VisionLLM:** *"First, You can also try turning off the computer and other electrical appliances in the room to reduce the heat generated by their operation. As an alternative, you can use the desk fan provided in the image to create a cooling effect in the room. You can plug in the fan and adjust its settings to ensure it blows air in a direction that reaches the room's occupants effectively."*

(d) Visual Question Answer with *Complex Reasoning*

# What is VisionLLM?

- It’s a framework that aligns vision tasks with LLM methodologies.
- As a result, VisionLLM is able to perform in an open ended manner on vision-centric tasks on a level that is on par with its specialized model counterparts.

# Overview of How VisionLLM Works


▨ 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.*"

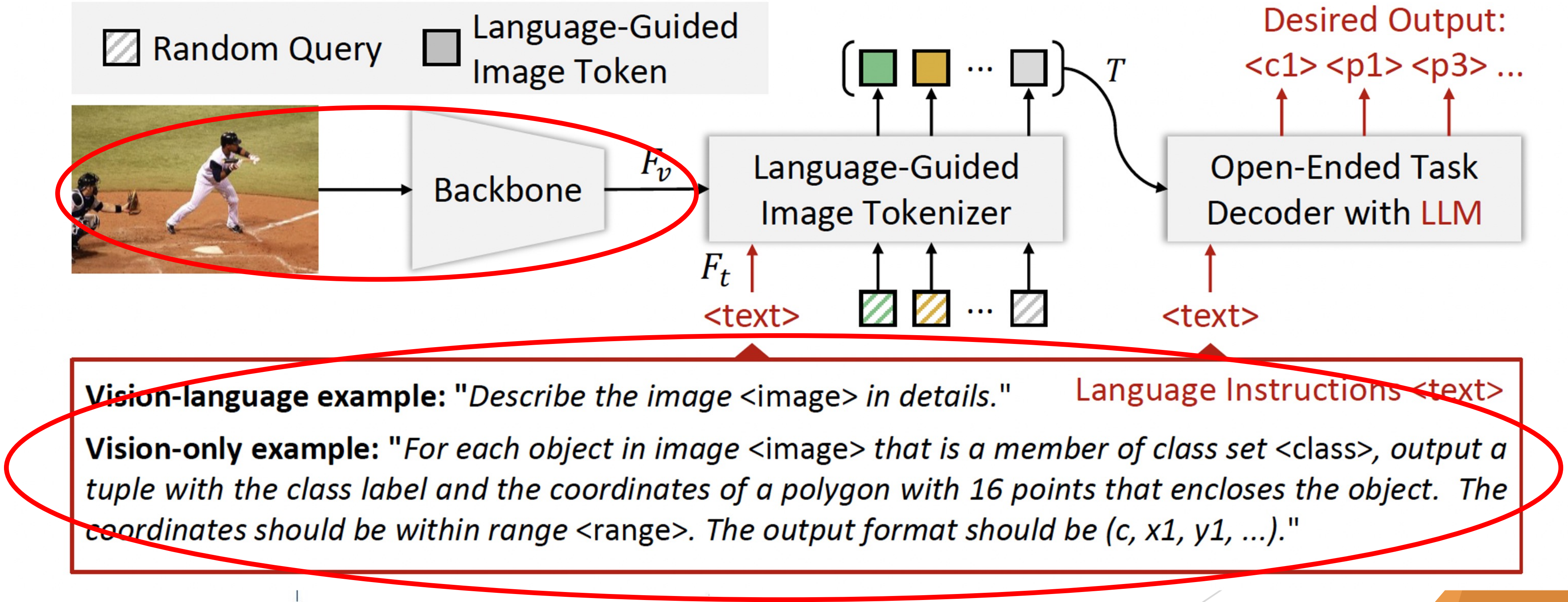
**Vision-only example:** "*For each object in image* <image> *that is a member of class set* <class>, *output a tuple with the class label and the coordinates of a polygon with 16 points that encloses the object. The coordinates should be within range* <range>. *The output format should be (c, x1, y1,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

Figure 3: **Overall architecture of the proposed VisionLLM.** It consists of three parts: a unified language instruction designed to accommodate both vision and vision-language tasks, an image tokenizer that encodes visual information guided by language instructions, and an LLM-based open-ended task decoder that executes diverse tasks defined by language instructions.

# Overview of How VisionLLM Works

3 parts:

1. “a unified language instruction designed to accommodate both vision and vision-language tasks”


▨ 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>

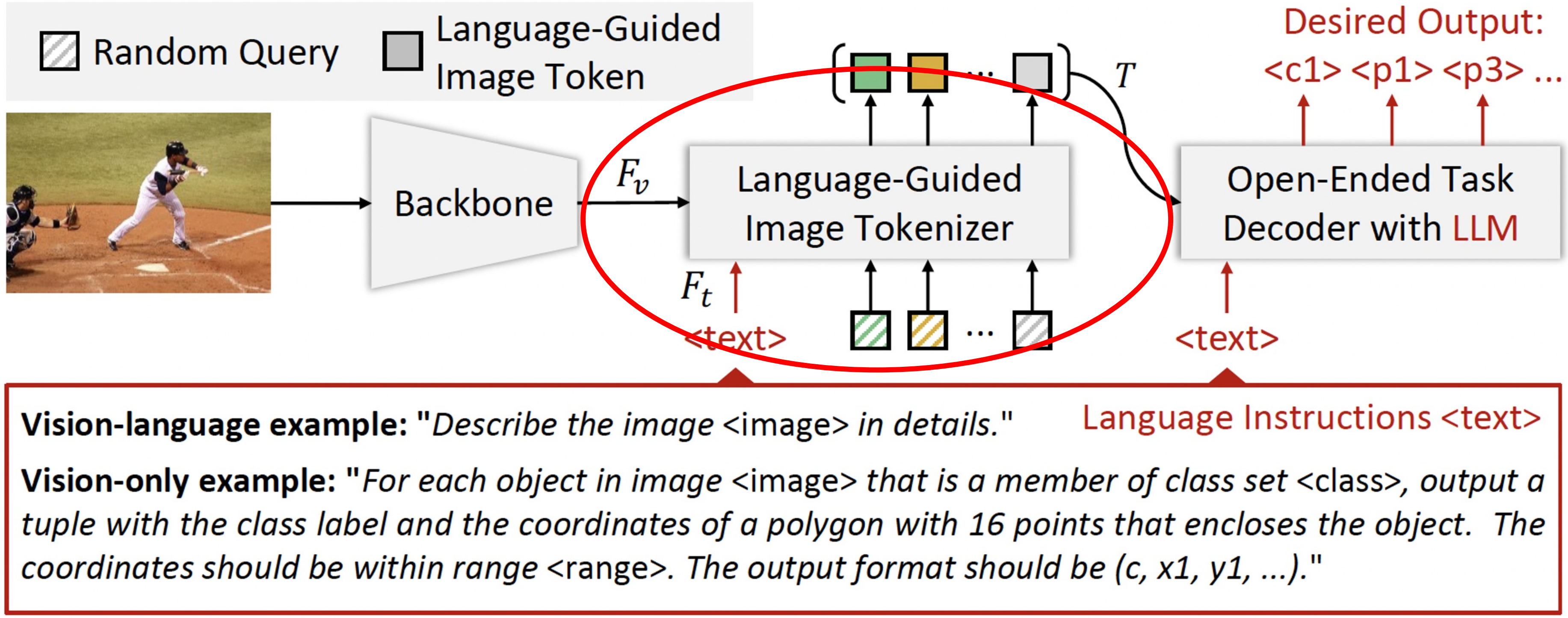
**Vision-language example:** "*Describe the image* <image> *in details.*"

**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, ...).*"

# Overview of How VisionLLM Works

3 parts:

2. “an image tokenizer that encodes visual information guided by language instructions”



▨ 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>

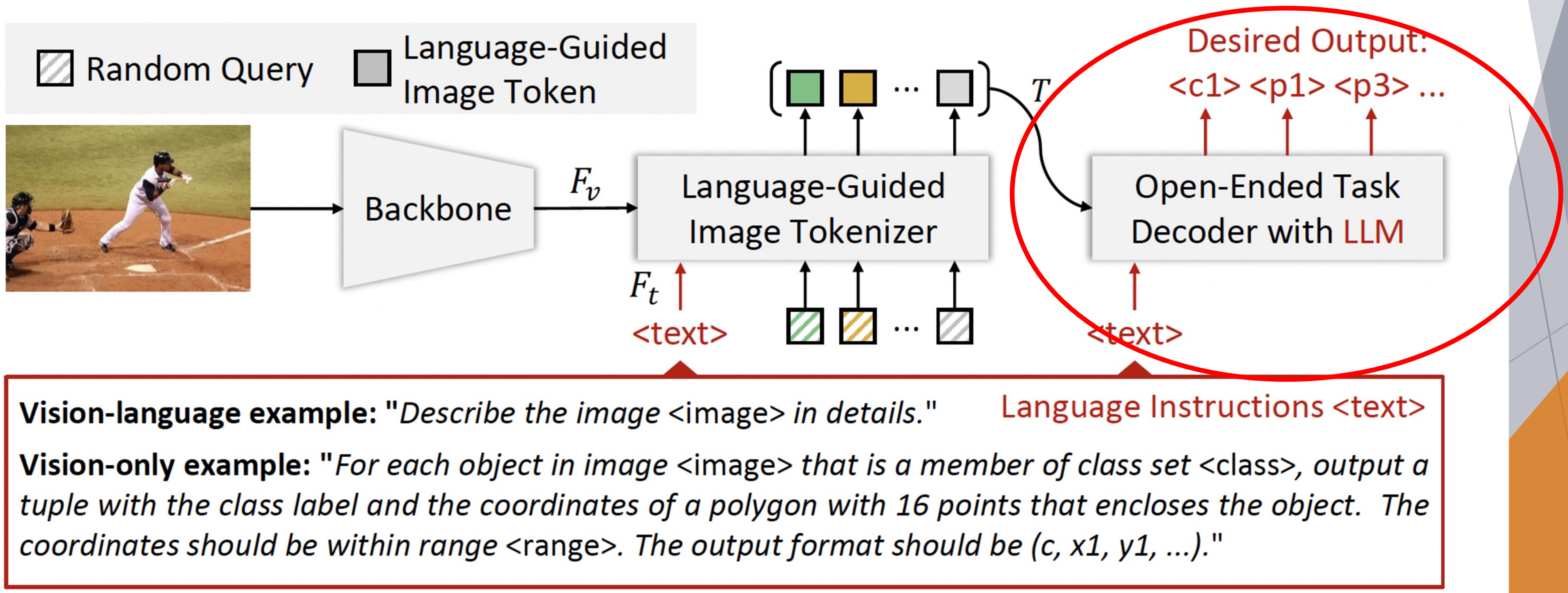
**Vision-language example:** "*Describe the image* <image> *in details.*"

**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, ...).*"

# Overview of How VisionLLM Works

3 parts:

3. "an LLM-based open-ended task decoder that executes diverse tasks defined by language instructions"


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

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

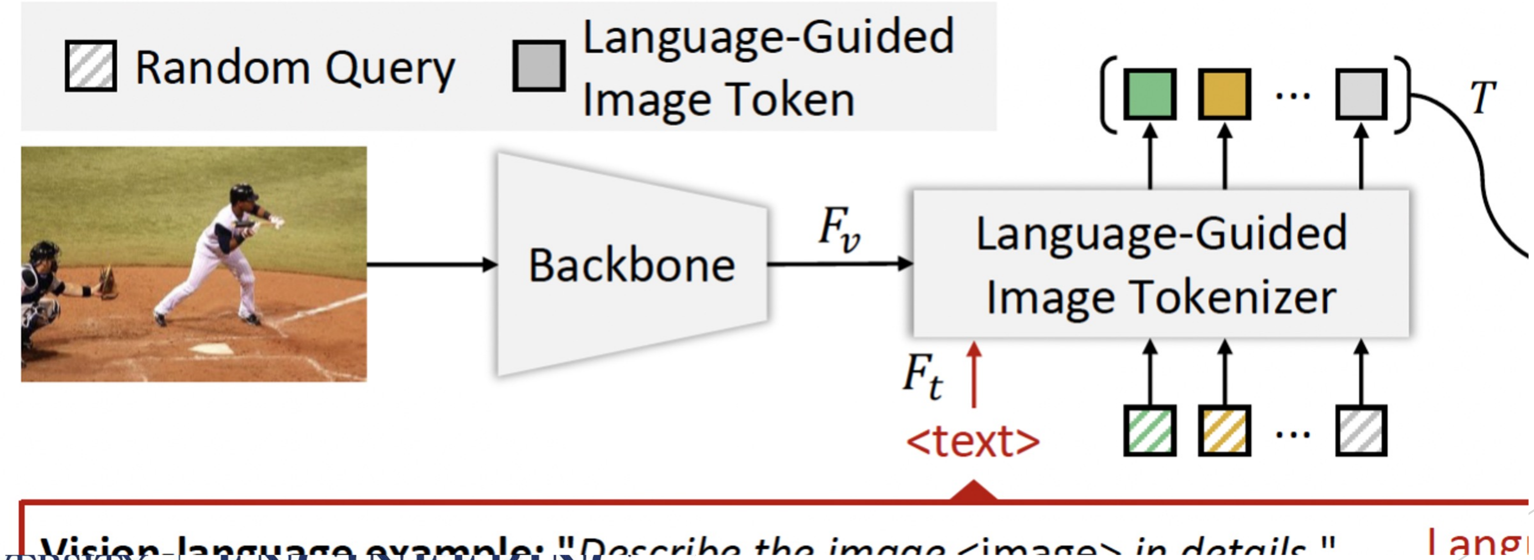
**Vision-only example:** "*For each object in image* <image> *that is a member of class set* <class>*, output a tuple with the class label and the coordinates of a polygon with 16 points that encloses the object. The coordinates should be within range* <range>*. The output format should be (c, x1, y1,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

# Unified Visual Instruction

- Vision-language Tasks:
  - E.g. Image captioning:
    - “The image is <image>. Please generate a caption for the image: ”
  - These instructions are straightforward since they are similar to NLP tasks.
- Vision-only Tasks:
  - E.g. object segmentation
  - Challenge to create instructions for these tasks due to difference in modality between vision and language.
  - LLM used to create set of instructions with various task descriptions (randomly selected at training)
  - Specify output to have a class index from set of categories and a tuple showing where in the segment is.
  - “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>”

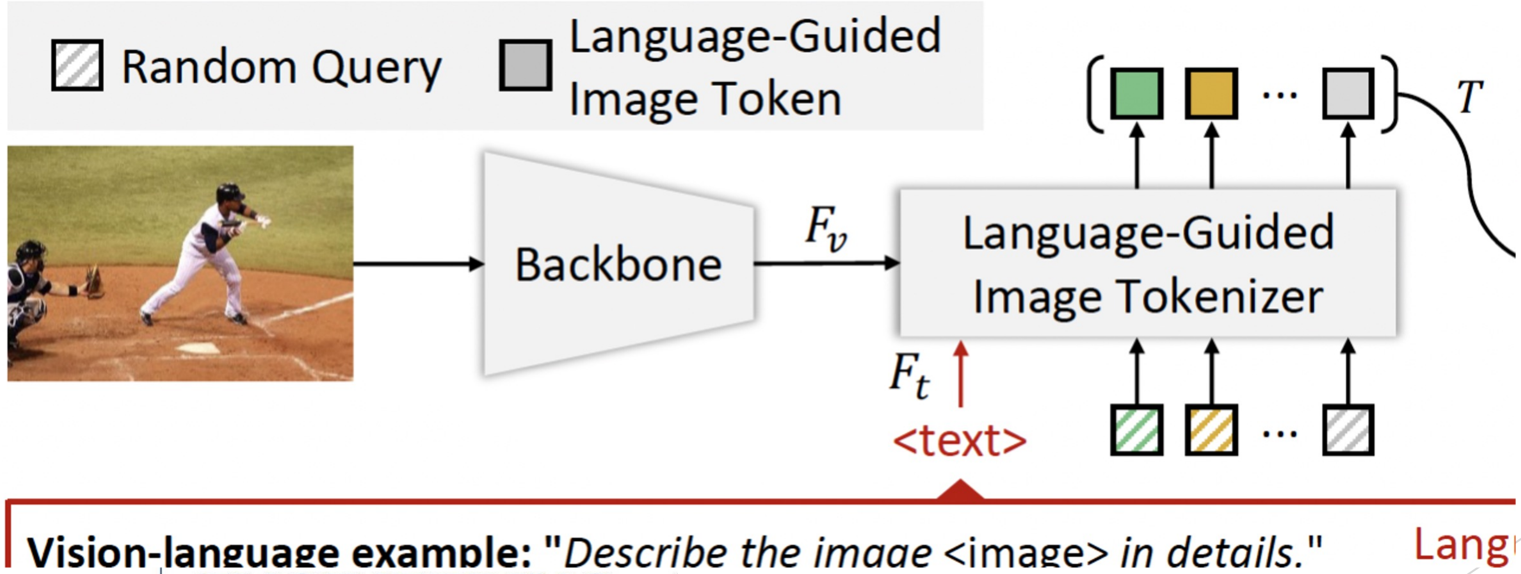
# Language-Guided Image Tokenizer

- Instead of fixed-size batch embeddings, VisionLLM considers images as a foreign language and converts them into a token representation.
- This design design allows tokenizer to “flexibly encode visual information that aligns with task-specific language prompts or instructions.”


▨ Random Query
□ Language-Guided Image Token
Backbone
$F_v$
Language-Guided Image Tokenizer
$T$
$F_t$
<text>
Vision-language example: "Describe the image <image> in details."
Lang

# Language-Guided Image Tokenizer

- Image features from model like ResNet and Language features from model like BERT
- Transformer like Deformable DETR with M randomly initialized Queries produces M tokens, each represented by an embedding and location.


▨ Random Query
■ Language-Guided Image Token
Backbone
$F_v$
Language-Guided Image Tokenizer
$F_t$
<text>
$T$
Vision-language example: "Describe the image <image> in details."
Lang

$$T = \{(e_i, l_i)\}_{i=1}^{M},$$

# LLM-based Open-Ended Task Decoder

- Built on top of Alpaca (LLaMa based LLM adapted to handle some vision tasks).
- Drawbacks:
  - Only has few digits numbers in vocabulary (e.g. 0-9), this makes it hard for model locate objects by numbers.
  - Uses multiple tokens to represent category name; this causes some inefficiencies.
  - Since the model is causal, it is inefficient for visual perception tasks.

# LLM-based Open-Ended Task Decoder

- Mitigation 1:
  - Introduce a set of location tokens:
    - {<p-512>, ..., <p0>,..., <p512>}
    - <p i> ,where i ∈ [−512, 512], is the offset to the location l_i of the image token
    - Relative value to image height or width is equal to i/512
  - These tokens change object localization from a continuous variable prediction task into a discrete bin classification task.

# LLM-based Open-Ended Task Decoder

- Mitigation 2:
  - Introduce set of semantic-agnostic classification tokens.
    - {<c0>, <c1>, ..., <c511>}
    - This replaces category names, which are inefficient since they originally could take more than one token.
    - Category names to tokens are mapped. E.g.: {"person":<c0>, "car":<c1>, "black cat":<c2>,...}

# LLM-based Open-Ended Task Decoder

- Mitigation 3:
  - Output-format-as-query decoding
  - Parse structural tokens and input as query to the decoder.
  - This avoids inefficient token-by-token decoding for vision perception tasks, while keeping unified framework for vision-language tasks.
  - Outputs of object location are treated as foreign language

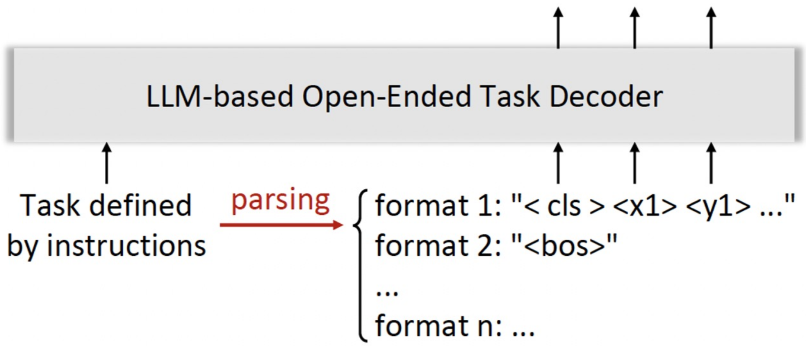

LLM-based Open-Ended Task Decoder
Task defined by instructions
parsing
format 1: "< cls > <x1> <y1> ..."
format 2: "<bos>"
...
format n: ...

Figure 4: Illustration of the "output-format-as-query" decoding process. "<cls> <x1> <y1> ..." denote the queries of the object's class index and boundary points, and "<bos>" denotes the beginning of string.

# Training Details

- Cross-entropy loss objective
- Low-Rank Adaptation (LoRA) is used in training the models.
    - This makes training more efficient and helps bridge gap between modalities.
- Datasets:
    - COCO2017: Used for training and evaluation in object detection and instance segmentation tasks.
    - RefCOCO, RefCOCO+, and RefCOCOg: These datasets are combined for training in visual grounding tasks. The models are evaluated on the validation set of RefCOCO.
    - COCO Caption: Used as the training source for image captioning tasks.
    - LLaVA-Instruct-150K: Employed for training in visual question answering tasks.
- 50 epochs
- M = 100 queries

# Experiments: Benchmarks

- Variety of task:
    - Object Detection: Identifying and localizing objects within an image.
    - Instance Segmentation: Identifying and segmenting individual objects within an image.
    - Visual Grounding: Associating textual descriptions with corresponding regions or objects within an image.
    - Image Captioning: Generating descriptive text for an image.
    - Visual Question Answering (VQA): Answering questions based on the content of an image.

# Experiments: Vision-centric task results

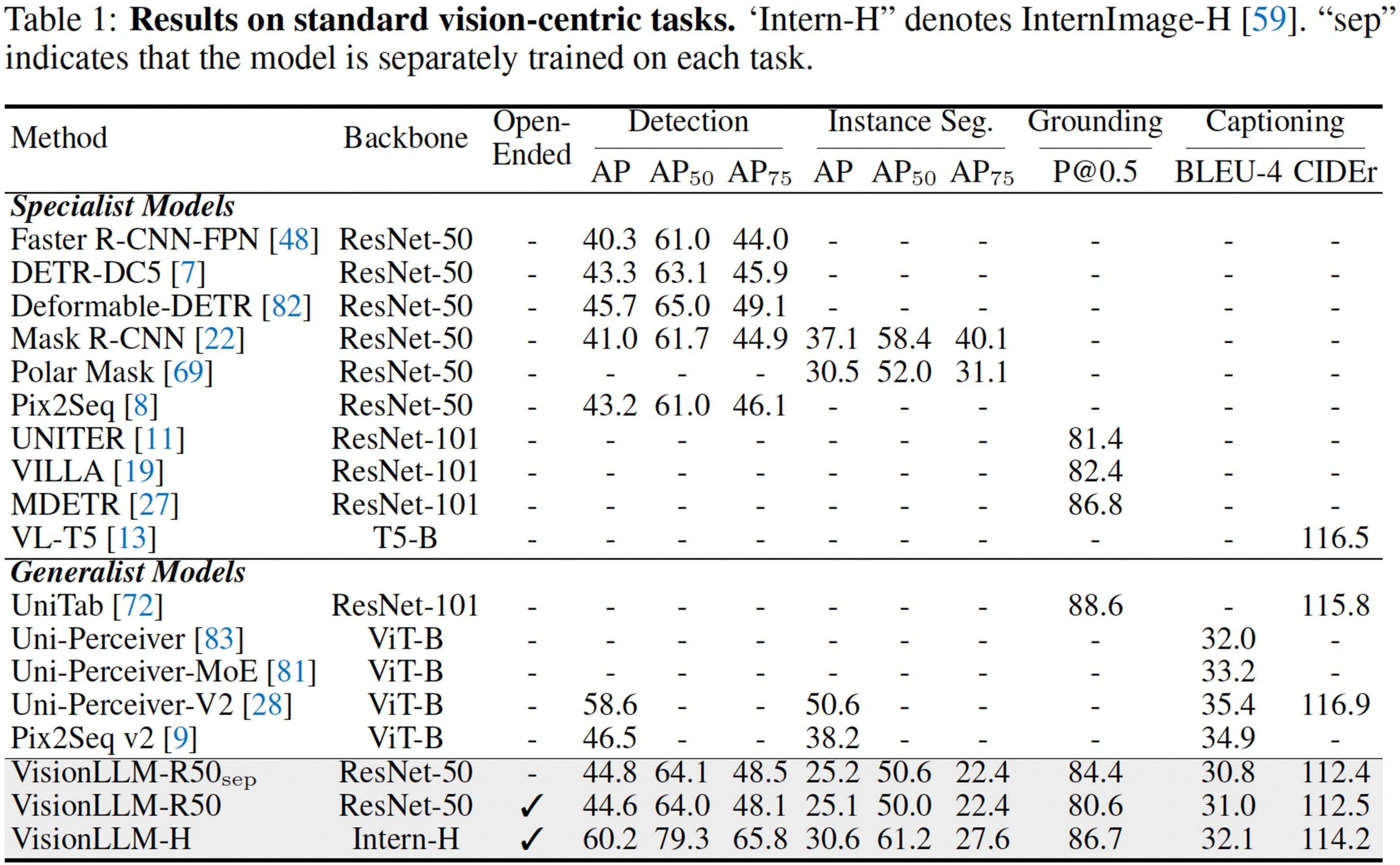


Table 1: **Results on standard vision-centric tasks.** ‘Intern-H” denotes InternImage-H [59]. “sep” indicates that the model is separately trained on each task.

| 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: Vision-centric task results

- Competitive results in:
  - Object detection
  - Visual grounding
  - Image Captioning
- Not as well result Instance Segmentation:
  - AP_50 (61.2% with InternImage-H [59]) but relatively low mask AP_75 (27.6%).

# Experiments: object-level and output format customization

- Authors alter the <class> tag within the language instructions to change the model's recognition targets 10 classes to 80 classes.
- Author also alter number of boundary points in output format.
- The results demonstrate VisionLLM's ability to customize the target object and output format.

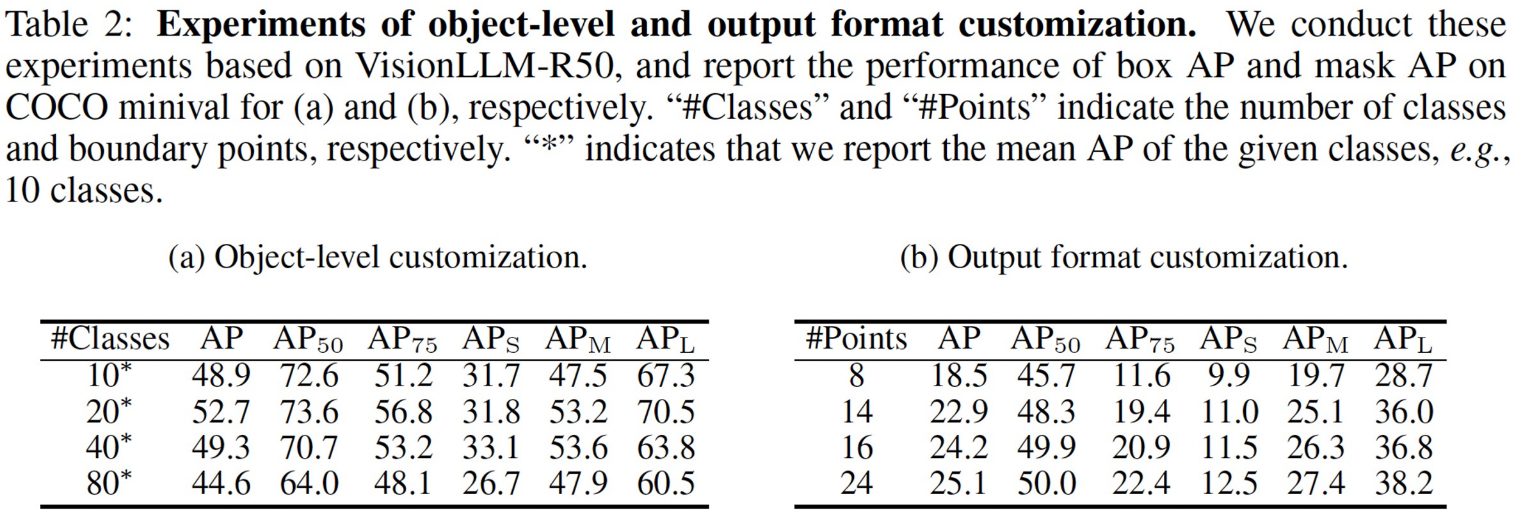


Table 2: **Experiments of object-level and output format customization.** We conduct these experiments based on VisionLLM-R50, and report the performance of box AP and mask AP on COCO minival for (a) and (b), respectively. "#Classes" and "#Points" indicate the number of classes and boundary points, respectively.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

(a) Object-level customization.

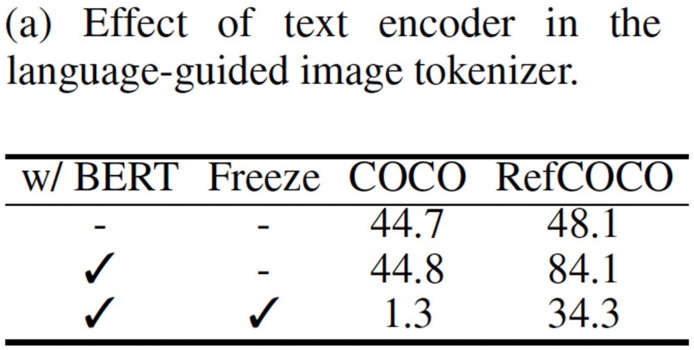
| #Classes | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|
| 10* | 48.9 | 72.6 | 51.2 | 31.7 | 47.5 | 67.3 |
| 20* | 52.7 | 73.6 | 56.8 | 31.8 | 53.2 | 70.5 |
| 40* | 49.3 | 70.7 | 53.2 | 33.1 | 53.6 | 63.8 |
| 80* | 44.6 | 64.0 | 48.1 | 26.7 | 47.9 | 60.5 |

(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 |

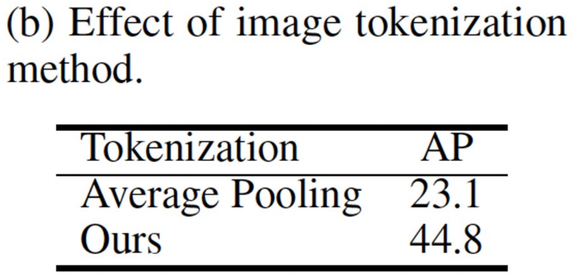
# Ablation Study

- Single Task vs. Multiple Tasks
  - VisionLLM trained on a single task only works slightly better than its multi-task counterpart for all tasks except image captioning.
- Text Encoder in Language-Guided Image Tokenizer
  - Examining role of text encoder (BERT) in language-guided image tokenizer.
  - “BERT is not essential for object detection but it is crucial for visual grounding”
  - Freezing BERT model hinders alignment of text and vision modalities.
- Image Tokenization Method
  - Image tokenization method works superior to employing average
  - pooling on the feature maps from the D-DETR encoder to obtain M patch embeddings
- Number of Localization Tokens
  - The increase of localization tokens improves performances.



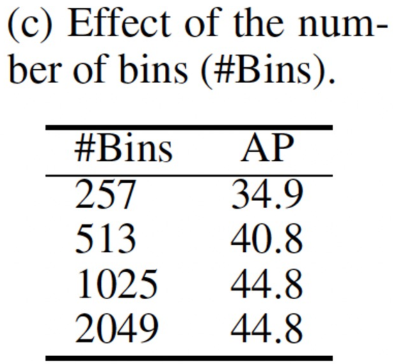
(a) Effect of text encoder in the language-guided image tokenizer.

| w/ BERT | Freeze | COCO | RefCOCO |
|---|---|---|---|
| - | - | 44.7 | 48.1 |
| ✓ | - | 44.8 | 84.1 |
| ✓ | ✓ | 1.3 | 34.3 |



(b) Effect of image tokenization method.

| Tokenization | AP |
|---|---|
| Average Pooling | 23.1 |
| Ours | 44.8 |



(c) Effect of the number of bins (#Bins).

| #Bins | AP |
|---|---|
| 257 | 34.9 |
| 513 | 40.8 |
| 1025 | 44.8 |
| 2049 | 44.8 |

# Limitations

- VisionLLM’s performance is bottlenecked by the performance of open-source LLMs
  - LLM must be trained, so proprietary models cannot be used.
- Lacks in performance on instance segmentation
- Author’s don’t explore in-context learning or few-shot capabilities.

# Conclusion

- The paper presents VisionLLM, a novel framework that aligns vision-centric tasks with language models' methodologies.
- VisionLLM allows for seamless integration and handling of diverse vision-centric tasks like object detection, instance segmentation, and image captioning through language instructions.

# Visual Instruction Tuning

Haotian Liu, Chunyuan Li, Qingyang Wu, Yong Jae Lee

University of Virginia | ENGINEERING
Department of Computer Science

# Introduction

- Develop a **general-purpose assistant** that **effectively follows multi-modal vision-and-language instructions**
  → Large Language Models + visual instruction-following

- Contributions
  - Generate multimodal instruction-following **data** using GPT-4
  - Large multimodal models (**LLaVA**: Large Language and Vision Assistant)
  - Multimodal instruction-following **benchmark**
  - **Open-source**

# Background

- **End-to-end trained models**
  - Vision-Language Navigation task: navigate real or virtual environments based on textual instruction
  - InstructPix2Pix: image editing based on textual instruction

**→ Specialized Single-models, Domain-specific**

- **Systems coordinating multiple models**
  - Visual ChatGPT, MM-REACT, ViperGPT
  - Combine different models to enhance instruction-following capabilities

**→ Integrated systems**

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

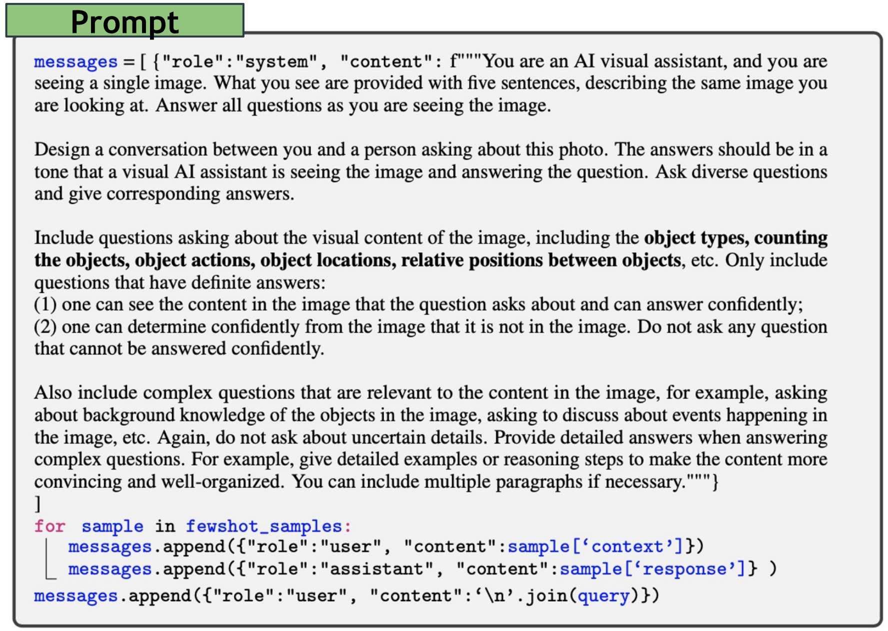
# Background

- **Instruction Tuning**
  - Effective in improving LLM performances to **align closely with human instructions**
  - Success in NLP: achieved **better generalization** capabilities in **zero and few-shot** settings
  - Potential for Computer Vision
    - currently less explored in multimodal tasks for visual & text

# GPT-assisted Visual Instruction Data Generation

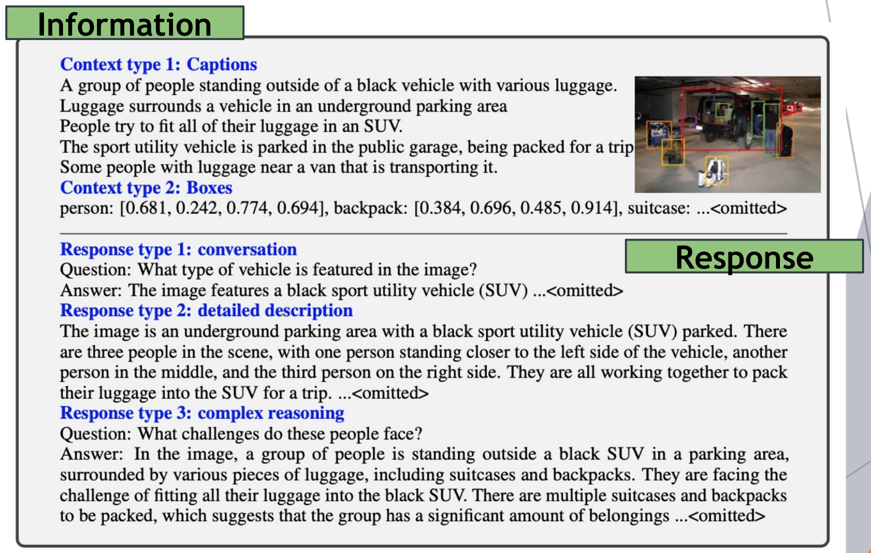
- Data generation process
  - GPT-4: transform image-text pairs into multimodal instruction-following data

Prompt



```
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)})
```

Information



**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

**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>

Table 1: One example to illustrate the instruction-following data. The top block shows the contexts such as captions and boxes used to prompt GPT, and the bottom block shows the three types of responses. Note that the visual image is not used to prompt GPT, we only show it here as a reference.

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

# GPT-assisted Visual Instruction Data Generation

- Data generation process
  - GPT-4:  transform image-text pairs into multimodal instruction-following data
  - Provide captions & bounding boxes information

- Types of data generated
  - Conversation
  - Detailed description
  - Complex reasoning

- Outcome
  - 158,000 unique language-image instruction-following samples
    - 58,000 conversations
    - 23,000 detailed descriptions
    - 77,000 complex reasoning

# Visual Instruction Tuning: Architecture

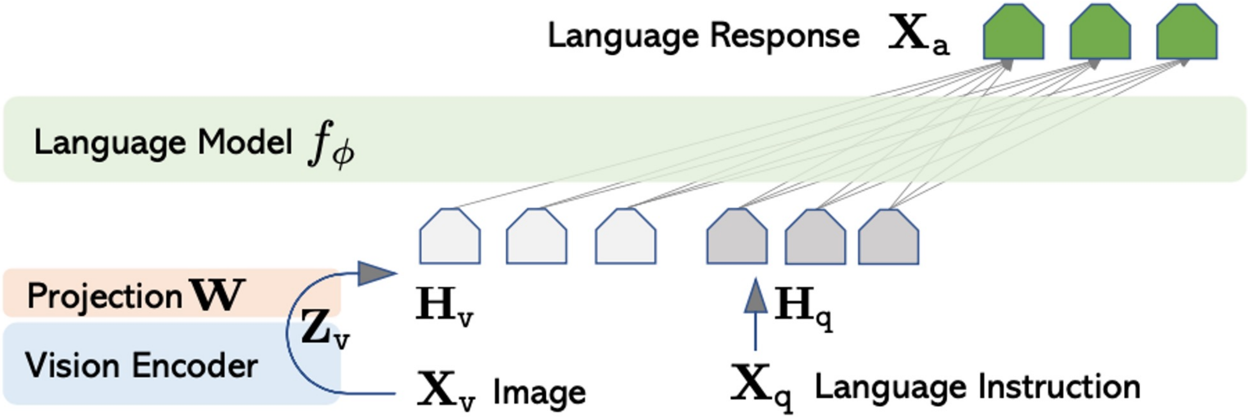
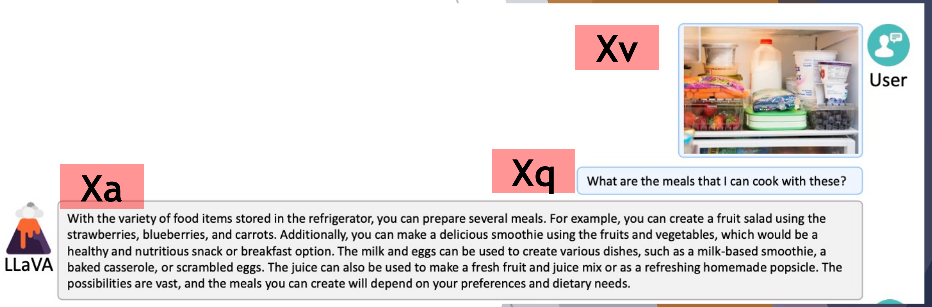


Figure 1: LLaVA network architecture.



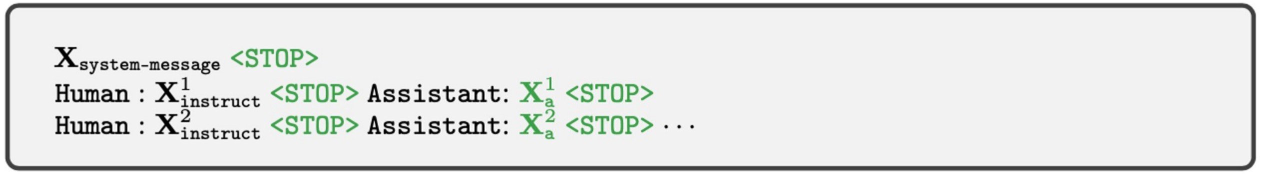
What are the meals that I can cook with these?

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.

- Language Model: Vicuna/LLaMA
- Visual Encoder: CLIP
  - Understand images in natural language descriptions
- Projection Layer: connects output of visual encoder, translate features into what LM can process, input to LM

# Visual Instruction Tuning: Training

- Input



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

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

- Multi-turn conversion data $(\mathbf{X}^1_{\texttt{q}}, \mathbf{X}^1_{\texttt{a}}, \cdots, \mathbf{X}^T_{\texttt{q}}, \mathbf{X}^T_{\texttt{a}})$

- Probability of generating correct sequence of words for the answers

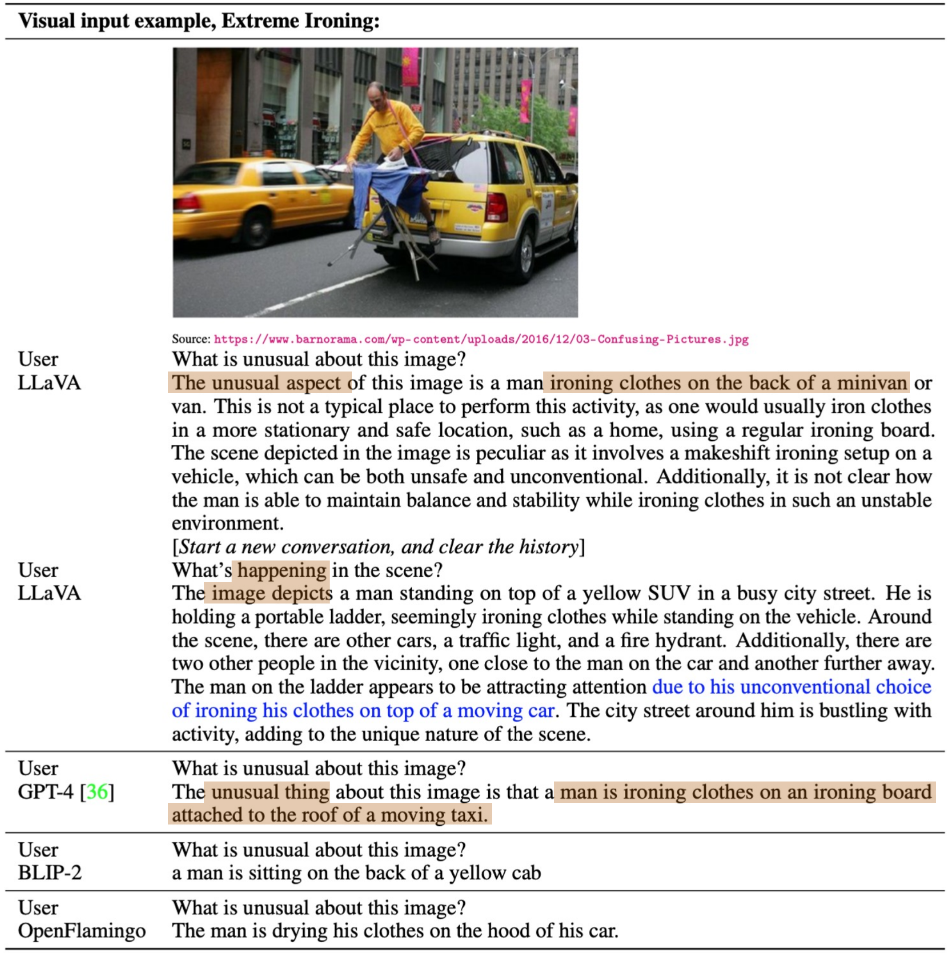
$$p(\mathbf{X}_{\texttt{a}} | \mathbf{X}_{\texttt{v}}, \mathbf{X}_{\texttt{instruct}}) = \prod_{i=1}^{L} p_{\boldsymbol{\theta}}(\boldsymbol{x}_i | \mathbf{X}_{\texttt{v}}, \mathbf{X}_{\texttt{instruct},<i}, \mathbf{X}_{\texttt{a},<i}),$$

# Visual Instruction Tuning: Training

- **Stage 1: Pre-training for Feature Alignment**
  - Align visual features with LM using image-text pairs
  - Maximize likelihood of generating appropriate **text captions** from images
    → **Train projection layer**, Weights of visual encoder and LM kept frozen

- **Stage 2: Fine-tuning End-to-End**
  - Fine **tuning projection layer and LM**, using generated dataset (no tuning on visual encoder)
  - Fine tuning for specific applications
    - Multimodal chatbot
    - Science QA dataset
      → **adaptability on different tasks**

# Experiment: Multimodal Chatbot



**Visual input example, Extreme Ironing:**

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

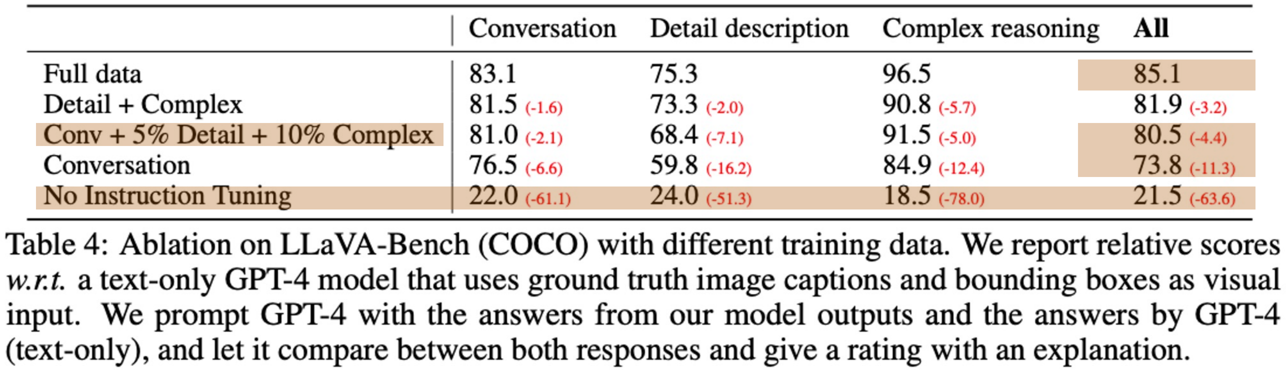
| | |
|---|---|
| User | What is unusual about this image? |
| LLaVA | The unusual aspect of this image is a man ironing clothes on the back of a minivan or van. This is not a typical place to perform this activity, as one would usually iron clothes in a more stationary and safe location, such as a home, 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 | What's happening in the scene? |
| LLaVA | 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. |
| User | What is unusual about this image? |
| GPT-4 [36] | The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi. |
| User | What is unusual about this image? |
| BLIP-2 | a man is sitting on the back of a yellow cab |
| User | What is unusual about this image? |
| OpenFlamingo | The man is drying his clothes on the hood of his car. |

- Examples required **in-depth image understanding**

- Results
  : LLaVA reasoning similar to multi-modal GPT-4, better than BLIP-2 and OpenFlamingo
  - **Addresses prompts** about images
  - Provide **meaningful responses** including **reasoning** (beyond basic description)

# Experiment: LLaVA-Bench

- COCO
  - Dataset: 30 images from COCO-Val-2014 with 90 questions



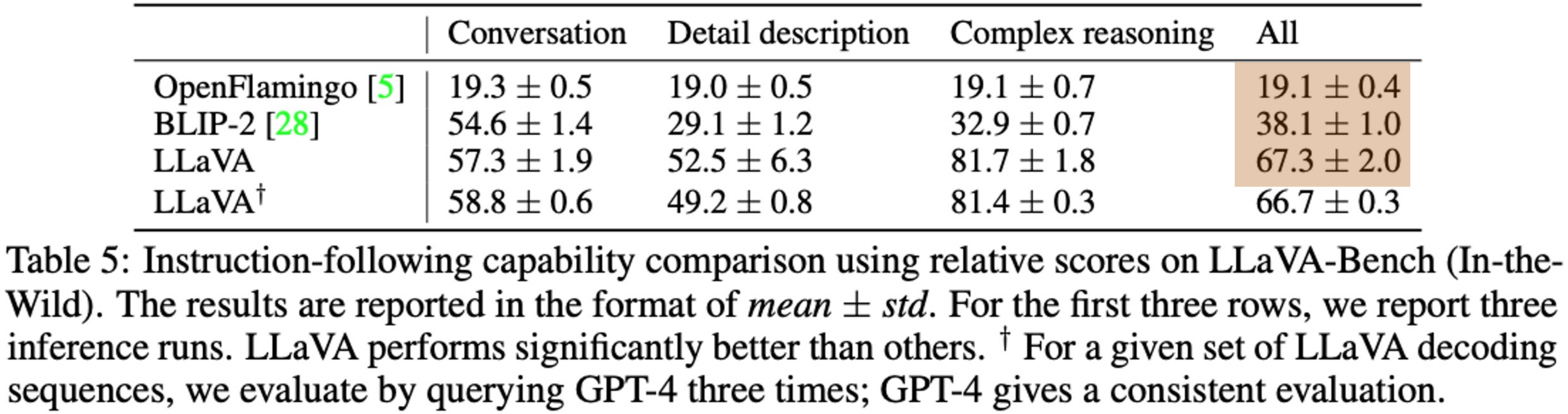
| | Conversation | Detail description | Complex reasoning | **All** |
|---|---|---|---|---|
| Full data | 83.1 | 75.3 | 96.5 | 85.1 |
| Detail + Complex | 81.5 (-1.6) | 73.3 (-2.0) | 90.8 (-5.7) | 81.9 (-3.2) |
| Conv + 5% Detail + 10% Complex | 81.0 (-2.1) | 68.4 (-7.1) | 91.5 (-5.0) | 80.5 (-4.4) |
| Conversation | 76.5 (-6.6) | 59.8 (-16.2) | 84.9 (-12.4) | 73.8 (-11.3) |
| No Instruction Tuning | 22.0 (-61.1) | 24.0 (-51.3) | 18.5 (-78.0) | 21.5 (-63.6) |

Table 4: Ablation on LLaVA-Bench (COCO) with different training data. We report relative scores *w.r.t.* a text-only GPT-4 model that uses ground truth image captions and bounding boxes as visual input. We prompt GPT-4 with the answers from our model outputs and the answers by GPT-4 (text-only), and let it compare between both responses and give a rating with an explanation.

  - Instruction tuning → 50% improvement
  - Mixed data types → 7% improvement
  - Full data, best results

# Experiment: LLaVA-Bench

- In-the-Wild
  - Indoor and outdoor scenes, memes, paintings, sketches, etc. with 60 sets of descriptions and questions
  - Test the generalizability and performance on diverse tasks



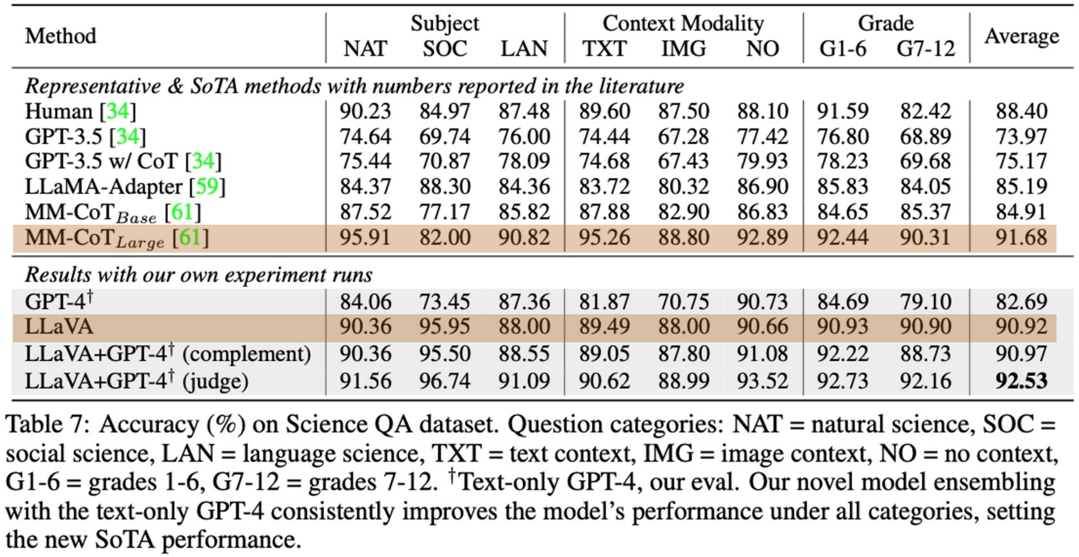
| | Conversation | Detail description | Complex reasoning | All |
|---|---|---|---|---|
| OpenFlamingo [5] | 19.3 ± 0.5 | 19.0 ± 0.5 | 19.1 ± 0.7 | 19.1 ± 0.4 |
| BLIP-2 [28] | 54.6 ± 1.4 | 29.1 ± 1.2 | 32.9 ± 0.7 | 38.1 ± 1.0 |
| LLaVA | 57.3 ± 1.9 | 52.5 ± 6.3 | 81.7 ± 1.8 | 67.3 ± 2.0 |
| LLaVA† | 58.8 ± 0.6 | 49.2 ± 0.8 | 81.4 ± 0.3 | 66.7 ± 0.3 |

Table 5: Instruction-following capability comparison using relative scores on LLaVA-Bench (In-the-Wild). The results are reported in the format of *mean* ± *std*. For the first three rows, we report three inference runs. LLaVA performs significantly better than others. † For a given set of LLaVA decoding sequences, we evaluate by querying GPT-4 three times; GPT-4 gives a consistent evaluation.

UNIVERSITY of VIRGINIA | ENGINEERING Department of Computer Science

# Experiment: ScienceQA

- Dataset: ScienceQA
  - 21k multiple-choice questions on scientific domains and skills
  - natural science, social science, language science, etc.



| Method | Subject |  |  | Context Modality |  |  | Grade |  | Average |
|---|---|---|---|---|---|---|---|---|---|
|  | NAT | SOC | LAN | TXT | IMG | NO | G1-6 | G7-12 |  |
| *Representative & SoTA methods with numbers reported in the literature* |  |  |  |  |  |  |  |  |  |
| Human [34] | 90.23 | 84.97 | 87.48 | 89.60 | 87.50 | 88.10 | 91.59 | 82.42 | 88.40 |
| GPT-3.5 [34] | 74.64 | 69.74 | 76.00 | 74.44 | 67.28 | 77.42 | 76.80 | 68.89 | 73.97 |
| GPT-3.5 w/ CoT [34] | 75.44 | 70.87 | 78.09 | 74.68 | 67.43 | 79.93 | 78.23 | 69.68 | 75.17 |
| LLaMA-Adapter [59] | 84.37 | 88.30 | 84.36 | 83.72 | 80.32 | 86.90 | 85.83 | 84.05 | 85.19 |
| MM-CoT$_{Base}$ [61] | 87.52 | 77.17 | 85.82 | 87.88 | 82.90 | 86.83 | 84.65 | 85.37 | 84.91 |
| MM-CoT$_{Large}$ [61] | 95.91 | 82.00 | 90.82 | 95.26 | 88.80 | 92.89 | 92.44 | 90.31 | 91.68 |
| *Results with our own experiment runs* |  |  |  |  |  |  |  |  |  |
| GPT-4$^{\dagger}$ | 84.06 | 73.45 | 87.36 | 81.87 | 70.75 | 90.73 | 84.69 | 79.10 | 82.69 |
| LLaVA | 90.36 | 95.95 | 88.00 | 89.49 | 88.00 | 90.66 | 90.93 | 90.90 | 90.92 |
| LLaVA+GPT-4$^{\dagger}$ (complement) | 90.36 | 95.50 | 88.55 | 89.05 | 87.80 | 91.08 | 92.22 | 88.73 | 90.97 |
| LLaVA+GPT-4$^{\dagger}$ (judge) | 91.56 | 96.74 | 91.09 | 90.62 | 88.99 | 93.52 | 92.73 | 92.16 | **92.53** |

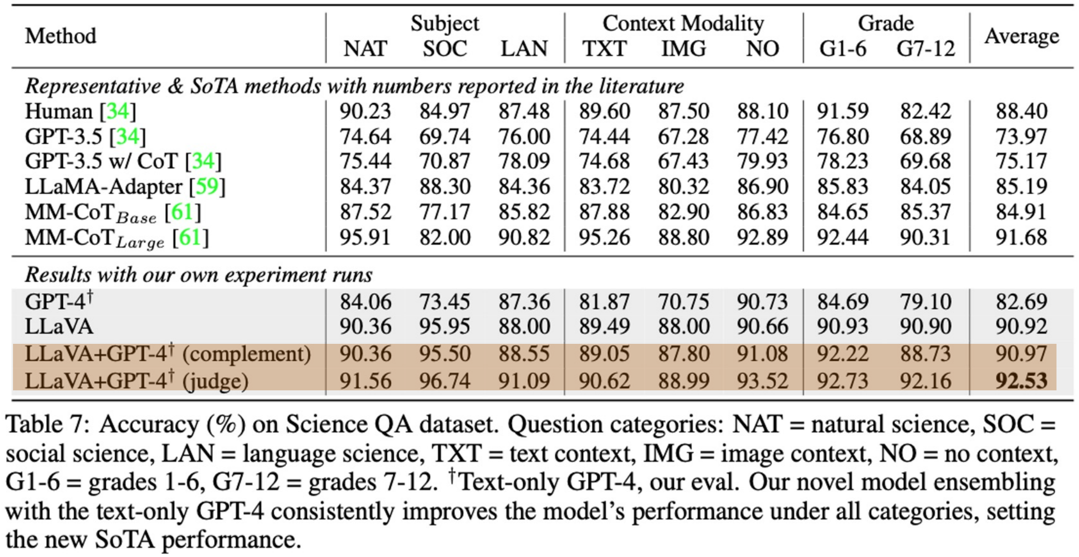
Table 7: Accuracy (%) on Science QA dataset. Question categories: NAT = natural science, SOC = social science, LAN = language science, TXT = text context, IMG = image context, NO = no context, G1-6 = grades 1-6, G7-12 = grades 7-12. $^{\dagger}$Text-only GPT-4, our eval. Our novel model ensembling with the text-only GPT-4 consistently improves the model's performance under all categories, setting the new SoTA performance.

UNIVERSITY of VIRGINIA

ENGINEERING
Department of Computer Science

# Experiment: ScienceQA

- Dataset: ScienceQA
  - 21k multiple-choice questions on scientific domains and skills
  - natural science, social science, language science, etc.



| Method | Subject | | | Context Modality | | | Grade | | Average |
|---|---|---|---|---|---|---|---|---|---|
| | NAT | SOC | LAN | TXT | IMG | NO | G1-6 | G7-12 | |
| *Representative & SoTA methods with numbers reported in the literature* | | | | | | | | | |
| Human [34] | 90.23 | 84.97 | 87.48 | 89.60 | 87.50 | 88.10 | 91.59 | 82.42 | 88.40 |
| GPT-3.5 [34] | 74.64 | 69.74 | 76.00 | 74.44 | 67.28 | 77.42 | 76.80 | 68.89 | 73.97 |
| GPT-3.5 w/ CoT [34] | 75.44 | 70.87 | 78.09 | 74.68 | 67.43 | 79.93 | 78.23 | 69.68 | 75.17 |
| LLaMA-Adapter [59] | 84.37 | 88.30 | 84.36 | 83.72 | 80.32 | 86.90 | 85.83 | 84.05 | 85.19 |
| MM-CoT$_{Base}$ [61] | 87.52 | 77.17 | 85.82 | 87.88 | 82.90 | 86.83 | 84.65 | 85.37 | 84.91 |
| MM-CoT$_{Large}$ [61] | 95.91 | 82.00 | 90.82 | 95.26 | 88.80 | 92.89 | 92.44 | 90.31 | 91.68 |
| *Results with our own experiment runs* | | | | | | | | | |
| GPT-4$^\dagger$ | 84.06 | 73.45 | 87.36 | 81.87 | 70.75 | 90.73 | 84.69 | 79.10 | 82.69 |
| LLaVA | 90.36 | 95.95 | 88.00 | 89.49 | 88.00 | 90.66 | 90.93 | 90.90 | 90.92 |
| LLaVA+GPT-4$^\dagger$ (complement) | 90.36 | 95.50 | 88.55 | 89.05 | 87.80 | 91.08 | 92.22 | 88.73 | 90.97 |
| LLaVA+GPT-4$^\dagger$ (judge) | 91.56 | 96.74 | 91.09 | 90.62 | 88.99 | 93.52 | 92.73 | 92.16 | **92.53** |

Table 7: Accuracy (%) on Science QA dataset. Question categories: NAT = natural science, SOC = social science, LAN = language science, TXT = text context, IMG = image context, NO = no context, G1-6 = grades 1-6, G7-12 = grades 7-12. $^\dagger$Text-only GPT-4, our eval. Our novel model ensembling with the text-only GPT-4 consistently improves the model's performance under all categories, setting the new SoTA performance.

# Conclusion

- Contribution
  - Multimodal instruction-following data
  - LLaVA & LLaVA-Bench

- Future Works
  - Expand model’s knowledge base and multilingual capabilities
  - Enhance high-resolution image processing and semantic understanding for better visual comprehension
  - Methods for integrating external data sources

# NExT-GPT: Any-to-Any Multimodal LLM

Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, Tat-Seng Chua

University of Virginia | ENGINEERING
Department of Computer Science

# Introduction

- Background & Motivation
  - Rapid advancement in AI-generated content (AIGC)
  - Importance of multimodality **as humans perceive and communicate**
- Limitations of Current Multimodal LLMs (MM-LLMs)
  - Multi-modal understanding 👍
  - **Producing multi-modal contents** 👎

- **NExT-GPT**
  - **End-to-end, general-purpose any-to-any MM-LLM system**
  - Understanding and generation on **text, images, videos, and audio**
  - Use existing encoders and decoders, tune small fraction for low-cost training
  - Introduce **Modality-switching Instruction Tuning** (MosIT) for complex semantic understanding & generation

University of Virginia | ENGINEERING Department of Computer Science

# Related Work

- Cross-modal understanding and generation
  - Image/Video captioning (COCO dataset challenges)
  - Text to Image/Video/Synthesis (DALL-E, RAVE)

    → **Difficulties to create unified models for varied modalities**

- Multimodal Large Language Models
  - Integration with modal encoders with text based LLMs (Flamingo, LLaVa)

    → **primarily focus on multimodal input comprehension, not multimodal generation**

# Overall Architecture: NExT-GPT

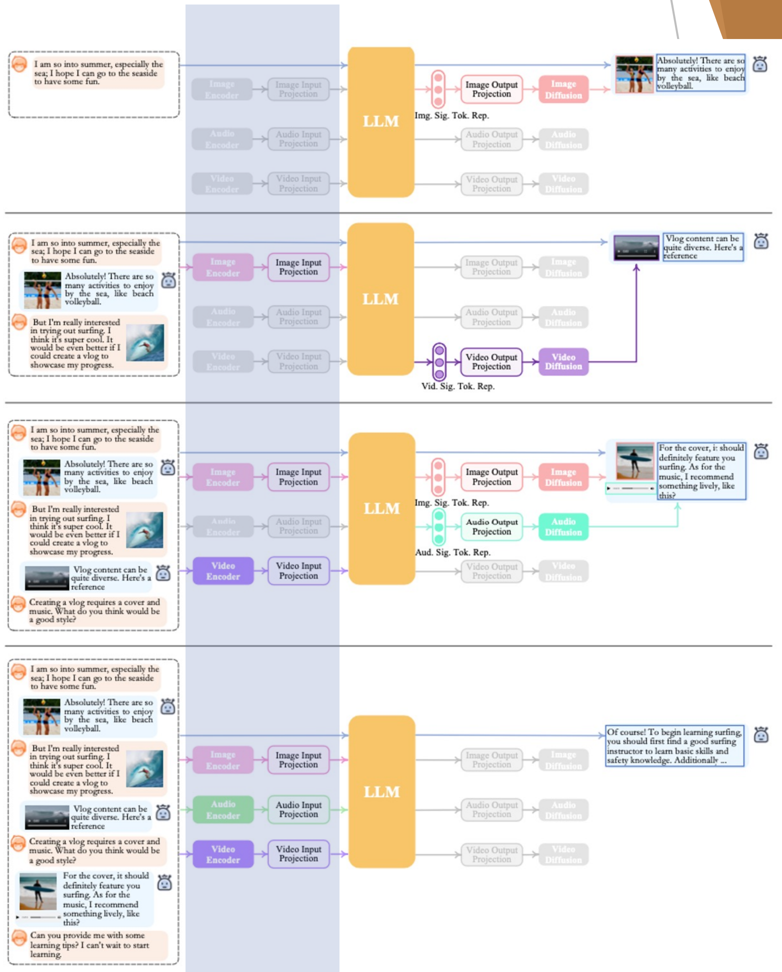
**Multimodal encoding**
: used existing models for encoding various inputs (ImageBind)

**LLM understanding and reasoning**
: use **Vicuna** to process encoded multimodal inputs

semantic understanding
reasoning over inputs
deciding modality of output
generate textual signal tokens (instructions)

**Multimodal generation**
: transformer-based output projection layer

translate LLM's instruction into different diffusion models
generate final content



I am so into summer, especially the sea; I hope I can go to the seaside to have some fun.

Absolutely! There are so many activities to enjoy by the sea, like beach volleyball.

I am so into summer, especially the sea; I hope I can go to the seaside to have some fun.

Absolutely! There are so many activities to enjoy by the sea, like beach volleyball.

But I'm really interested in trying out surfing. I think it's super cool. It would be even better if I could create a vlog to showcase my progress.

Vlog content can be quite diverse. Here's a reference

I am so into summer, especially the sea; I hope I can go to the seaside to have some fun.

Absolutely! There are so many activities to enjoy by the sea, like beach volleyball.

But I'm really interested in trying out surfing. I think it's super cool. It would be even better if I could create a vlog to showcase my progress.

Vlog content can be quite diverse. Here's a reference

Creating a vlog requires a cover and music. What do you think would be a good style?

For the cover, it should definitely feature you surfing. As for the music, I recommend something lively, like this?

I am so into summer, especially the sea; I hope I can go to the seaside to have some fun.

Absolutely! There are so many activities to enjoy by the sea, like beach volleyball.

But I'm really interested in trying out surfing. I think it's super cool. It would be even better if I could create a vlog to showcase my progress.

Vlog content can be quite diverse. Here's a reference

Creating a vlog requires a cover and music. What do you think would be a good style?

For the cover, it should definitely feature you surfing. As for the music, I recommend something lively, like this?

Can you provide me with some learning tips? I can't wait to start learning.

Of course! To begin learning surfing, you should first find a good surfing instructor to learn basic skills and safety knowledge. Additionally ...

Figure 2: NExT-GPT inference process. Grey colors denote the deactivation of the modules.

UNIVERSITY of VIRGINIA

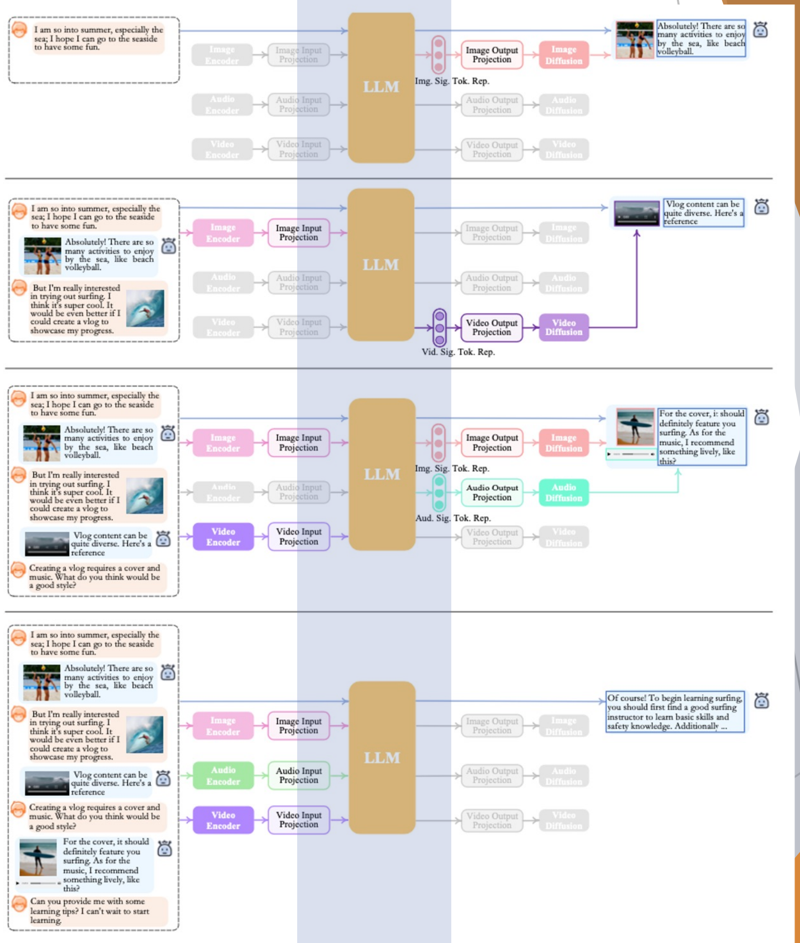
ENGINEERING
Department of Computer Science

# Overall Architecture: NExT-GPT

**Multimodal encoding**
: used existing models for encoding various inputs (ImageBind)

**LLM understanding and reasoning**
: use **Vicuna** to process encoded multimodal inputs
- semantic understanding
- reasoning over inputs
- deciding modality of output
- generate textual signal tokens (instructions)

**Multimodal generation**
: transformer-based output projection layer
- translate LLM's instruction into different diffusion models
- generate final content



I am so into summer, especially the sea; I hope I can go to the seaside to have some fun.

Absolutely! There are so many activities to enjoy by the sea, like beach volleyball.

I am so into summer, especially the sea; I hope I can go to the seaside to have some fun.

Absolutely! There are so many activities to enjoy by the sea, like beach volleyball.

But I'm really interested in trying out surfing. I think it's super cool. It would be even better if I could create a vlog to showcase my progress.

Vlog content can be quite diverse. Here's a reference

I am so into summer, especially the sea; I hope I can go to the seaside to have some fun.

Absolutely! There are so many activities to enjoy by the sea, like beach volleyball.

But I'm really interested in trying out surfing. I think it's super cool. It would be even better if I could create a vlog to showcase my progress.

Vlog content can be quite diverse. Here's a reference

Creating a vlog requires a cover and music. What do you think would be a good style?

For the cover, it should definitely feature you surfing. As for the music, I recommend something lively, like this?

I am so into summer, especially the sea; I hope I can go to the seaside to have some fun.

Absolutely! There are so many activities to enjoy by the sea, like beach volleyball.

But I'm really interested in trying out surfing. I think it's super cool. It would be even better if I could create a vlog to showcase my progress.

Vlog content can be quite diverse. Here's a reference

Creating a vlog requires a cover and music. What do you think would be a good style?

For the cover, it should definitely feature you surfing. As for the music, I recommend something lively, like this?

Can you provide me with some learning tips? I can't wait to start learning.

Of course! To begin learning surfing, you should first find a good surfing instructor to learn basic skills and safety knowledge. Additionally ...

Figure 2: NExT-GPT inference process. Grey colors denote the deactivation of the modules.

UNIVERSITY of VIRGINIA | ENGINEERING
Department of Computer Science

# Overall Architecture: NExT-GPT

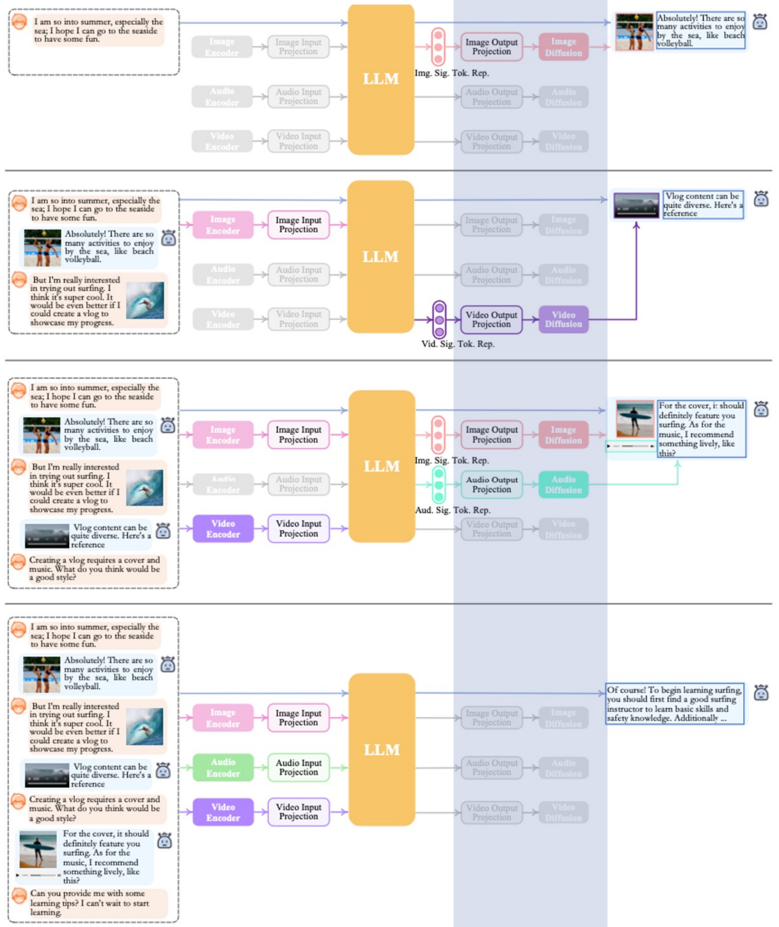
**Multimodal encoding**
: used existing models for encoding various inputs (ImageBind)

**LLM understanding and reasoning**
: use **Vicuna** to process encoded multimodal inputs

semantic understanding
reasoning over inputs
deciding modality of output
generate textual signal tokens (instructions)

**Multimodal generation**
: transformer-based output projection layer

translate LLM's instruction into different diffusion models
generate final content



I am so into summer, especially the sea; I hope I can go to the seaside to have some fun.

Absolutely! There are so many activities to enjoy by the sea, like beach volleyball.

But I'm really interested in trying out surfing. I think it's super cool. It would be even better if I could create a vlog to showcase my progress.

Vlog content can be quite diverse. Here's a reference

Creating a vlog requires a cover and music. What do you think would be a good style?

For the cover, it should definitely feature you surfing. As for the music, I recommend something lively, like this?

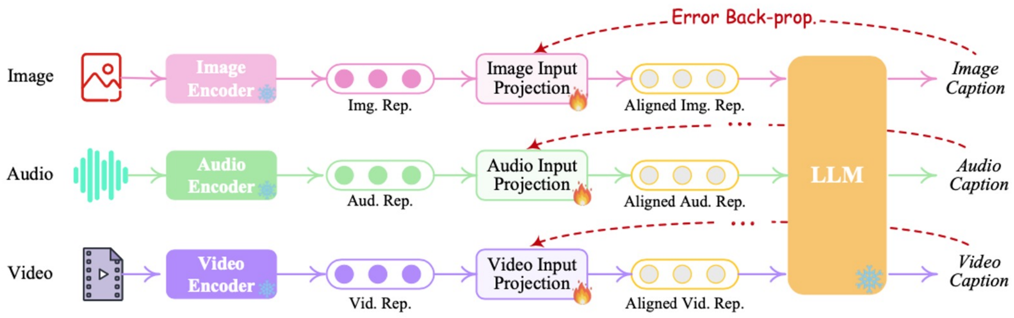
Can you provide me with some learning tips? I can't wait to start learning.

Of course! To begin learning surfing, you should first find a good surfing instructor to learn basic skills and safety knowledge. Additionally ...

Figure 2: NExT-GPT inference process. Grey colors denote the deactivation of the modules.

# Lightweight Multimodal Alignment Learning

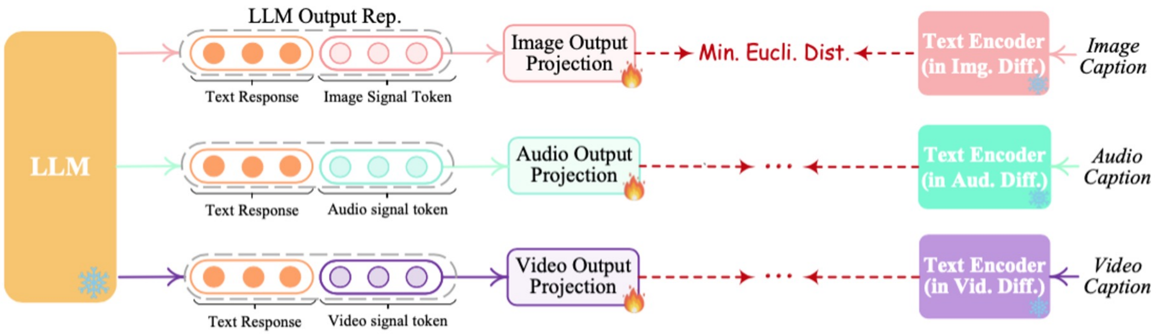
- Encoding Side LLM Centric Multimodal Alignment



(a) Encoding-side LLM-centric Alignment

**encoder** captures important features
→ process into **image representations**
→ **image input projection** transforms them into **aligned image representation** (LM input format)

- Decoding Side Instruction-following Alignment



(b) Decoding-side Instruction-following Alignment

Figure 3: Illustration of the lightweight multimodal alignment learning of encoding and decoding.

**LLM output representation** generate **signal tokens** that are **commands** to making images
→ **image output projection** transforms signal tokens
→ **text encoder** encodes final captions and descriptions for the diffusion model

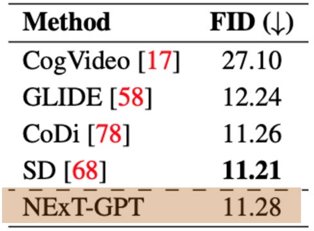
# Instruction Dataset

- **Existing Data (Text+X — Text)**
  - input is a combination of text and different modality (X)
- **Constructed Data (Text — Text+X)**
  - new text to multimodal (T2M) data constructed
  - X-caption pairs with text instructions, further processed with GPT-4
- **MosIT**
  - dataset **specifically designed to train NExT-GPT**
  - Multimodal Interactions: simulate **real-world conversation** scenarios
  - **Multi-turn** Dialogues: comprises **3~7 Q&A pairs** with various modality
  - **Logical** and **Semantic Complexity**: coherent, logically connected, semantically rich, in-depth reasoning data

→ **5,000 dialogues** covering spectrum of instruction-following scenarios on different modalities

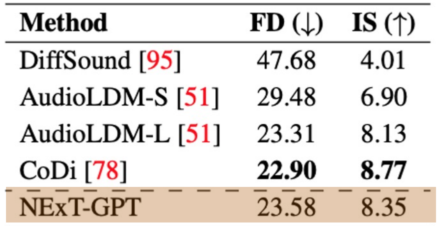
# Experiments:
# Any-to-any Multimodal Generation

- Text-to-X Generation



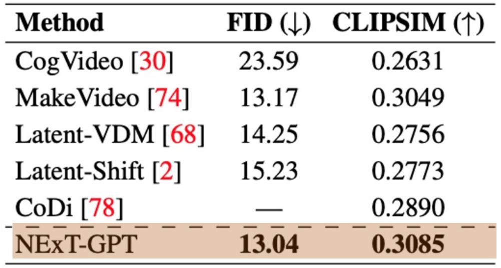
| Method | FID (↓) |
|---|---|
| CogVideo [17] | 27.10 |
| GLIDE [58] | 12.24 |
| CoDi [78] | 11.26 |
| SD [68] | **11.21** |
| NExT-GPT | 11.28 |

Table 3: Text-to-image generation results on COCO-caption data [50].



| Method | FD (↓) | IS (↑) |
|---|---|---|
| DiffSound [95] | 47.68 | 4.01 |
| AudioLDM-S [51] | 29.48 | 6.90 |
| AudioLDM-L [51] | 23.31 | 8.13 |
| CoDi [78] | **22.90** | **8.77** |
| NExT-GPT | 23.58 | 8.35 |

Table 4: Text-to-audio generation results on AudioCaps [38].



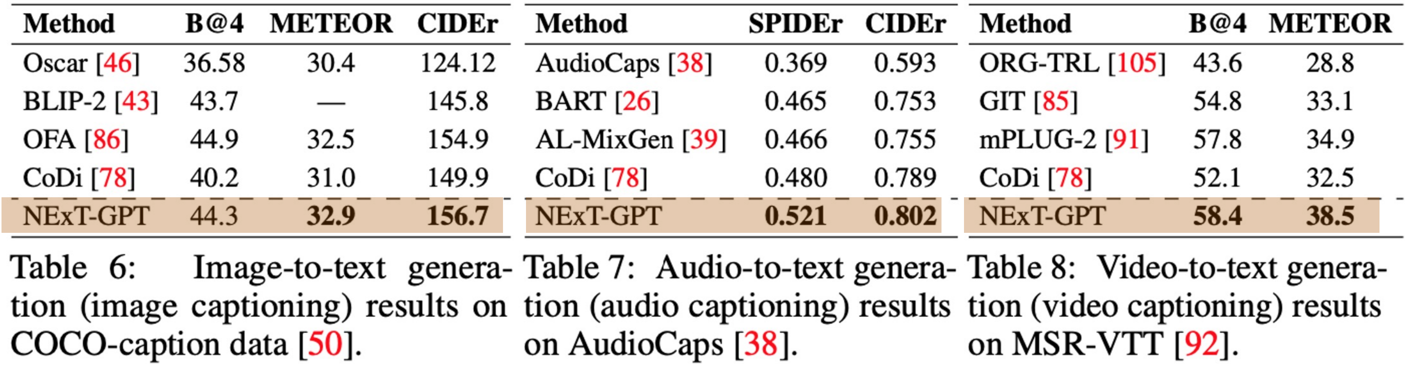
| Method | FID (↓) | CLIPSIM (↑) |
|---|---|---|
| CogVideo [30] | 23.59 | 0.2631 |
| MakeVideo [74] | 13.17 | 0.3049 |
| Latent-VDM [68] | 14.25 | 0.2756 |
| Latent-Shift [2] | 15.23 | 0.2773 |
| CoDi [78] | — | 0.2890 |
| NExT-GPT | **13.04** | **0.3085** |

Table 5: Text-to-video generation results (zero-shot) on MSR-VTT [92].

* FID: Images / Lower FID, more similar generated images to real images
* FD: Audio / Lower FD, closer to real audio features
* IS: Images & Audio / Higher IS, realistic and varied audio
* CLIPSIM: Images & Videos / HIgher CLIPSIM, more semantically aligned with the prompt

# Experiments:
# Any-to-any Multimodal Generation

- X-to-Text Generation



| Method | B@4 | METEOR | CIDEr |
|---|---|---|---|
| Oscar [46] | 36.58 | 30.4 | 124.12 |
| BLIP-2 [43] | 43.7 | — | 145.8 |
| OFA [86] | 44.9 | 32.5 | 154.9 |
| CoDi [78] | 40.2 | 31.0 | 149.9 |
| NExT-GPT | 44.3 | **32.9** | **156.7** |

Table 6: Image-to-text generation (image captioning) results on COCO-caption data [50].

| Method | SPIDEr | CIDEr |
|---|---|---|
| AudioCaps [38] | 0.369 | 0.593 |
| BART [26] | 0.465 | 0.753 |
| AL-MixGen [39] | 0.466 | 0.755 |
| CoDi [78] | 0.480 | 0.789 |
| NExT-GPT | **0.521** | **0.802** |

Table 7: Audio-to-text generation (audio captioning) results on AudioCaps [38].

| Method | B@4 | METEOR |
|---|---|---|
| ORG-TRL [105] | 43.6 | 28.8 |
| GIT [85] | 54.8 | 33.1 |
| mPLUG-2 [91] | 57.8 | 34.9 |
| CoDi [78] | 52.1 | 32.5 |
| NExT-GPT | **58.4** | **38.5** |

Table 8: Video-to-text generation (video captioning) results on MSR-VTT [92].

* B@4: Text(captions) / Higher score, better quality text

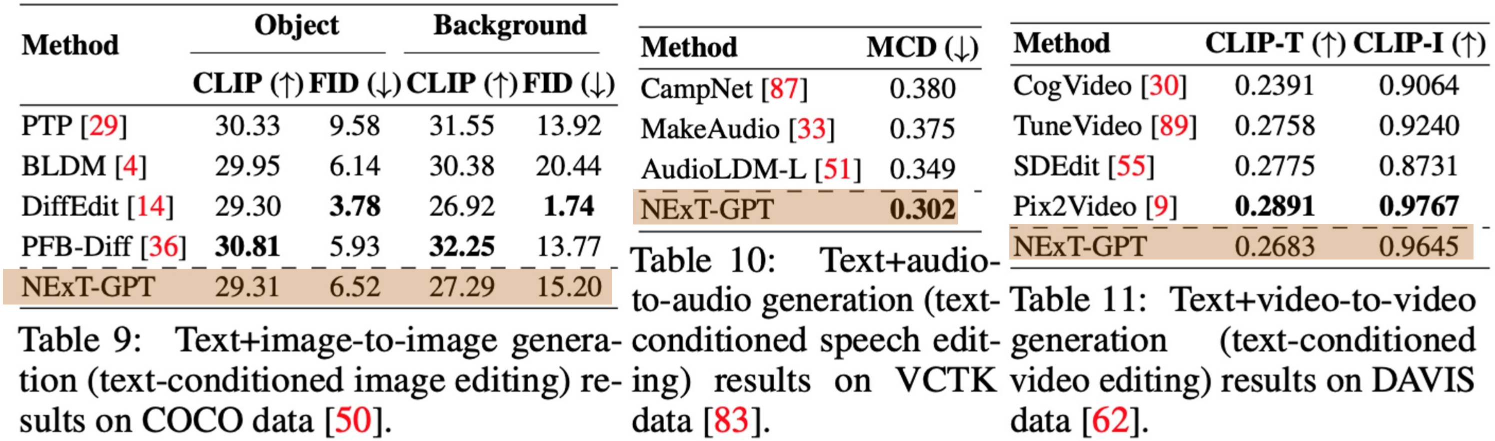
* METEOR: Text(captions) / Higher score, better descriptions

* SPIDEr: Audio captioning / Higher score, better audio descriptions

* CIDEr: Image & Audio captioning / Higher score, description in agreement with reference

# Experiments:
# Any-to-any Multimodal Generation

- Text+X to X Generation



| Method | Object | | Background | |
|---|---|---|---|---|
| | CLIP (↑) | FID (↓) | CLIP (↑) | FID (↓) |
| PTP [29] | 30.33 | 9.58 | 31.55 | 13.92 |
| BLDM [4] | 29.95 | 6.14 | 30.38 | 20.44 |
| DiffEdit [14] | 29.30 | **3.78** | 26.92 | **1.74** |
| PFB-Diff [36] | **30.81** | 5.93 | **32.25** | 13.77 |
| NExT-GPT | 29.31 | 6.52 | 27.29 | 15.20 |

Table 9: Text+image-to-image generation (text-conditioned image editing) results on COCO data [50].

| Method | MCD (↓) |
|---|---|
| CampNet [87] | 0.380 |
| MakeAudio [33] | 0.375 |
| AudioLDM-L [51] | 0.349 |
| NExT-GPT | **0.302** |

Table 10: Text+audio-to-audio generation (text-conditioned speech editing) results on VCTK data [83].

| Method | CLIP-T (↑) | CLIP-I (↑) |
|---|---|---|
| CogVideo [30] | 0.2391 | 0.9064 |
| TuneVideo [89] | 0.2758 | 0.9240 |
| SDEdit [55] | 0.2775 | 0.8731 |
| Pix2Video [9] | **0.2891** | **0.9767** |
| NExT-GPT | 0.2683 | 0.9645 |

Table 11: Text+video-to-video generation (text-conditioned video editing) results on DAVIS data [62].

* CLIP: Image & Video / Higher score, better alignment

* FID: Images / Lower FID, more similar generated images to real images

* MCD: Audio / Lower FID, less distortion and natural sounding audio

University of Virginia | ENGINEERING Department of Computer Science

# Experiments:
# Any-to-any Multimodal Generation

- Human Evaluation on Complex Any-to-any QA

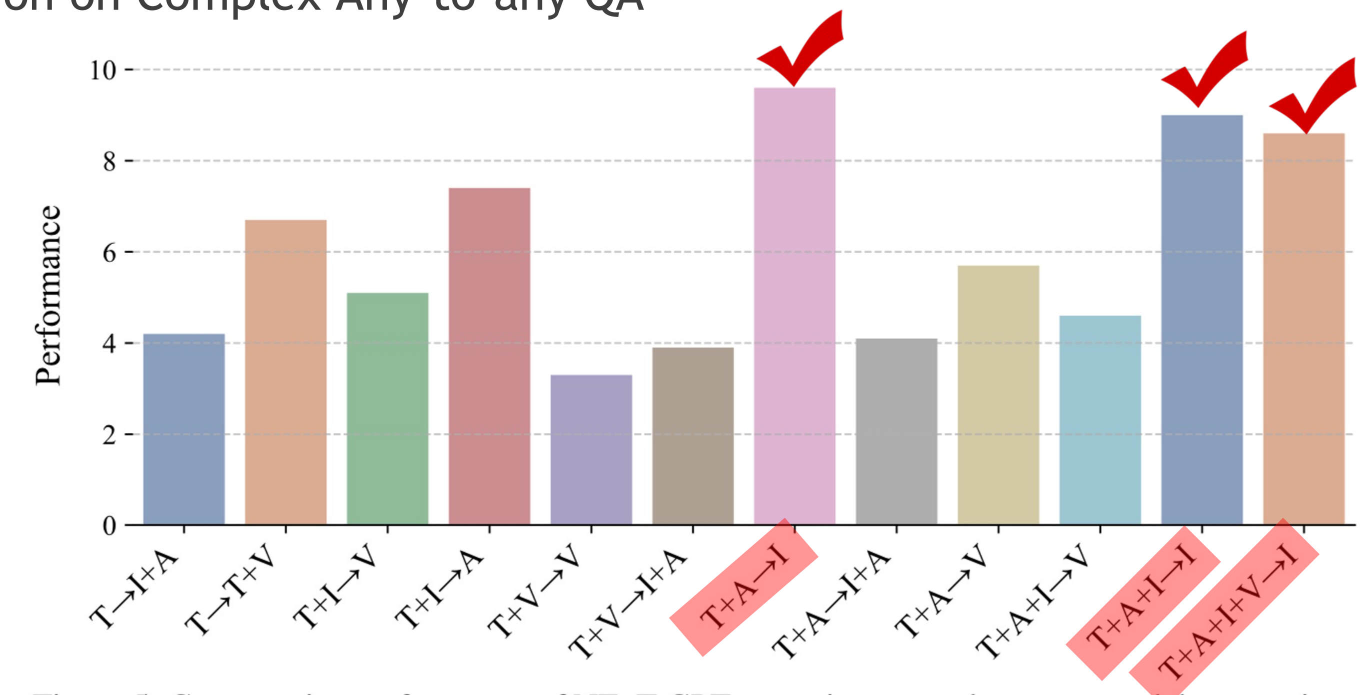


Figure 5: Comparative performance of NExT-GPT on various complex cross-modal conversions.

# Conclusion

- Contributions
    - NExT-GPT
    - MosIT Dataset
- Future Works
    - Expanding NExT-GPT to additional modalities
    - Broader range of tasks
    - Incorporate different types and sizes of LLMs
    - Expand MosIT dataset

# Key Takeaways

- Flamingo
  - Open-ended vision & language model that can reach SOTA performance on various image-language tasks with zero or few-shots.
- VisionLLM
  - Open-ended vision & language model that can perform both vision-language and vision-centric tasks with natural language instruction.
- Visual Instruction Tuning (LLaVA)
  - Vision & Language Model trained through visual instruction tuning, following human intent
  - Pipeline to create language-image instruction following data
- NExT-GPT
  - Any-to-any model understanding and generating text/images/audio/video
  - Construct & generate dataset (MosIT)

# Thank you!

UNIVERSITY of VIRGINIA | ENGINEERING
Department of Computer Science