

Multimodal Language Models

- Flamingo: a Visual Language Model for Few-Shot Learning
- Visual Instruction Tuning
- VITA-1.5: Towards GPT-4o Level Real-Time Vision and Speech Interaction

Zhoubin Kou, Bryant Lisk

March 26, 2025

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

Few-shot In-Context Learning in LLMs

Large language models (LLMs) excel at few-shot in-context learning (ICL) -- learning from a few examples provided in context at inference, without any weight updates.



Can we extend in-context learning to include visual inputs as well?

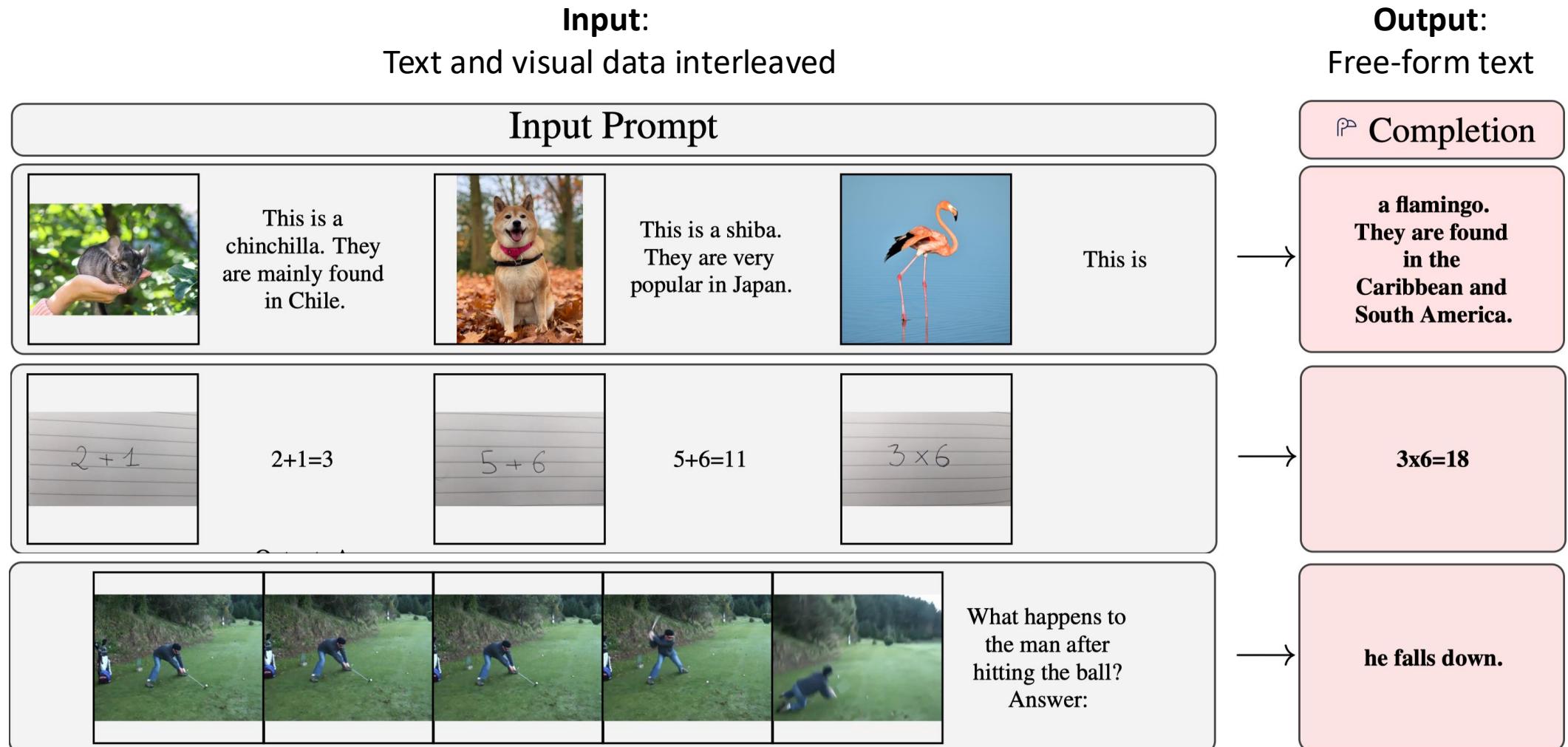
Motivation

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 or videos) along with language input, and produce language output.
- **Generalist ... rapidly adapted to different multimodal tasks:** one model can address multiple tasks (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 (32 examples are used)

Flamingo: a Visual Language Model for Few-Shot Learning

What is Flamingo?



Flamingo comes in many colors depending on what it eats

How does it work?



The perception



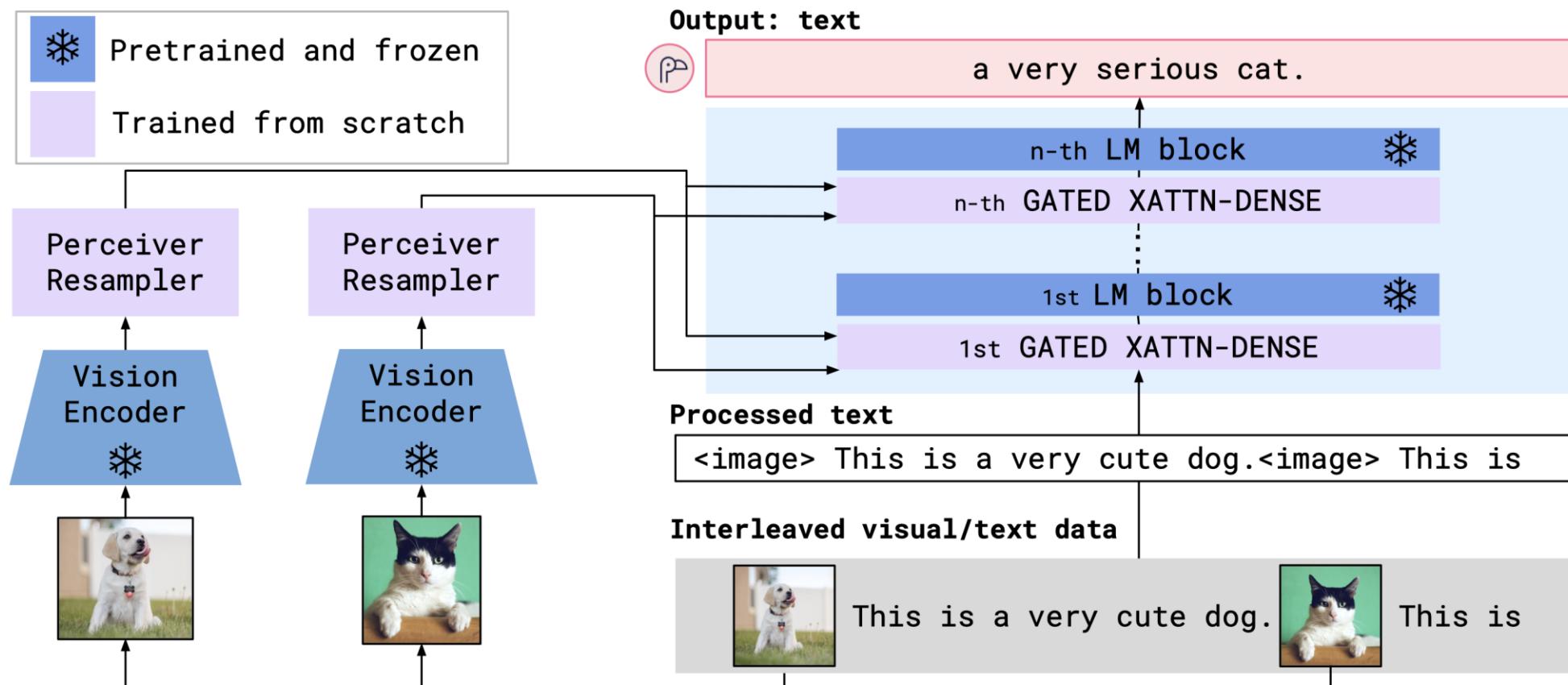
The “reasoning part”
and “knowledge source”

Flamingo: a Visual Language Model for Few-Shot Learning

How does it work?

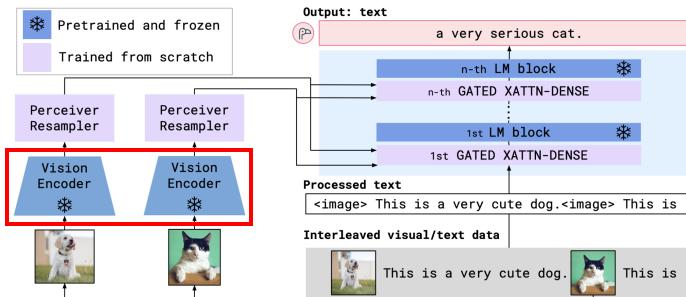
Model overview:

- Pretrained parts of the model are frozen, the Vision Encoder and LLM.

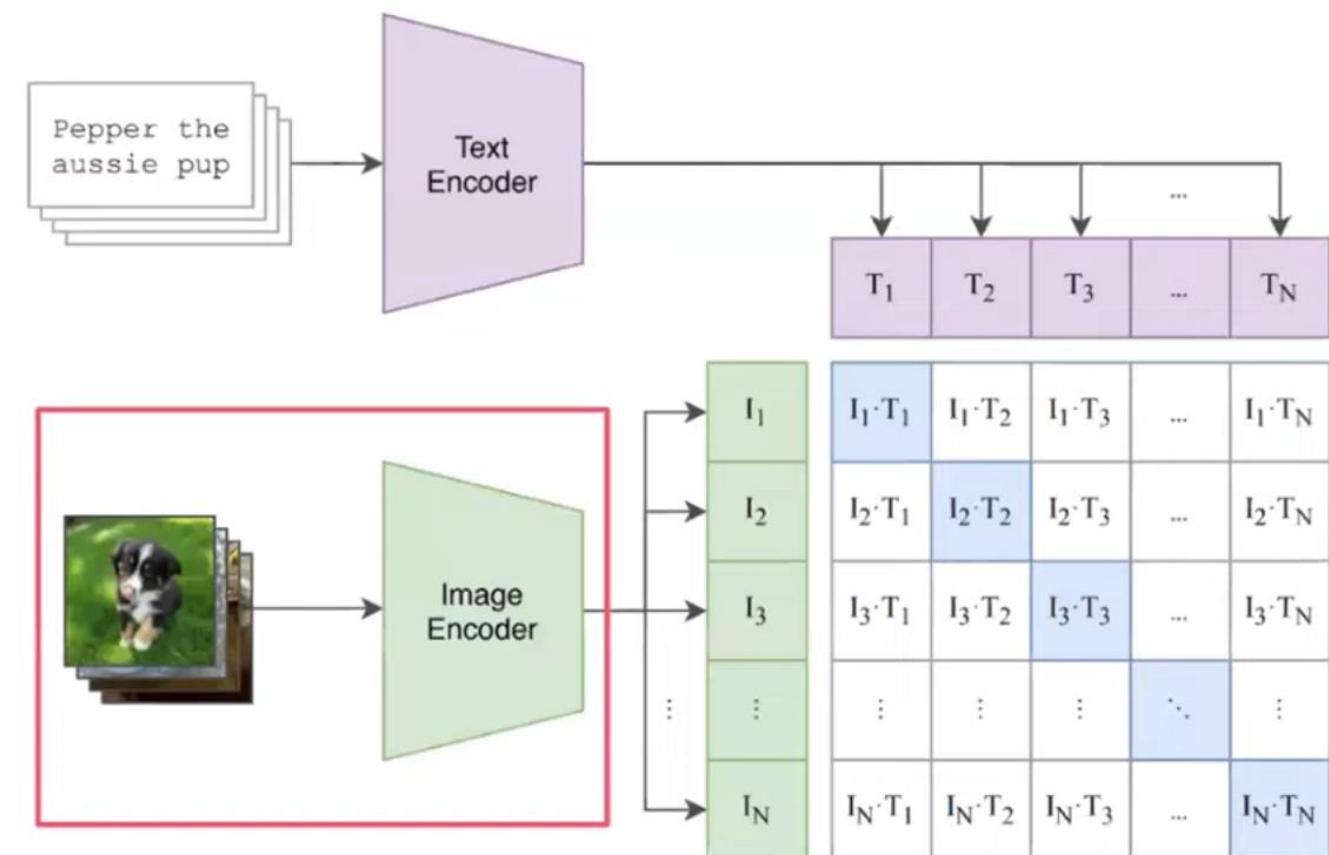


Flamingo: a Visual Language Model for Few-Shot Learning

Visual processing – Vision Encoder

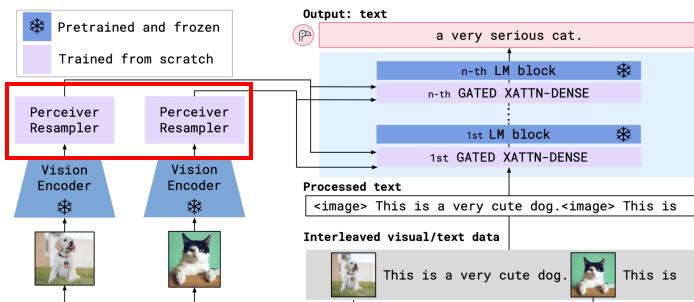


pretrained and frozen Normalizer Free ResNet (NFNet)



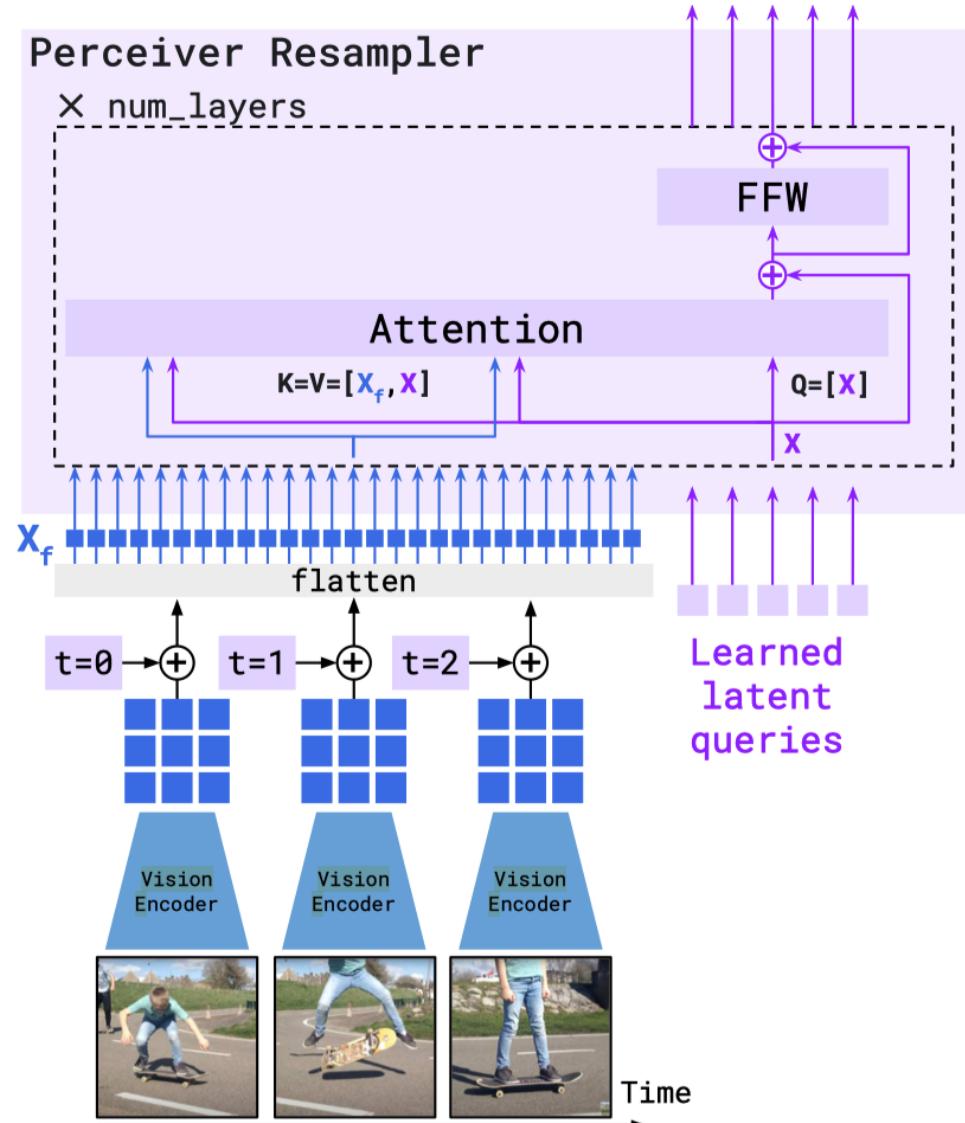
Flamingo: a Visual Language Model for Few-Shot Learning

Visual processing – Perceiver Resampler



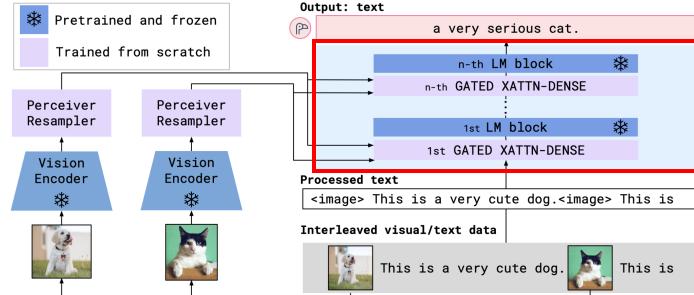
```
def perceiver_resampler(
    x_f, # The [T, S, d] visual features (T=time, S=space)
    time_embeddings, # The [T, 1, d] time pos embeddings.
    x, # R learned latents of shape [R, d]
    num_layers, # Number of layers
):
    """The Perceiver Resampler."""

    # Add the time position embeddings and flatten.
    x_f = x_f + time_embeddings
    x_f = flatten(x_f) # [T, S, d] -> [T * S, d]
    # Apply the Perceiver Resampler layers.
    for i in range(num_layers):
        # Attention.
        x = x + attention_i(q=x, kv=concat([x_f, x]))
        # Feed forward.
        x = x + ffw_i(x)
    return x
```



Flamingo: a Visual Language Model for Few-Shot Learning

Gated Cross-attention Dense Blocks



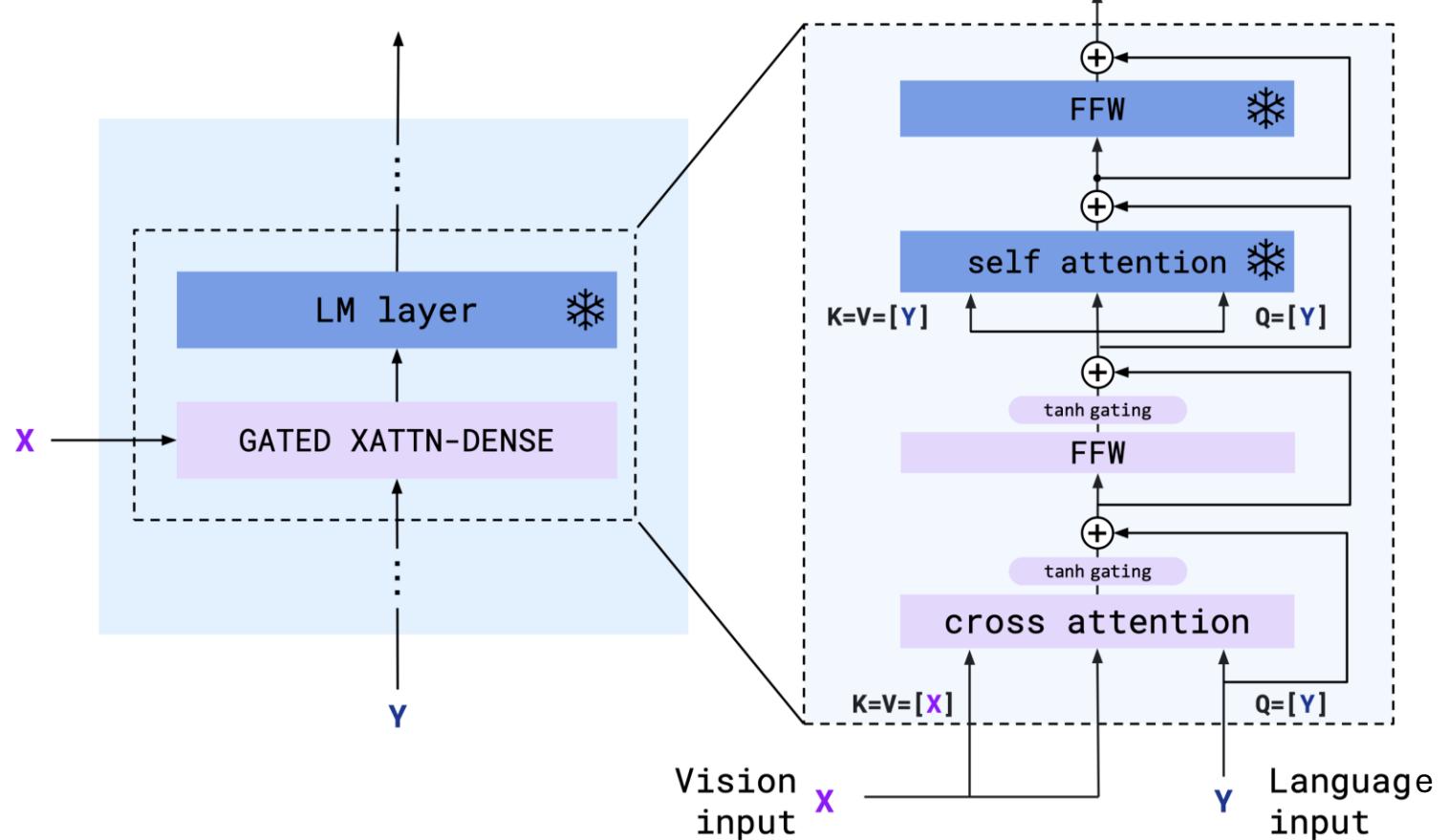
```
def gated_xattn_dense(
    y, # input language features
    x, # input visual features
    alpha_xattn, # xattn gating parameter - init at 0.
    alpha_dense, # ffw gating parameter - init at 0.
):
    """Applies a GATED XATTN-DENSE layer."""

    # 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
```

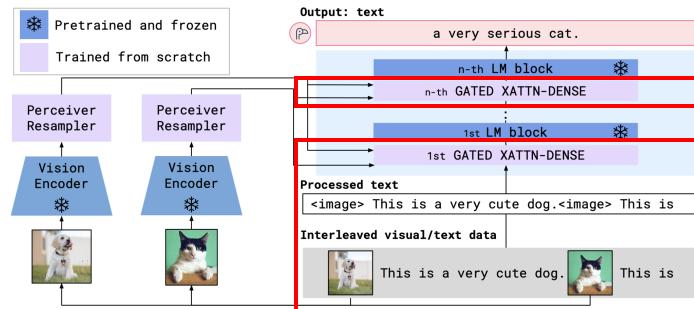
Leveraging an existing language model

- At initialization, tanh gates are all 0.



Flamingo: a Visual Language Model for Few-Shot Learning

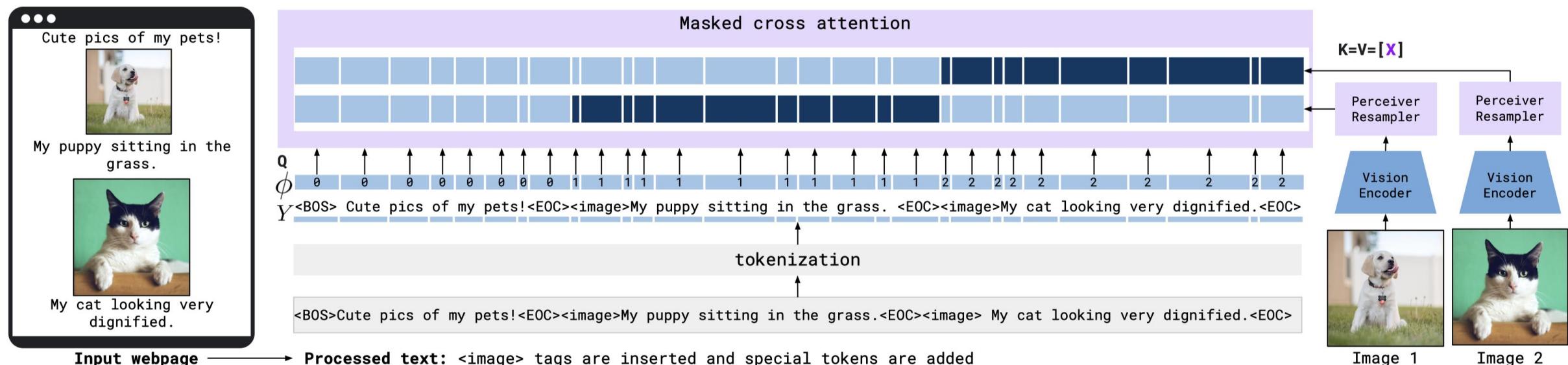
Per-image/video Masking



Deal with interleaved visual and text sequence

- Each text token cross-attend to the image that precedes it in the interleaved sequence

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



Training Data

M3W: Massive Multimodal Web Dataset

- 44M scraped webpages with interleaved text and images
- 180M images in total. (4 on average per webpage)

ALIGN Dataset

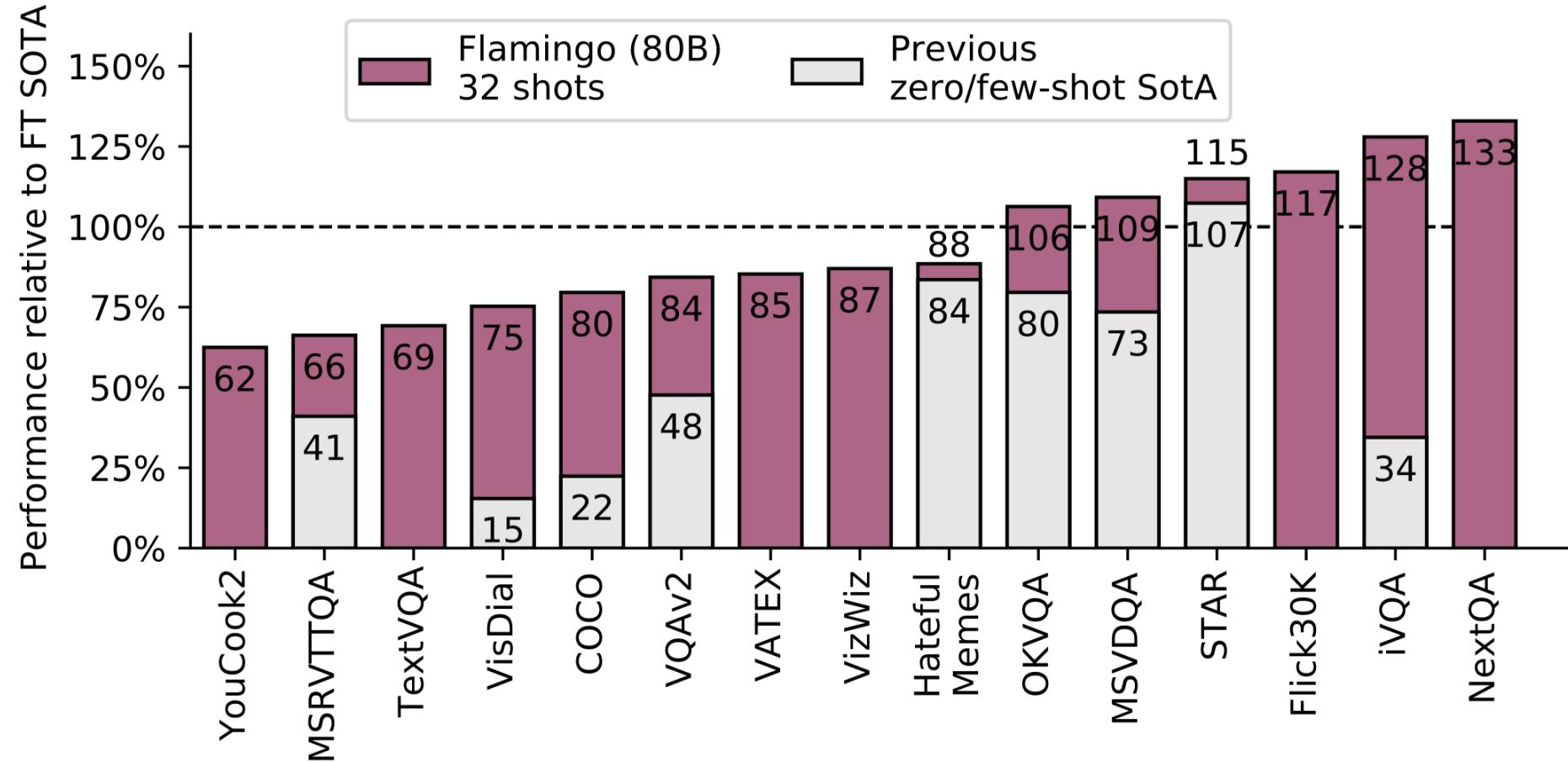
- 1.8 billion images paired with alt-text
- Complemented with in house dataset of Long Text & Image Pairs which consists of 312 million image and text pairs

Video-Text pairs

- In house datasets of 27 million short videos paired with sentence descriptions

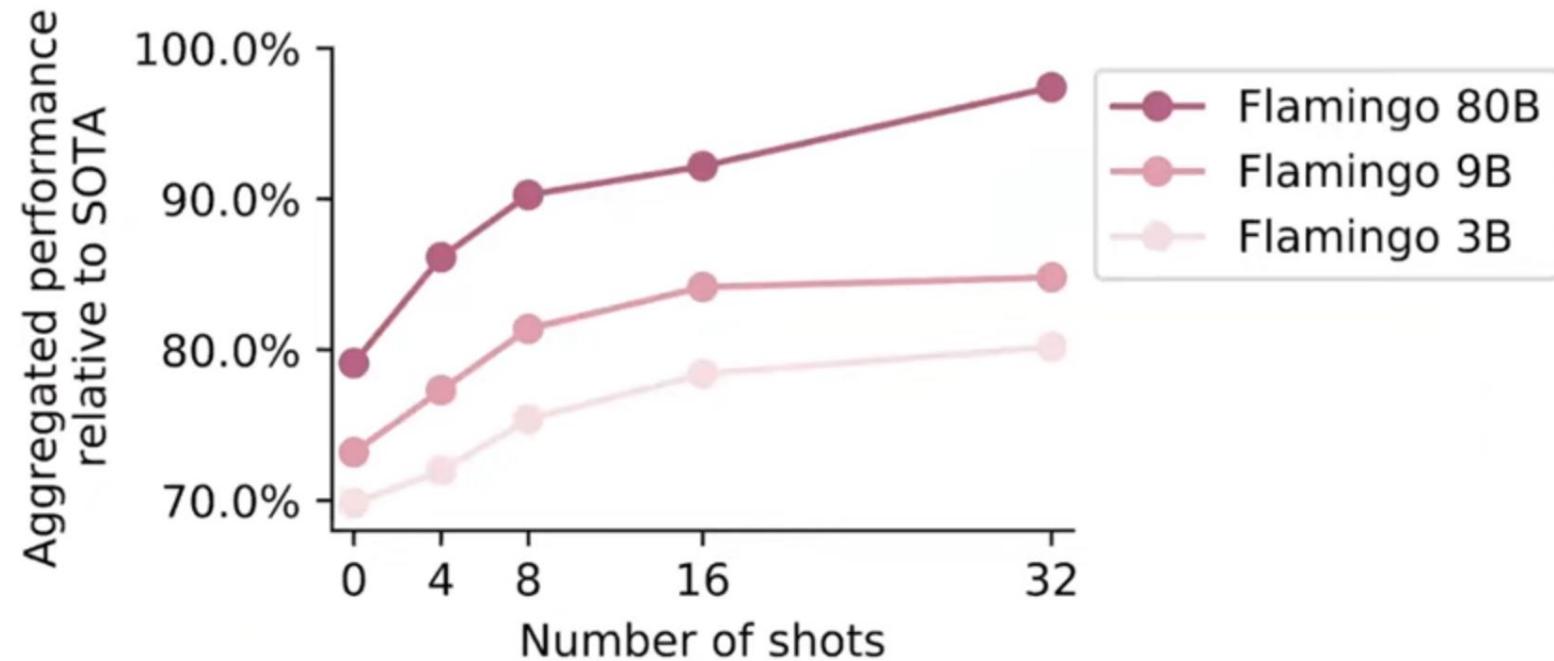
Flamingo: a Visual Language Model for Few-Shot Learning

Experiments



Flamingo: a Visual Language Model for Few-Shot Learning

Experiments



Flamingo: a Visual Language Model for Few-Shot Learning

Experiments – Few-shot learning on vision-language tasks

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

Experiments – Fine-tuning Flamingo as a pretrained vision-language model

Method	VQAV2		COCO test	VATEX test	VizWiz		MSRVTTQA test	VisDial		YouCook2 valid	TextVQA		HatefulMemes test seen
	test-dev	test-std			test-dev	test-std		valid	test-std		valid	test-std	
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 [†] [133]	81.3 [†] [133]	149.6[†] [119]	81.4 [†] [153]	57.2 [†] [65]	60.6 [†] [65]	46.8 [51]	75.2 [79]	75.4[†] [123]	138.7 [132]	54.7 [137]	73.7 [84]	84.6 [†] [152]

Finetuned Flamingo achieves the state of the art on 5 additional benchmarks where few-shot was not enough

Flamingo: a Visual Language Model for Few-Shot Learning

Experiments - Ablation Studies

Ablated setting	<i>Flamingo-3B</i> original value	Changed value	Param. count ↓	Step time ↓	COCO CIDEr↑	OKVQA top1↑	VQAv2 top1↑	MSVDQA top1↑	VATEX CIDEr↑	Overall score↑	
	<i>Flamingo-3B</i> 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

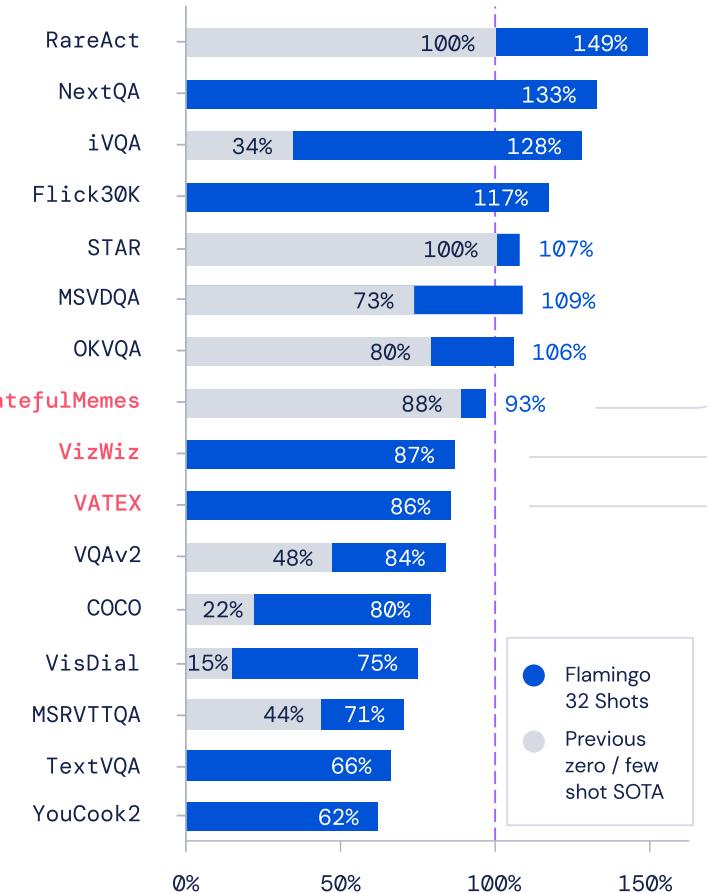
Limitations

- **Performance on classification**
 - lags behind contrastive models (e.g., CLIP) on tasks that require precise classification
- **Limitations of In-Context Learning**
 - performance is highly sensitive to the selection and order of examples, incurs high inference costs (especially with many shots, such as 32-shot settings),
 - does not scale as well as fine-tuning methods.
- **Inherited Weaknesses from the Base Language Model**
 - hallucination (generating unrelated or false content)
 - lack of factual grounding
 - poor generalization to longer sequences
 - low sample efficiency during training

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.

Performance relative to SOTA



Visual Instruction Tuning

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

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

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

Visual Instruction Tuning

Introduction

- One of the core aspirations in AI is to develop a general-purpose assistant that can effectively follow both vision and language instructions and able to complete real world tasks.
- LLMs such as GPT-4 have had great success in aligning LLMs with human instructions, but are entirely text based
- **Visual instruction tuning** attempts to extend instruction tuning to the language-image multimodal space to work towards building a general purpose visual assistant

Visual Instruction Tuning

Contributions

- **Multimodal instruction following data:**
 - A data reformation perspective and pipeline to convert image-text pairs to an appropriate instruction-following format, using ChatGPT/GPT-4
- **Large multimodal models:**
 - Building a large multimodal model by connecting open-set visual encoder by connecting the open-set visual encoder of CLIP with the language decoder Vicuna, and fine tuning on generated instructional vision-language data
- **Multimodal instruction following benchmark:**
 - Two challenging benchmarks with a diverse selection of paired images, instructions and detailed annotations
- **Open source**

Visual Instruction Tuning

Related Work

- **Multimodal instruction-following:**
 - Can be categorized in two classes: End to end trained models, which are separately explored in each research topic and systems that coordinate various models via LagChain.
 - Focus of paper is to develop an end-to-end trained language vision multimodal model for multiple tasks
- **Instruction tuning:**
 - Instruction tuning has improved zero and few shot learning for LLMs
 - While many models are trained on image text pairs and present promising task transfer generalization performance, they aren't explicitly tuned with vision-language instruction data and their performance in multimodal tasks falls short of their performance in language only tasks.

Visual Instruction Tuning

Visual Instruction Data-Generation

- Prompt GPT-4 to create list of questions to instruct the assistant to describe the image content
- Leverage language-only GPT-4/ChatGPT as the strong teacher to create instruction-following data involving visual content
- Two types of symbolic representations of encoding an image into its visual features:
 - **Captions** typically describe the visual scene
 - **Bounding boxes** usually localize the objects in the scene, encoding the object concept and spatial location

Context type 1: Captions

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

Luggage surrounds a vehicle in an underground parking area

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

The sport utility vehicle is parked in the public garage, being packed for a trip

Some people with luggage near a van that is transporting it.



Context type 2: Boxes

person: [0.681, 0.242, 0.774, 0.694], backpack: [0.384, 0.696, 0.485, 0.914], suitcase: ...<omitted>

Response type 1: conversation

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

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

Response type 2: detailed description

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

Response type 3: complex reasoning

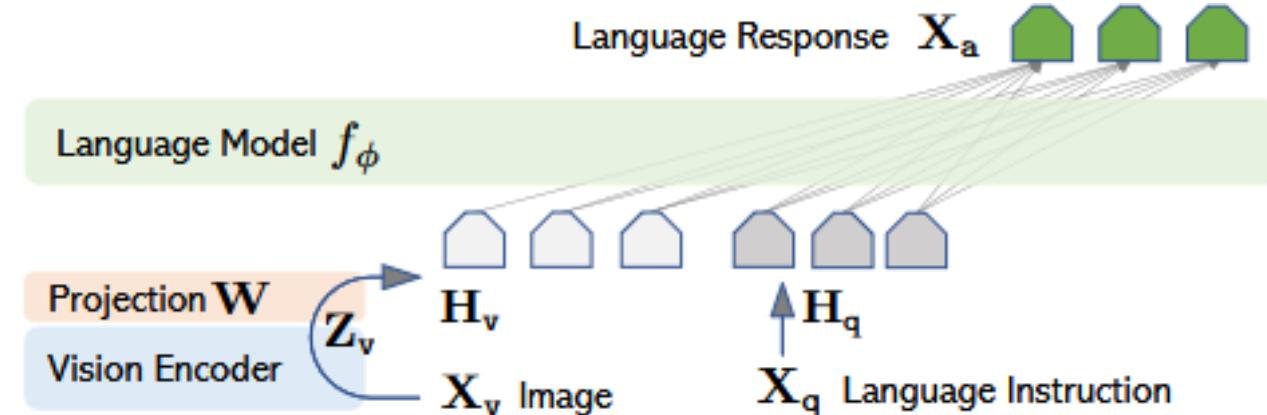
Question: What challenges do these people face?

Answer: In the image, a group of people is standing outside a black SUV in a parking area, surrounded by various pieces of luggage, including suitcases and backpacks. They are facing the challenge of fitting all their luggage into the black SUV. There are multiple suitcases and backpacks to be packed, which suggests that the group has a significant amount of belongings ...<omitted>

Visual Instruction Tuning

- **Architecture**

- A linear layer connects image features into the word embedding space
- A trainable projection matrix W converts Z_v into H_v



$$\mathbf{H}_v = \mathbf{W} \cdot \mathbf{Z}_v, \text{ with } \mathbf{Z}_v = g(\mathbf{X}_v)$$

Visual Instruction Tuning

Training

$$\mathbf{X}_{\text{instruct}}^t = \begin{cases} \text{Randomly choose } [\mathbf{X}_q^1, \mathbf{X}_v] \text{ or } [\mathbf{X}_v, \mathbf{X}_q^1], & \text{the first turn } t = 1 \\ \mathbf{X}_q^t, & \text{the remaining turns } t > 1 \end{cases} \quad (2)$$

$$p(\mathbf{X}_a | \mathbf{X}_v, \mathbf{X}_{\text{instruct}}) = \prod_{i=1}^L p_{\theta}(\mathbf{x}_i | \mathbf{X}_v, \mathbf{X}_{\text{instruct}, < i}, \mathbf{X}_{a, < i}), \quad (3)$$

L = length of the sequence

\mathbf{X}_a = target answers

θ = trainable parameters

$\mathbf{X}_{\text{instruct}, < i}$ = instruction tokens before current prediction token

$\mathbf{X}_{a, < i}$ = answer tokens before current prediction token

\mathbf{X}_v = the visual data

Visual Instruction Tuning

```
Xsystem-message <STOP>  
Human : Xinstruct1 <STOP> Assistant: Xa1 <STOP>  
Human : Xinstruct2 <STOP> Assistant: Xa2 <STOP> ...
```

Visual Instruction Tuning

Two Stage Instruction-Tuning Procedure

Stage 1: Pre-training for Feature Alignment

- Train 595,000 CC3M image text pairs converted into instruction-following data
- Each sample treated as single-turn conversation
- For an image, a question is randomly sampled asking the assistant to briefly describe the image
- The ground truth prediction is the original caption
- LLM and visual encoder weights are frozen, and training attempts to maximize the likelihood with trainable parameters $\theta = W$
- This stage is training a compatible visual tokenizer for the LLM

Stage 2: Fine-tuning End-to-End

- Chatbot developed by fine-tuning on 158,000 language image instruction-following data
- ScienceQA benchmark training organized as single turn conversation with question and context as $X_{instruct}$ and the reasoning and answer as X_a

Visual Instruction Tuning

Visual input example, Extreme Ironing:



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

User
LLaVA

What is unusual about this image?

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

[Start a new conversation, and clear the history]

User
LLaVA

What's happening in the scene?

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

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.

Visual Instruction Tuning

LLaVA-Bench

- **LLaVA-Bench (COCO):**
 - Randomly select 30 images from COCO-Val 2014 and generate 3 types of questions for each
- **LLaVA-Bench (In-the-Wild):**
 - Diverse set of 24 images with 60 questions in total associated with highly-detailed and manually-curated description
 - Designed to be challenging and test a model's weaknesses

Visual Instruction Tuning

LLaVA-Bench COCO:

	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)

LLaVA-Bench In-The-Wild:

	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

Visual Instruction Tuning

Challenging examples from LLaVA-Bench (In-the-Wild):



ICHIRAN Ramen [[source](#)]



Filled fridge [[source](#)]

Annotation	<p>A close-up photo of a meal at ICHI-RAN. The chashu ramen bowl with a spoon is placed in the center. The ramen is seasoned with chili sauce, chopped scallions, and served with two pieces of chashu. Chopsticks are placed to the right of the bowl, still in their paper wrap, not yet opened. The ramen is also served with nori on the left. On top, from left to right, the following sides are served: a bowl of orange spice (possibly garlic sauce), a plate of smoke-flavored stewed pork with chopped scallions, and a cup of matcha green tea.</p>	<p>An open refrigerator filled with a variety of food items. In the left part of the compartment, towards the front, there is a plastic box of strawberries with a small bag of baby carrots on top. Towards the back, there is a stack of sauce containers. In the middle part of the compartment, towards the front, there is a green plastic box, and there is an unidentified plastic bag placed on it. Towards the back, there is a carton of milk. In the right part of the compartment, towards the front, there is a box of blueberries with three yogurts stacked on top. The large bottle of yogurt is Fage non-fat yogurt, and one of the smaller cups is Fage blueberry yogurt. The brand and flavor of the other smaller cup are unknown. Towards the back, there is a container with an unknown content.</p>
Question 1	What's the name of the restaurant?	What is the brand of the blueberry-flavored yogurt?
Question 2	Describe this photo in detail.	Is there strawberry-flavored yogurt in the fridge?

Method	Subject			Context Modality			Grade		Average
	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	
<i>Representative & SoTA methods with numbers reported in the literature</i>									
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
<i>Results with our own experiment runs</i>									
GPT-4 [†]	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 [†] (complement)	90.36	95.50	88.55	89.05	87.80	91.08	92.22	88.73	90.97
LLaVA+GPT-4 [†] (judge)	91.56	96.74	91.09	90.62	88.99	93.52	92.73	92.16	92.53

Visual Instruction Tuning

Limitations

- LLaVA can view images as a "bag of patches" and can fail to grasp complex semantics
 - For LLaVA Bench In-The-Wild the model said there was strawberry yogurt in the fridge when there was only strawberries and yogurt
- While LLaVA can take in text and images, it can't take in video or output in the image modality

Conclusion

- Visual instruction tuning effectively increases performance in multimodal question answering and chat capabilities
- First benchmark to study multimodal instruction-following capabilities in LLaVA-Bench
- This paper is an initial step in visual instruction tuning and mainly focuses on real-life tasks

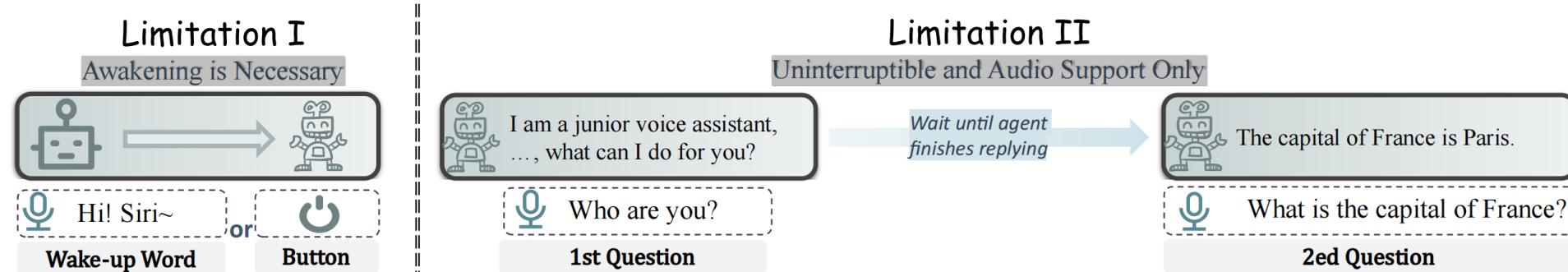
VITA-1.5: Towards GPT-4o Level Real-Time Vision and Speech Interaction

Chaoyou Fu^{1,♦}, Haojia Lin³, Xiong Wang², Yi-Fan Zhang⁴, Yunhang Shen²
Xiaoyu Liu¹, Haoyu Cao², Zuwei Long², Heting Gao², Ke Li², Long Ma²,
Xiawu Zheng³, Rongrong Ji³, Xing Sun^{2,†}, Caifeng Shan¹, Ran He⁴

¹NJU, ²Tencent Youtu Lab, ³XMU, ⁴CASIA

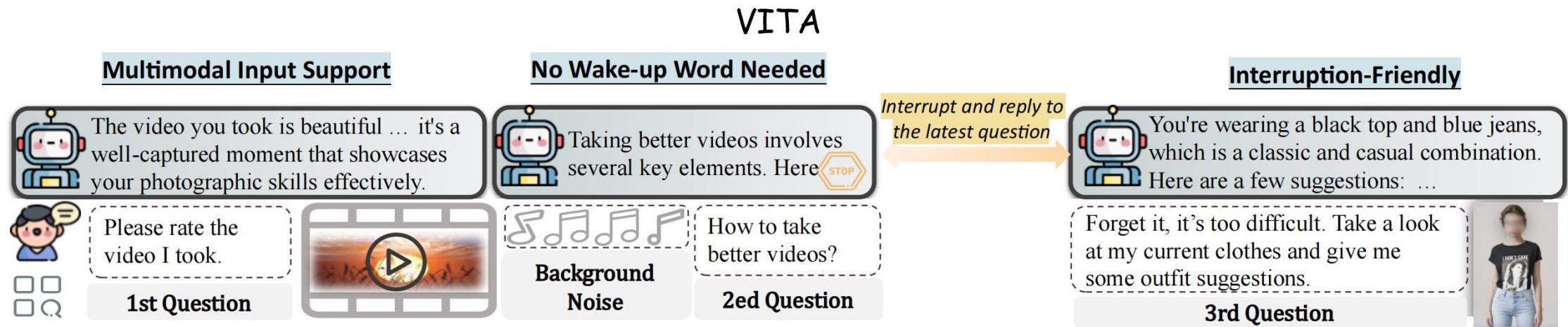
♦ Project Leader † Corresponding Author

Motivation



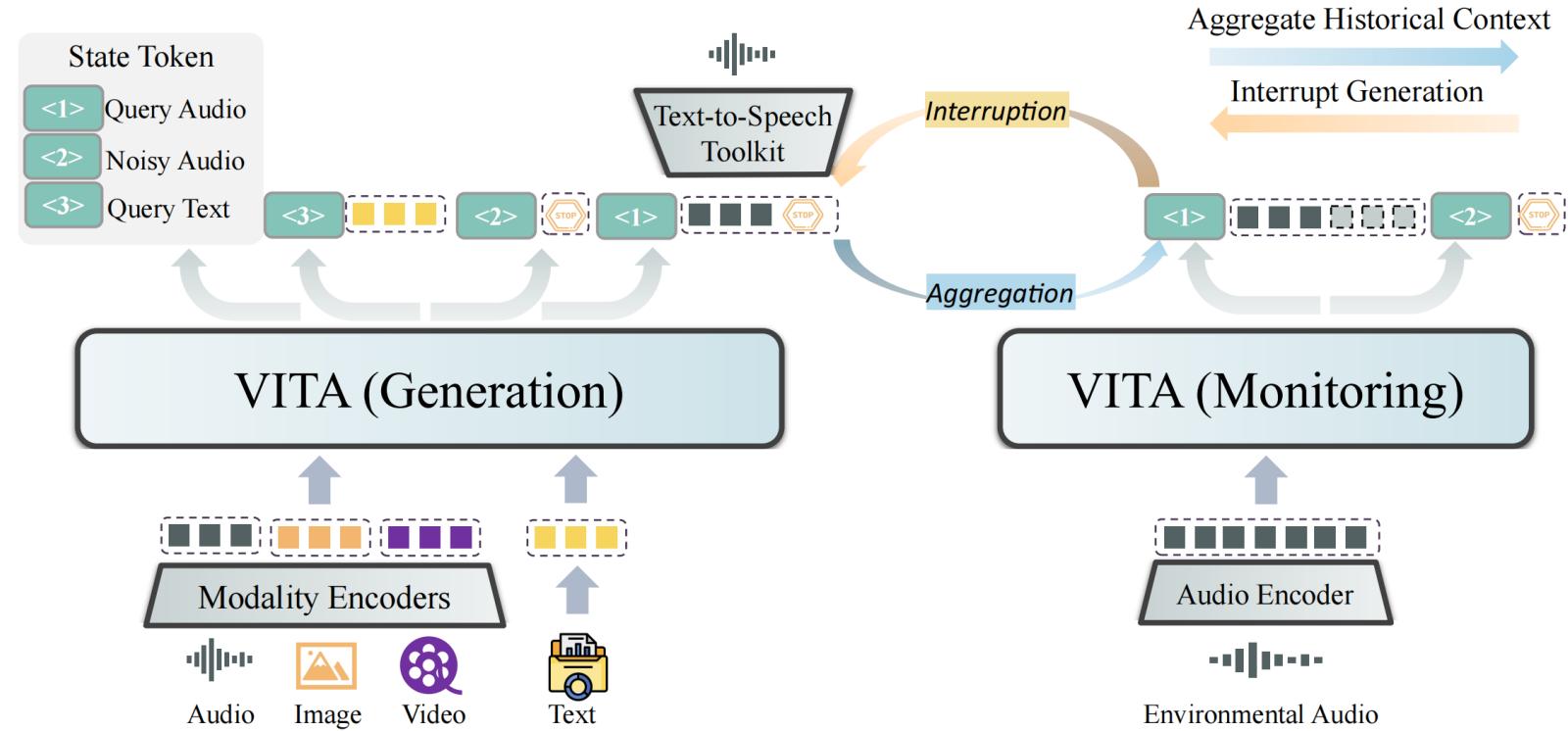
- **The Need for Multimodal Interaction**
 - Increasing demand for natural, real-time human-computer interaction
 - Existing multimodal models mainly focus on visual and textual inputs, with limited emphasis on speech
- **Limitations of Current Approaches**
 - Separate ASR and TTS modules lead to high latency and incoherent interactions
 - Fundamental modality differences create training conflicts when integrating speech with vision and text
- **Goal of VITA**
 - Build an end-to-end multimodal system that seamlessly fuses vision, language, and speech
 - Achieve near real-time speech-to-speech interaction without relying on separate ASR/TTS modules

Key features:



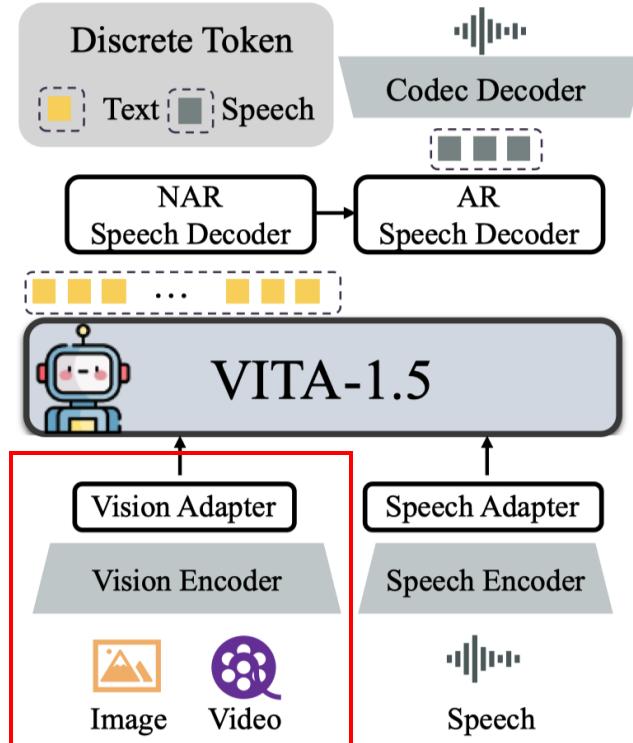
- Omni Multimodal Understanding.
- Non-awakening Interaction.
- Audio Interrupt Interaction.

Model Architecture



System view of VITA1.5

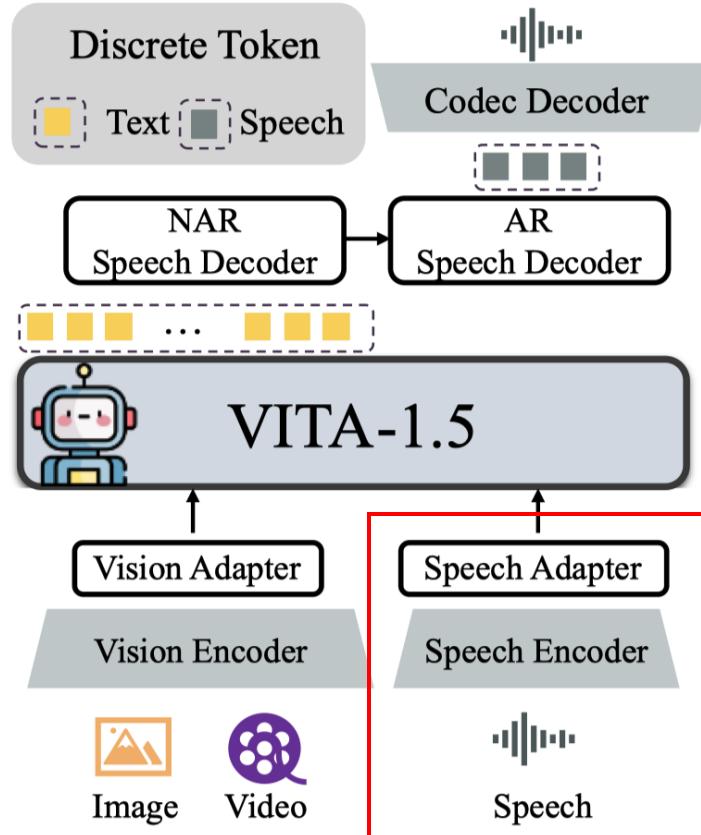
Model Architecture - Visional Modality



Overall Architecture of VITA-1.5.

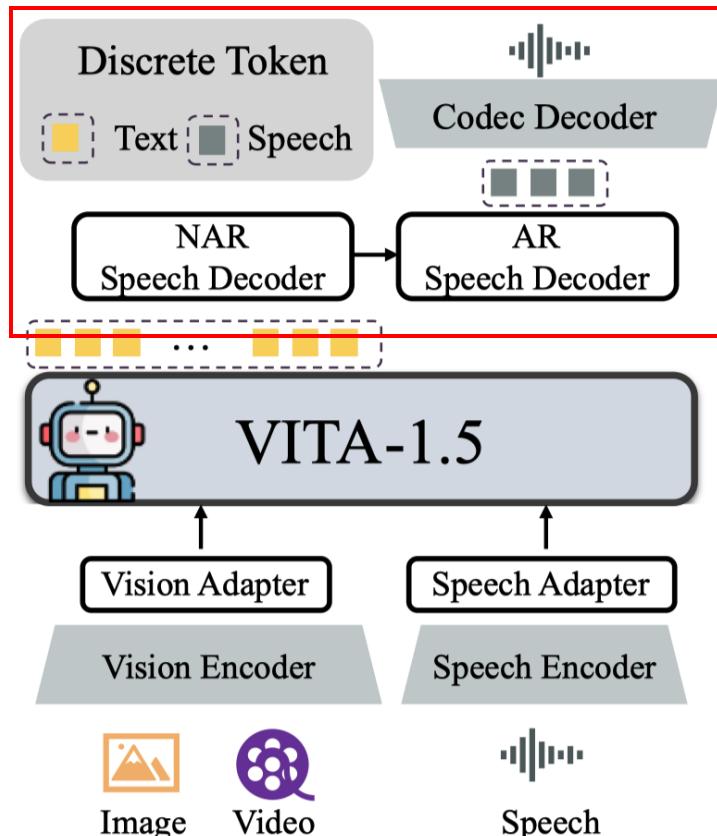
- **Visual Encoder**
 - InternViT300M visual encoder
 - Input image size of 448×448 pixels
 - Generates 256 visual tokens per image
 - High-resolution images
 - dynamic patching strategy
- **Video Processing**
 - Treated as multiple-image input
 - Samples between 4 and 16 frames depending on video length
 - No dynamic patching to video frames
- **Vision Adapter**
 - Two-layer MLP to map visual features to visual tokens

Model Architecture - Audio Modality



- **Speech Encoder**
 - Consists of:
 - Multiple downsampling convolutional layers (4x downsampling)
 - 24 Transformer blocks (hidden size of 1024)
 - Audio encoder has 350M parameters and output framerate of 12.5Hz.
 - Mel-filter bank features used as input of audio encoder with window size of 25ms and shift of 10ms
- **Speech Adapter**
 - Consists of multiple convolutional layers with 2x downsampling

Model Architecture - Audio Modality



- **Speech Decoder:**
 - Uses TiCodec for codec model, customizing single codebook with size of 1024
 - Codec model encodes continuous speech signals into discrete speech tokens
 - 40Hz frequency
 - 24KHz decode sample rate
- **Non-Autoregressive (NAR) Speech Decoder**
 - Processes text tokens globally
 - Models semantic features
 - Aims to generate initial distribution of speech tokens
- **Autoregressive (AR) Speech Decoder**
 - Generates higher quality speech tokens from tokens produced by NAR decoder
 - Final sequence of tokens decoded into continuous speech signal flow using speech decoder of Codec model

Training Data for multimodal instruction tuning

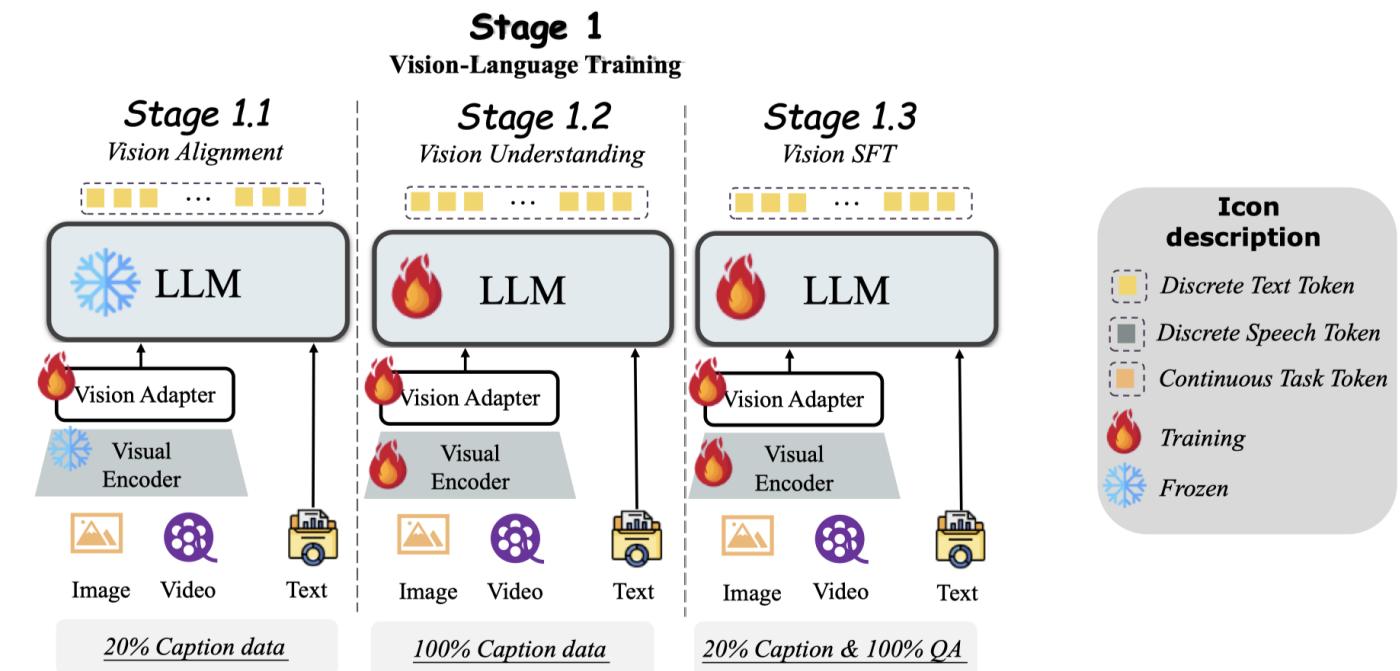
Data Scenario	QA Type	Dataset Name	Questions (K)	Language
General Image	Description	ShareGPT4V	99.50	Eng
		ALLaVA-Caption	697.40	Eng
		ShareGTP4o-Image	55.50	Eng
		Synthetic Data	593.70	CN
	QA	LLaVA-150K	218.36	CN
		LLaVA-Mixture-sample	1872.10	Eng
		LVIS-Instruct	939.36	Eng
		ScienceQA	12.72	Eng
OCR & Diagram	Description	ChatQA	7.39	Eng
		LLaVA-OV General	1754.65	Eng
		LLaVA-OV Math Reasoning	1140.92	Eng
		Synthetic Data	212.68	CN
		Anyword-3M	1709.30	CN
	QA	ICDAR2019-LSVT	366.30	CN
		UReader	100.00	Eng
		SynDOG-EN	100.00	Eng
		SynDOG-CN	101.90	CN
General Video	Description	ICDAR2019-LSVT-QA	630.08	CN
		LLaVA-OV Doc Chart Screen	4431.50	Eng
		LLaVA-OV General OCR	404.20	Eng
	QA	ShareGemini	205.70	CN
Pure Text	Description	Synthetic Data	569.40	CN & Eng
		Synthetic Data	4336.30	CN & Eng
	QA	Synthetic Data	1574.20	CN & Eng
Total		22133.16	CN & Eng	CN & Eng

- 110,000 hours of internal speech-transcription paired ASR data, covering both Chinese and English, are incorporated to train the audio encoder and align the audio encoder with the LLM.
- 3,000 hours of text-speech paired data generated by a TTS system are used to train the speech decoder.

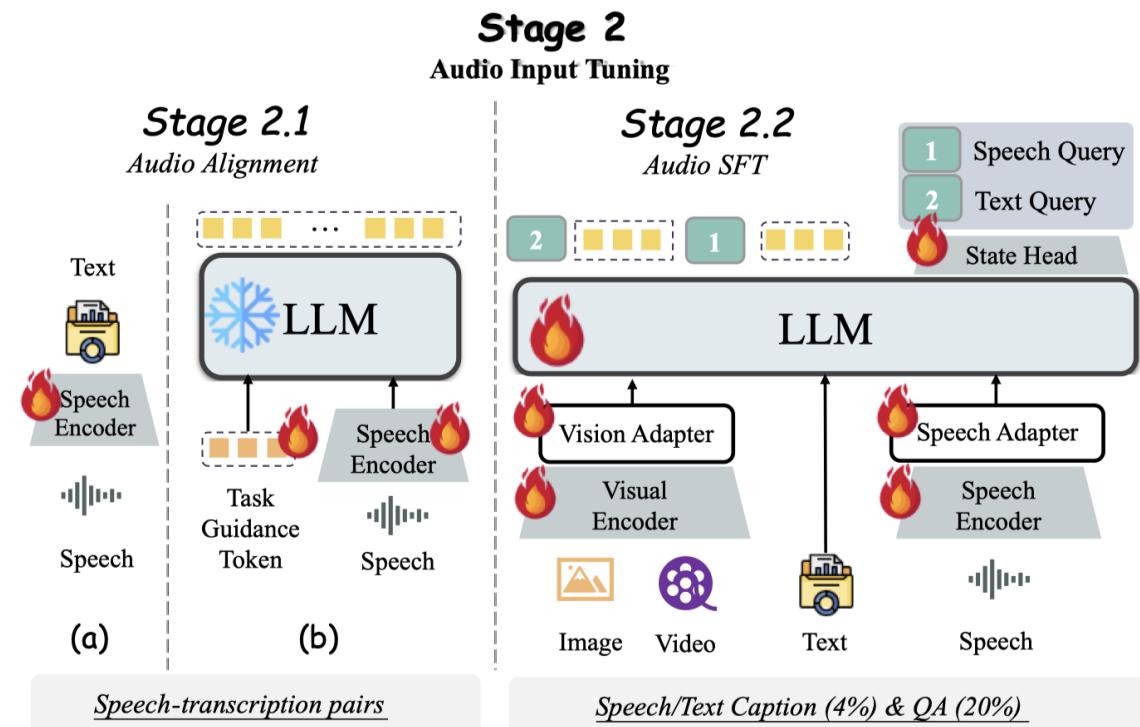
Training Strategies – Stage 1: Vision-Language Training

Key objectives:

- **Stage 1.1**
 - bridge the gap between vision and language
- **Stage 1.2**
 - Teach the LLM to transcribe image content
- **Stage 1.3**
 - enable the model not only to understand visual content but also to answer questions following instructions



Training Strategies – Stage 2: Audio Input Tuning



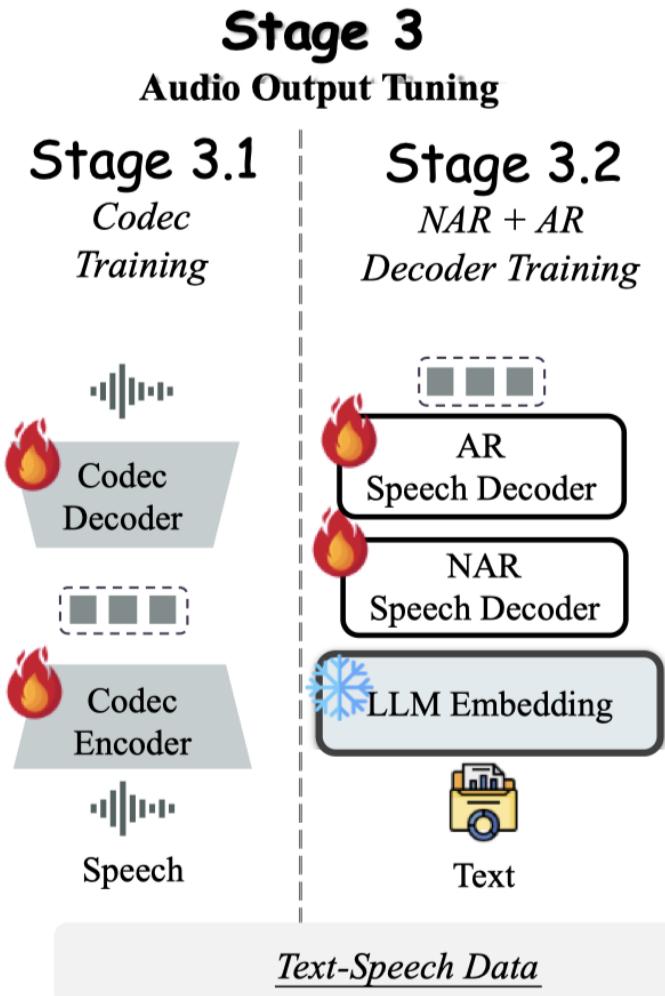
Key objectives:

- **Stage 2.1**
 - reduce the discrepancy between audio and language
 - enabling the LLM to understand audio inputs
- **Stage 2.2**
 - introduce the QA functionality with speech questions and text answers

Training Strategies – Stage 3: Audio Output Tuning

Key objectives:

- **Stage 3.1**
 - train a codec model with a single codebook using speech data
- **Stage 3.2**
 - leverage text-speech paired data to enable end-to-end speech generation



Evaluation – Baselines and Benchmarks

- Vision-Language Evaluation
 - Baseline
 - a series of open-source MLLMs
 - 5 closed-source MLLMs
 - Benchmarks
 - MME, MBench, MMStar, MMMU, MathVista, HallusionBench, AI2D, OCRBench, and MMVet.
- Speech Evaluation
 - Baseline
 - Wav2vec2-base, Mini-Omini2, Freeze-Omini, and VITA-1.0
 - Benchmarks
 - Mandarin Evaluation Sets
 - aishell-1, test net, test meeting
 - English Evaluation Sets
 - dev-clean, dev-other, test-clean, and test-other

Evaluation – Vision-Language Capabilities

Table 2: Evaluation on Image Understanding Benchmarks. VITA-1.5 shows performance comparable to the leading open-source models and advanced closed-source counterparts. MMB refers to MMBench, MMS to MMStar, Hal to HallusionBench, MathV to MathVista, and OCR to OCRCBench. Note that after the training of Stages 2 (Audio Input Tuning) and 3 (Audio Output Tuning), VITA-1.5 retains almost its original visual-language capabilities in Stage 1 (Vision-Language Training).

Method	LLM	MMB	MMS	MMMU	MathV	Hal	AI2D	OCR	MMVet	MME	Avg
VILA-1.5	Vicuna-v1.5-13B	68.5	44.2	41.1	42.5	39.3	69.9	460.0	45.0	1718.2	52.1
LLaVA-Next	Yi-34b	77.8	51.6	48.8	40.4	34.8	78.9	574.0	50.7	2006.5	58.3
CogVLM2	Llama3-8B-Instruct	70.7	50.5	42.6	38.6	41.3	73.4	757.0	57.8	1869.5	58.8
InternLM-Xcomposer2	InternLM2-7B	77.6	56.2	41.4	59.5	41.0	81.2	532.0	46.7	2220.4	61.2
Cambrian	Nous-Hermes-2-Yi-34B	77.8	54.2	50.4	50.3	41.6	79.5	591.0	53.2	2049.9	61.4
InternVL-Chat-1.5	InternLM2-20B	79.7	57.1	46.8	54.7	47.4	80.6	720.0	55.4	2189.6	65.1
Ovis1.5	Gemma2-9B-It	77.3	58.1	49.7	65.6	48.2	84.5	752.0	53.8	2125.2	66.9
InternVL2	InternLM2.5-7b	79.4	61.5	51.2	58.3	45.0	83.6	794.0	54.3	2215.1	67.3
MiniCPM-V 2.6	Qwen2-7B	78.0	57.5	49.8	60.6	48.1	82.1	852.0	60.0	2268.7	68.5
Proprietary											
GPT-4V	-	65.5	50.4	59.3	48.2	39.3	71.4	678.0	49.0	1790.3	58.5
GPT-4o mini	-	76.0	54.8	60.0	52.4	46.1	77.8	785.0	66.9	2003.4	66.3
Gemini 1.5 Pro	-	73.9	59.1	60.6	57.7	45.6	79.1	754.0	64.0	2110.6	67.2
GPT-4o	-	82.8	61.6	62.8	56.5	51.7	77.4	663.0	66.5	2328.7	69.3
Claude3.5 Sonnet	-	78.5	62.2	65.9	61.6	49.9	80.2	788.0	66.0	1920.0	69.3
Ours											
VITA-1.0	Mixtral-8x7B	71.8	46.4	47.3	44.9	39.7	73.1	678.0	41.6	2097.0	57.8
VITA-1.5 (Stage 1)	Qwen2-7B	77.1	59.1	53.1	66.2	44.1	80.3	752.0	51.1	2311.0	67.1
VITA-1.5-Audio (Stage 3)	Qwen2-7B	76.7	59.9	52.1	66.2	44.9	79.3	732.0	49.6	2352.0	66.8

Evaluation – Vision-Language Capabilities

Table 3: Evaluation on Video Understanding Benchmarks. Although VITA-1.5 still lags behind models like GPT-4o and Gemini-1.5-Pro, it performs comparably to many open-source models. Note that after the training of Stages 2 (Audio Input Tuning) and 3 (Audio Output Tuning), VITA-1.5 retains almost its original visual-language capabilities in Stage 1 (Vision-Language Training).

Method	LLM	Video-MME w/o sub	Video-MME w/ sub	MVBench	TempCompass
Video-LLaVA	Vicuna-v1.5-13B	39.9	41.6	-	49.8
SliME	Llama3-8B-Instruct	45.3	47.2	-	-
LongVA	Qwen2-7B	52.6	54.3	-	57.0
VILA-1.5	Llama3-8B-Instruct	-	-	-	58.8
InternLM-XComposer-2.5	InternLM2-7B	-	-	-	62.1
LLaVA-OneVision	Qwen2-7B	58.2	61.5	56.7	64.2
InternVL-2	InternLM2.5-7b	-	-	-	66.0
MiniCPM-V-2.6	Qwen2-7B	60.9	63.7	-	66.3
Proprietary					
GPT-4o-mini	-	64.8	68.9	-	-
Gemini-1.5-Pro	-	75.0	81.3	-	67.1
GPT-4o	-	71.9	77.2	-	73.8
Ours					
VITA-1.0	Mixtral-8x7B	55.8	59.2	-	62.3
VITA-1.5 (Stage 1)	Qwen2-7B	56.8	59.5	56.8	65.5
VITA-1.5 (Stage 3)	Qwen2-7B	56.1	58.7	55.4	66.7

Evaluation – ARS Performance

Table 4: Evaluation on ASR Benchmarks. VITA-1.5 has demonstrated strong performance in both Mandarin and English ASR tasks. It outperforms specialized speech models, achieving better results in both languages.

Model	CN (CER↓)				Eng (WER↓)			
	aishell-1	test net	test meeting	dev clean	dev other	test clean	test other	
Wav2vec2-base	-	-	-	6.0	13.4	-	-	
Mini-Omini2	-	-	-	4.8	9.8	4.7	9.4	
Freeze-Omini	2.8	12.6	14.2	4.2	10.2	4.1	10.5	
Ours								
VITA-1.0	-	12.2	16.5	7.6	16.6	8.1	18.4	
VITA-1.5	2.2	8.4	10.0	3.3	7.2	3.4	7.5	

Conclusion

- VITA-1.5 is a multimodal LLM designed to integrate vision and speech through a carefully crafted three stage training strategy.
- VITA-1.5 achieves robust capabilities in both vision and speech understanding, enabling efficient speech-to-speech interactions without relying on separate ASR or TTS modules

Limitations

- Without timing benchmarks, it's unclear whether the model can meet real-time requirements in practice.
- Multi-stage process increases engineering and compute complexity.
 - Pretraining with vision-language data
 - Alignment via instruction tuning
 - Multimodal continual reinforcement tuning
- The model may suffer from forgetting previously learned text or vision tasks.
 - especially during the final reinforcement tuning phase.
 - the paper **does not evaluate the model on pure text-only benchmarks**, making it difficult to confirm whether forgetting occurred.

Thanks