

CS 2770: Multimodal Language Models

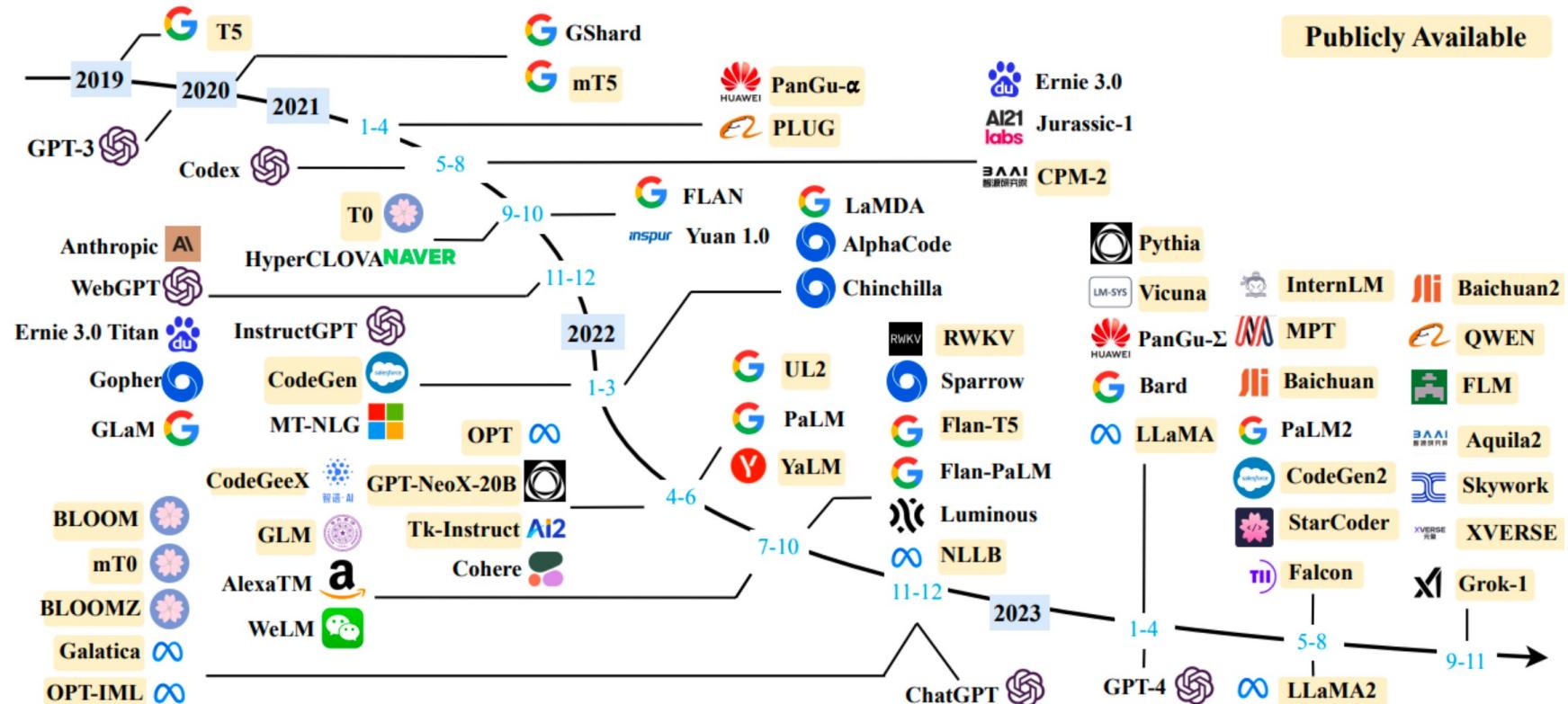
PhD. Nils Murrugarra-Llerena
nem177@pitt.edu



Plan for this lecture

1. Intro to Multimodal Large Language Models (MLLM)
2. Architectures
3. Encodings
4. Functionality
 - Contrastive Learning Models
 - Generative Models
5. Multimodal Instruction Tuning
6. What's Next?

Very Rapid Evolution of Language-based LLMs

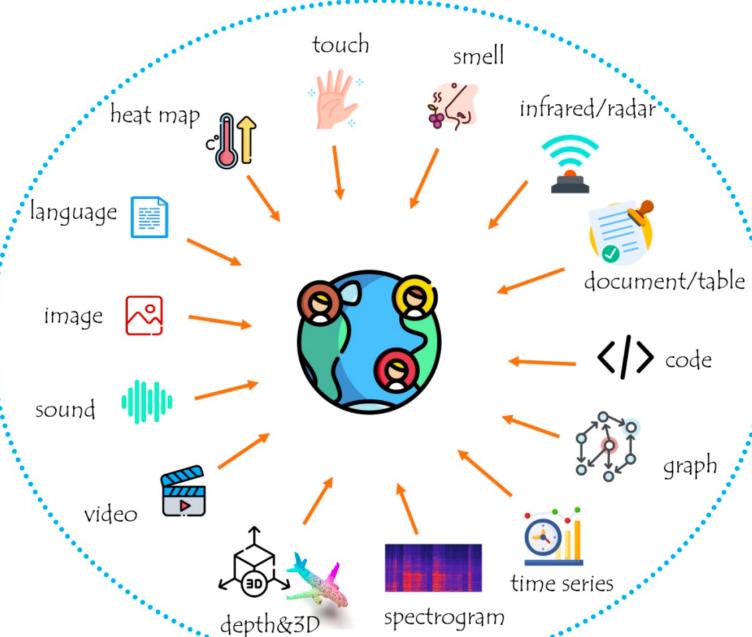


[1] A Survey of Large Language Models. <https://github.com/RUCAIBox/LLMSurvey>, 2023

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

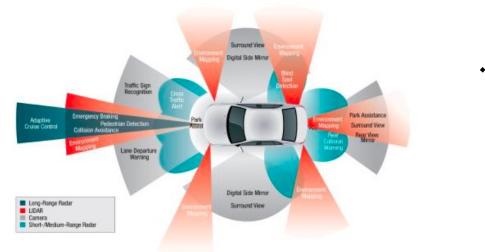
Expanding to Multimodality

This world we live in is replete with multimodal information & signals, not just language.



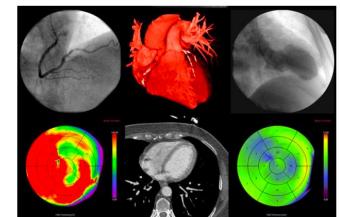
+ Autonomous Driving Systems

In this application, vehicles use a combination of visual data (cameras), spatial data (LiDAR), and auditory signals (sonar) to navigate safely.



Healthcare Diagnostics

Medical imaging tools like MRI, CT scans, and X-rays, along with patient history and verbal symptoms, are used to diagnose diseases.

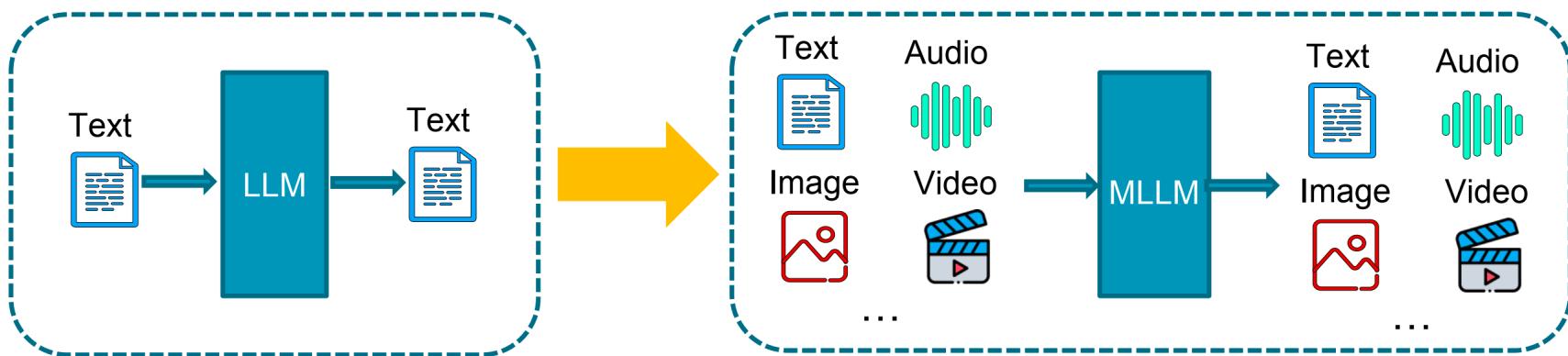


[1] A Survey of Large Language Models. <https://github.com/RUCAIBox/LLMSurvey>, 2023

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

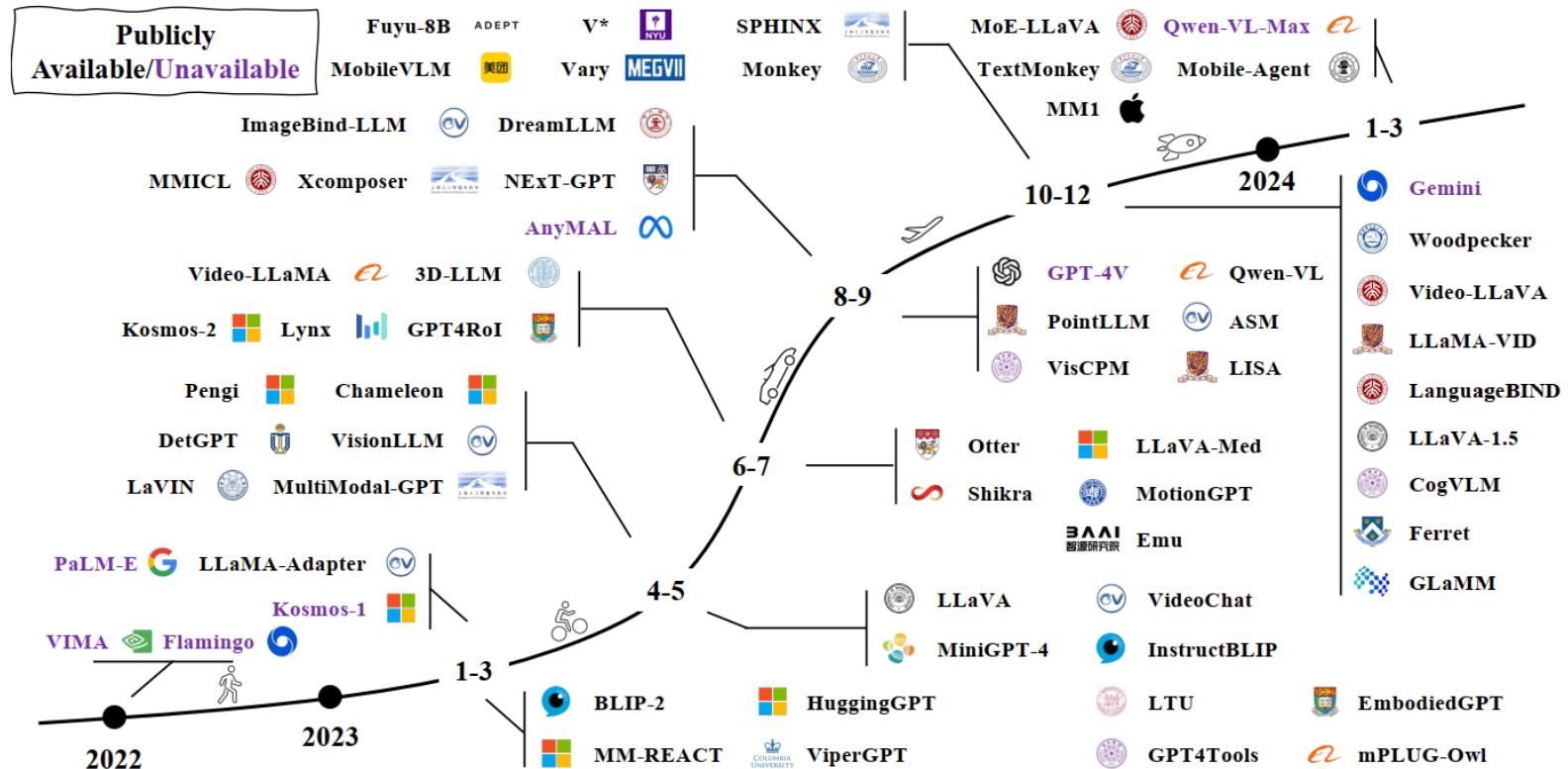
Building Multimodal LLMs

Can we transfer the success of LLMs to MLLMs, enabling LLMs to comprehend multimodal information as deeply as they understand language?



Perceiving and interacting with the world as HUMAN BEINGs do, might be the key to achieving human-level AI

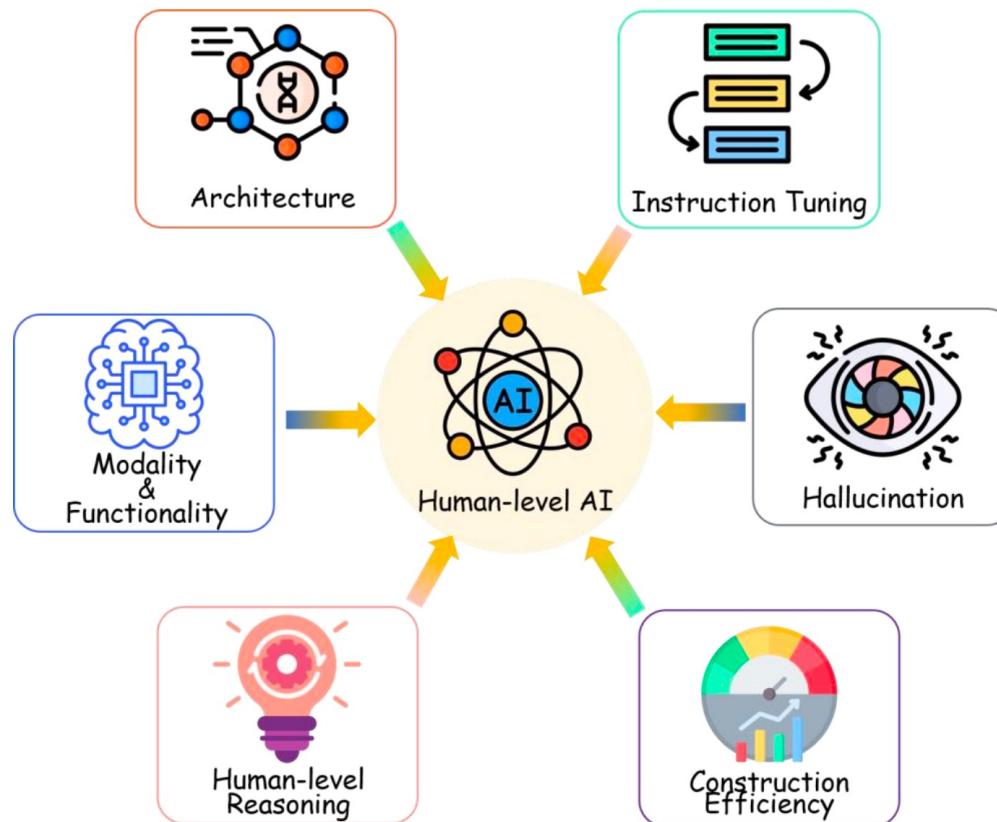
Building Multimodal LLMs



[1] A Survey on Multimodal Large Language Models. <https://github.com/BradyFU/Awesome-Multimodal-Large-Language-Models>, 2023.

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

Key Aspects for Building Powerful MLLMs



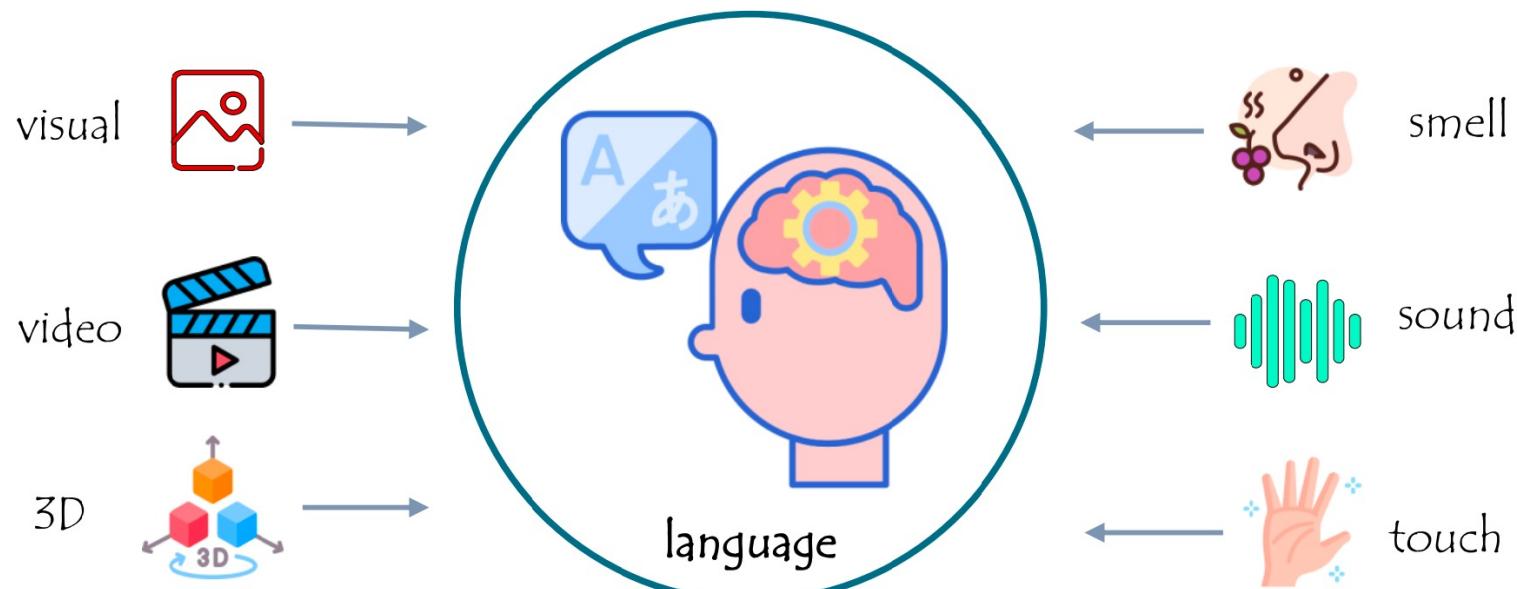
Plan for this lecture

1. Intro to Multimodal Large Language Models (MLLM)
2. Architectures
3. Encodings
4. Functionality
 - Contrastive Learning Models
 - Generative Models
5. Multimodal Instruction Tuning
6. What's Next?

Architecture MLLMs

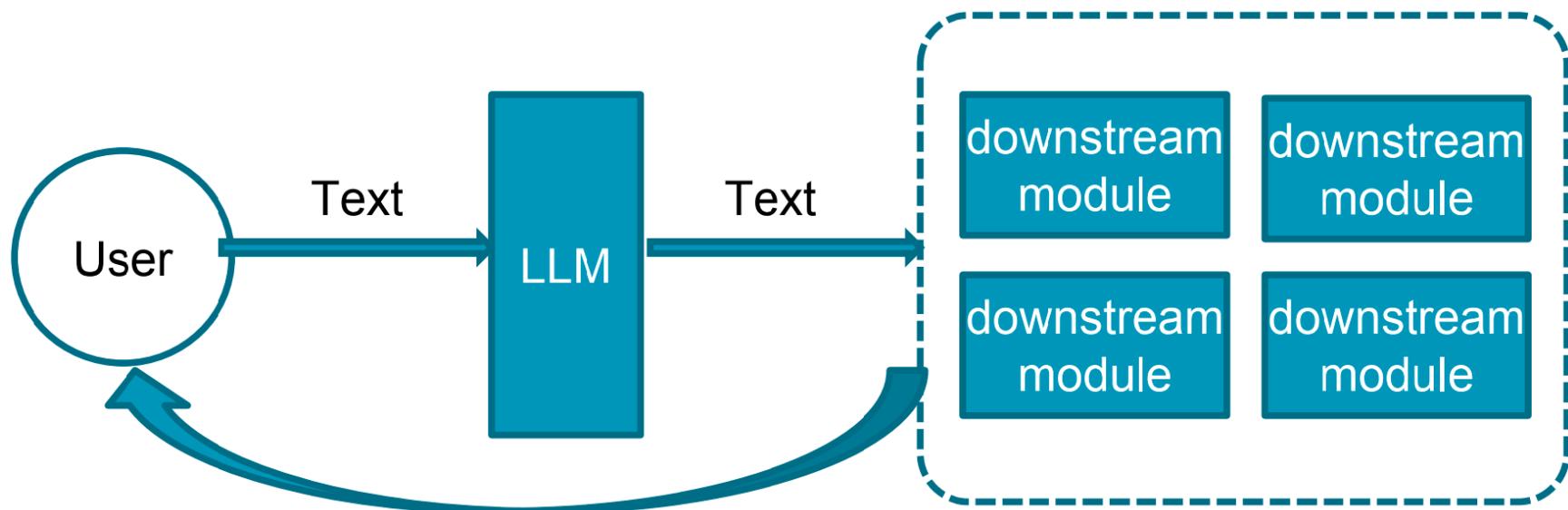
Nearly all CURRENT MLLMs are built based on language-based LLMs as the core decision-making module (i.e., the brain or central processor).

By adding additional external non-textual modality modules, LLMs are enabled with multimodal abilities.



[Arch] LLM as Discrete Scheduler/Controller

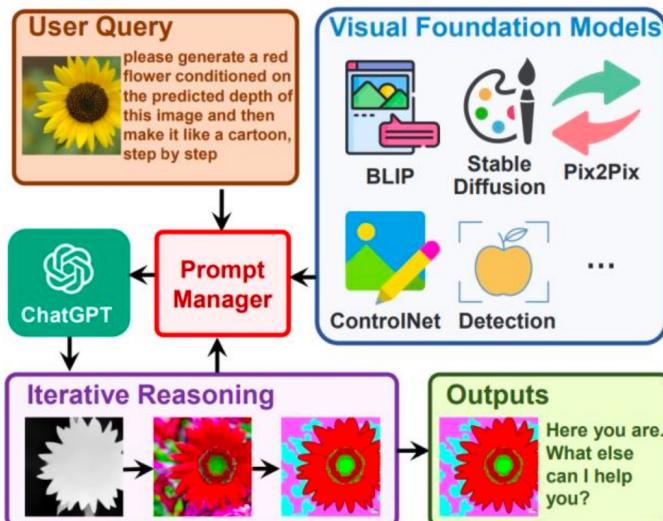
The role of the LLM is to receive textual signals and instruct textual commands to call downstream modules



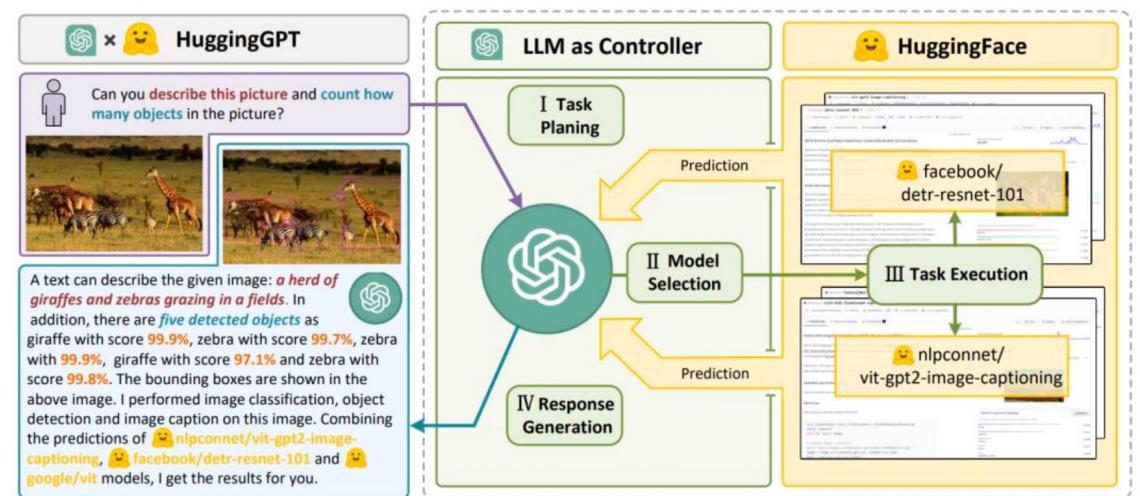
[Arch] LLM as Discrete Scheduler/Controller

- Quick to build (without training), flexible extension to many tool features
- Information loss in text medium, the bottle-neck

+ Visual-ChatGPT

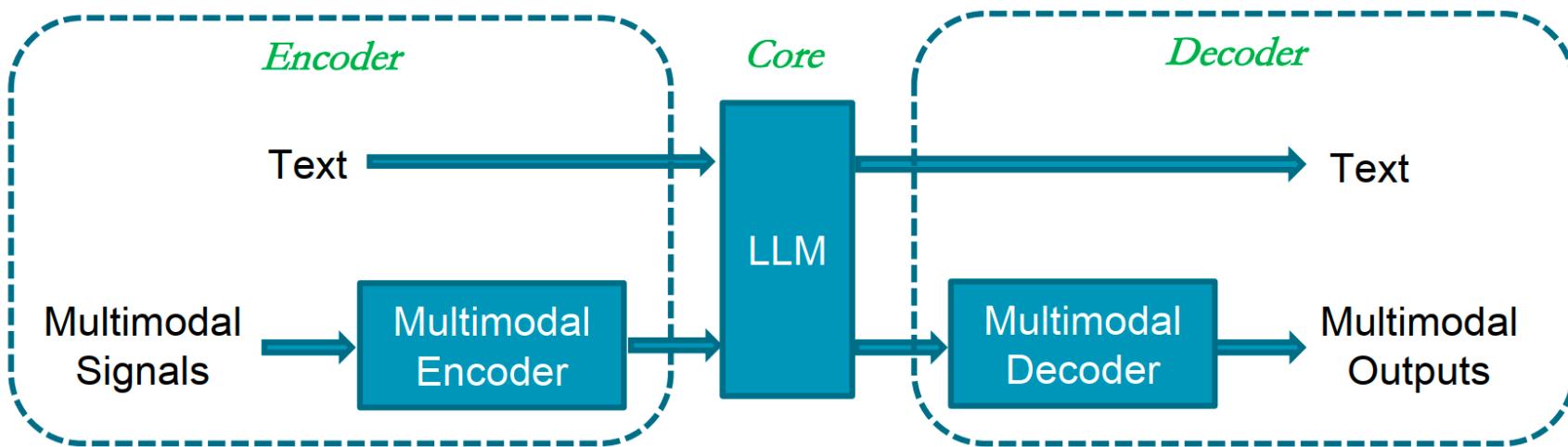


+ HuggingGPT



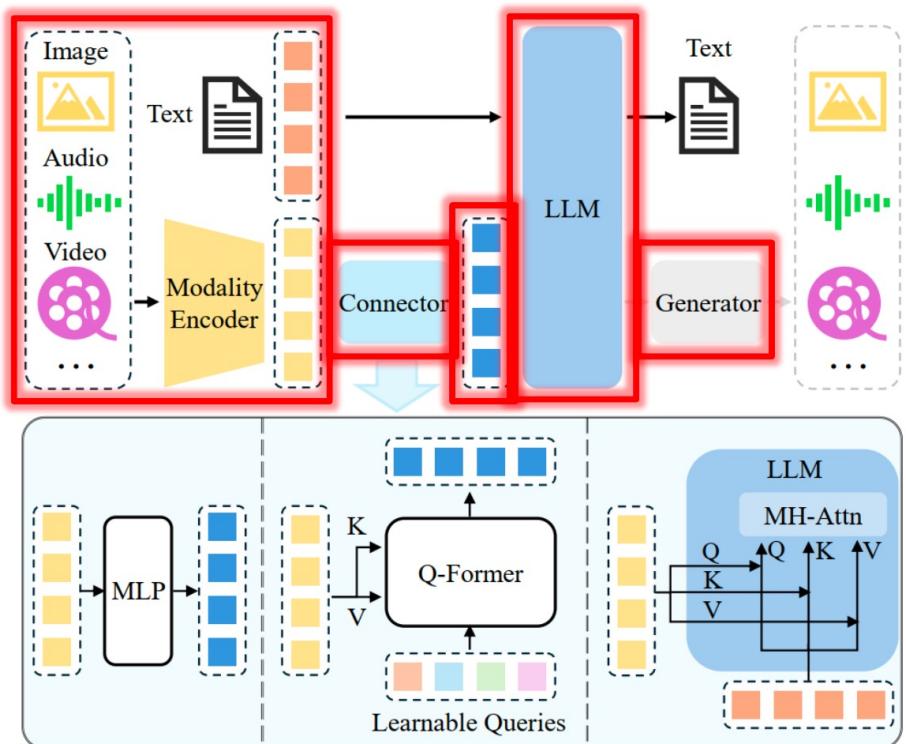
[Arch] LLM as Joint Part of System

The role of the LLM is to perceive multimodal information, and react by itself, in a structure of Encoder-LLM-Decoder.



[Arch] LLM as Joint Part of System

- > 90% MLLMs belong to this category.
- Higher upper-bound, better integrated into a unified model



[1] A Survey on Multimodal Large Language Models. <https://github.com/BradyFU/Awesome-Multimodal-LargeLanguage-Models> , 2023.

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

Plan for this lecture

1. Intro to Multimodal Large Language Models (MLLM)
2. Architectures
3. Encodings
4. Functionality
 - Contrastive Learning Models
 - Generative Models
5. Multimodal Instruction Tuning
6. What's Next?

Multimodal Encoding: Visual Encoder

[CLIP]

- CLIP-ViT is the most popular choice for vision-language models.
- SigLIP is gaining increasing popularity (smaller and stronger)

[High Resolution MLLMs]

- GPT-4V, LLaVA-NeXT, MiniCPM-V 2.0/2.5, LLaVA-UHD, mPLUGDocOwl 1.5, SPHINX, InternLM-XComposer2-4KHD, Monkey

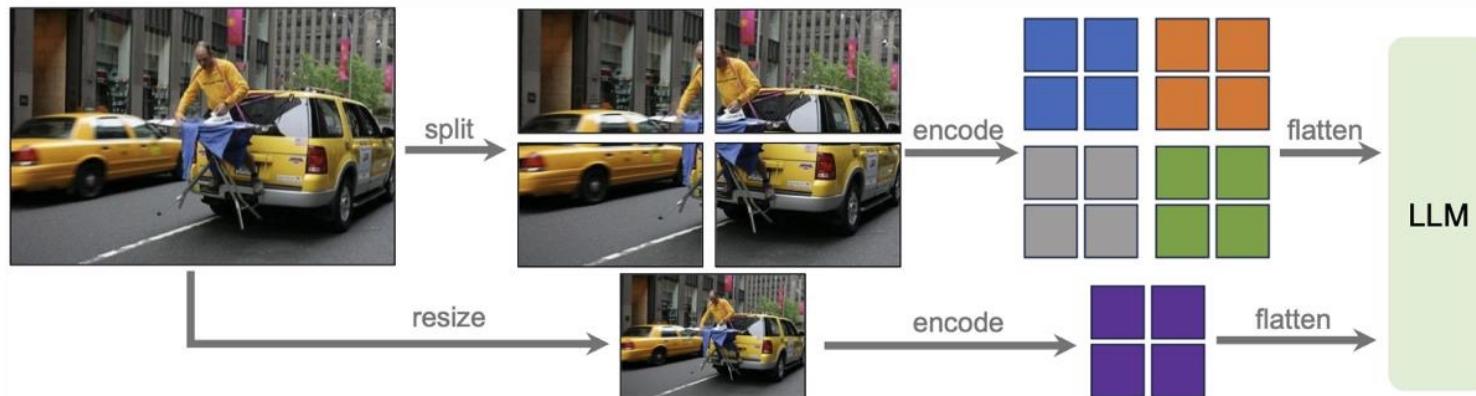
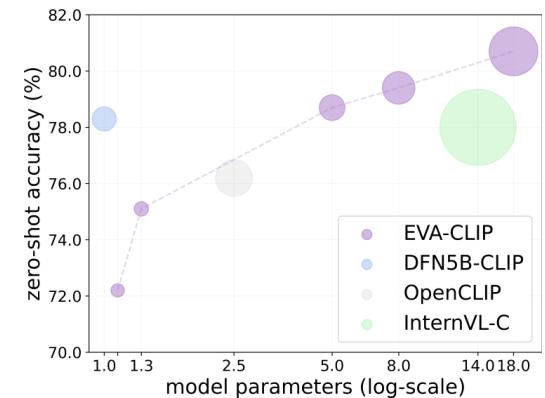


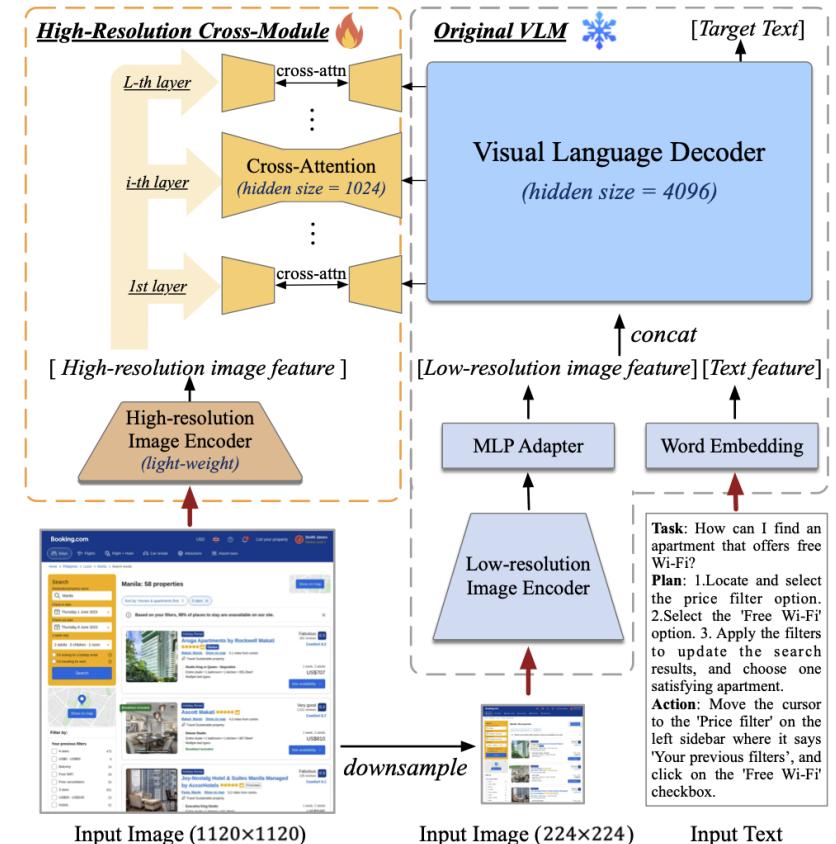
Illustration of dynamic high resolution scheme: a grid configuration of 2×2

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

Multimodal Encoding: Visual Encoder

[High Resolution MLLMs]

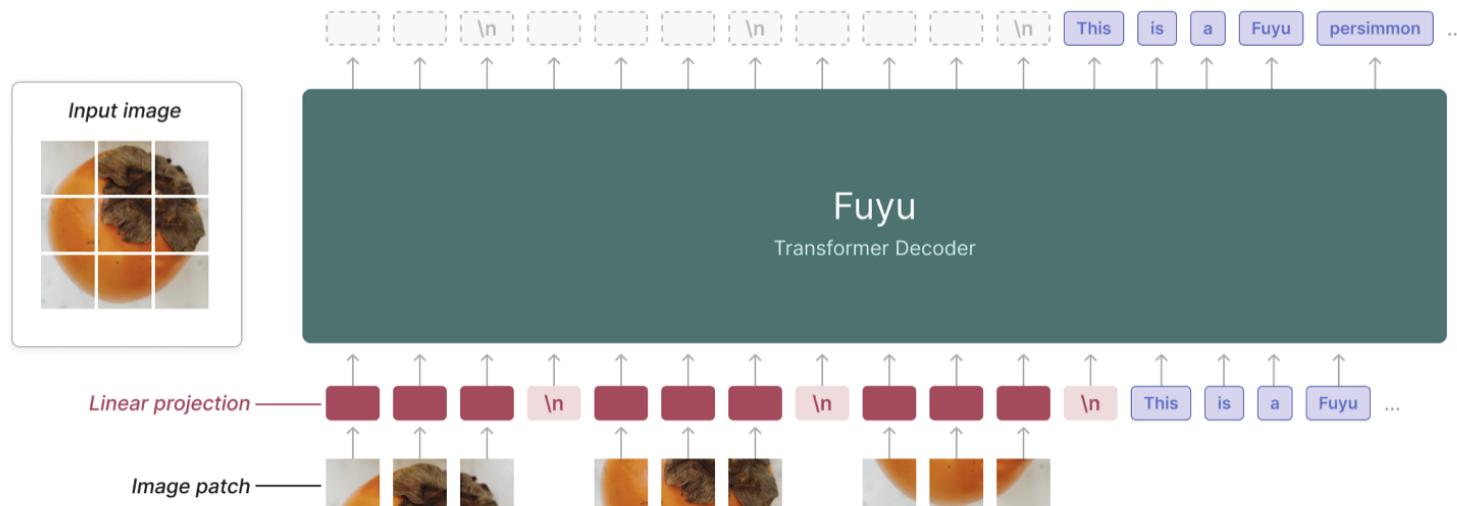
- Dual Branches Encoders
- CogAgent, Mini-Gemini, DeepSeek-VL, LLaVA-HR



Multimodal Encoding: Visual Encoder

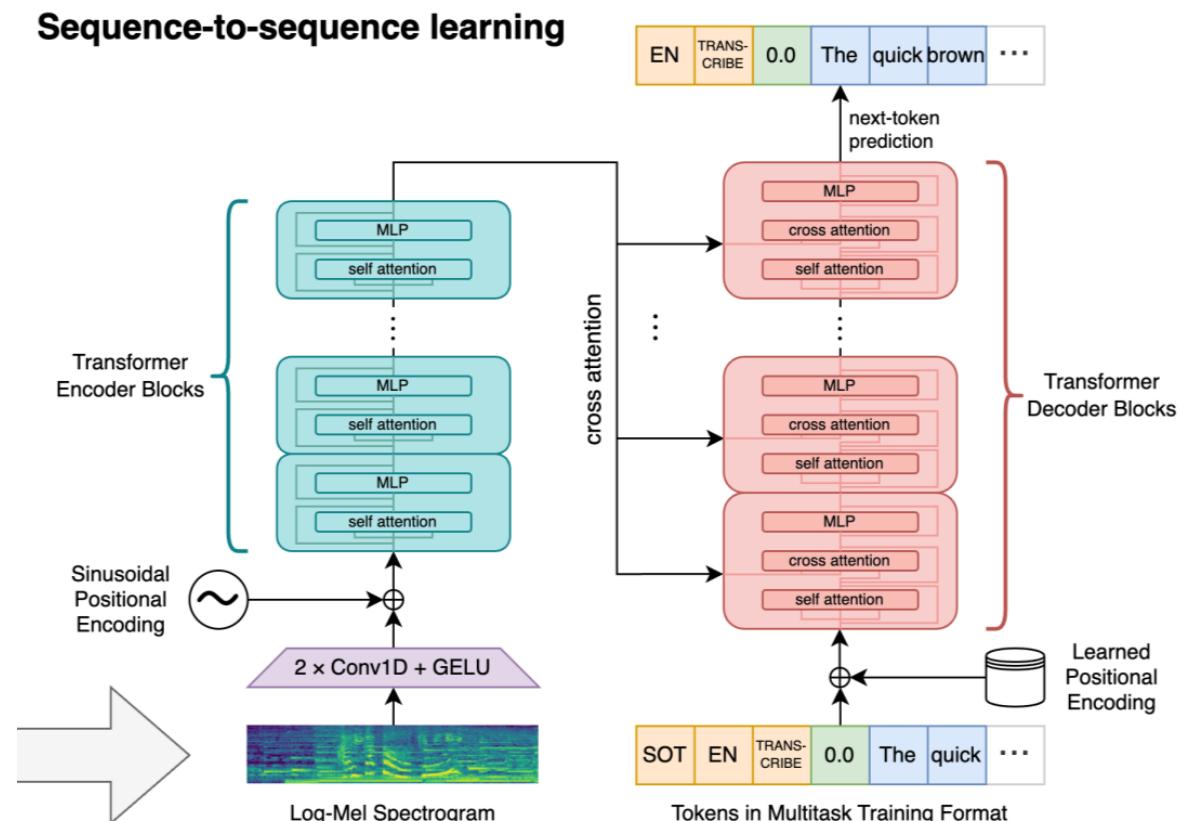
[High Resolution MLLMs]

- ViT-free: linear project pixel-patches into tokens
- Representatives: Fuyu, OtterHD
- A potential unified way for MLLMs, getting rid of ViTs
- More costly to train, produce lengthy visual tokens



Multimodal Encoding: Non-Visual Encoder

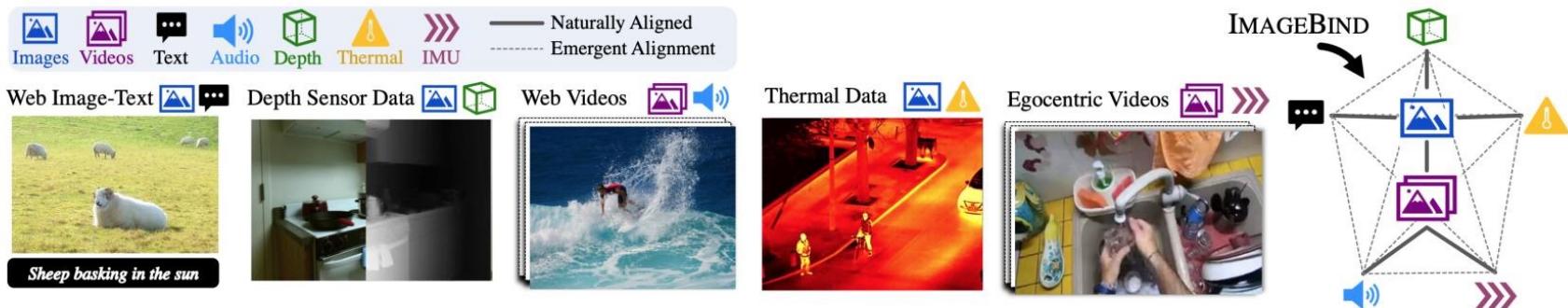
- Audio:
Whisper
AudioCLIP
HuBERT
BEATs
- 3D Point:
Point-BERT



Unified Multimodal Encoding

[ImageBind]

- Embedding all modalities into a joint representation space of Image.
- Well aligned modality representations can benefit LLM understanding



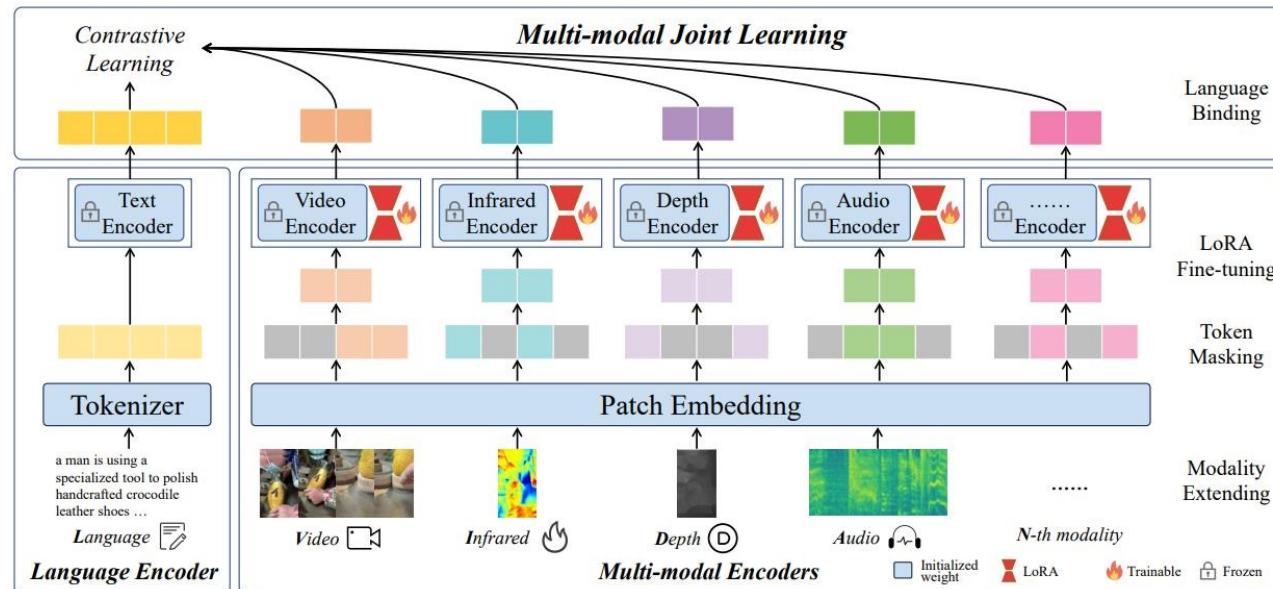
[1] ImageBind: One Embedding Space To Bind Them All. 2023

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

Unified Multimodal Encoding

[LanguageBind]

- Embedding all modalities into a joint representation space of Language.
- Well aligned modality representations can benefit LLM understanding

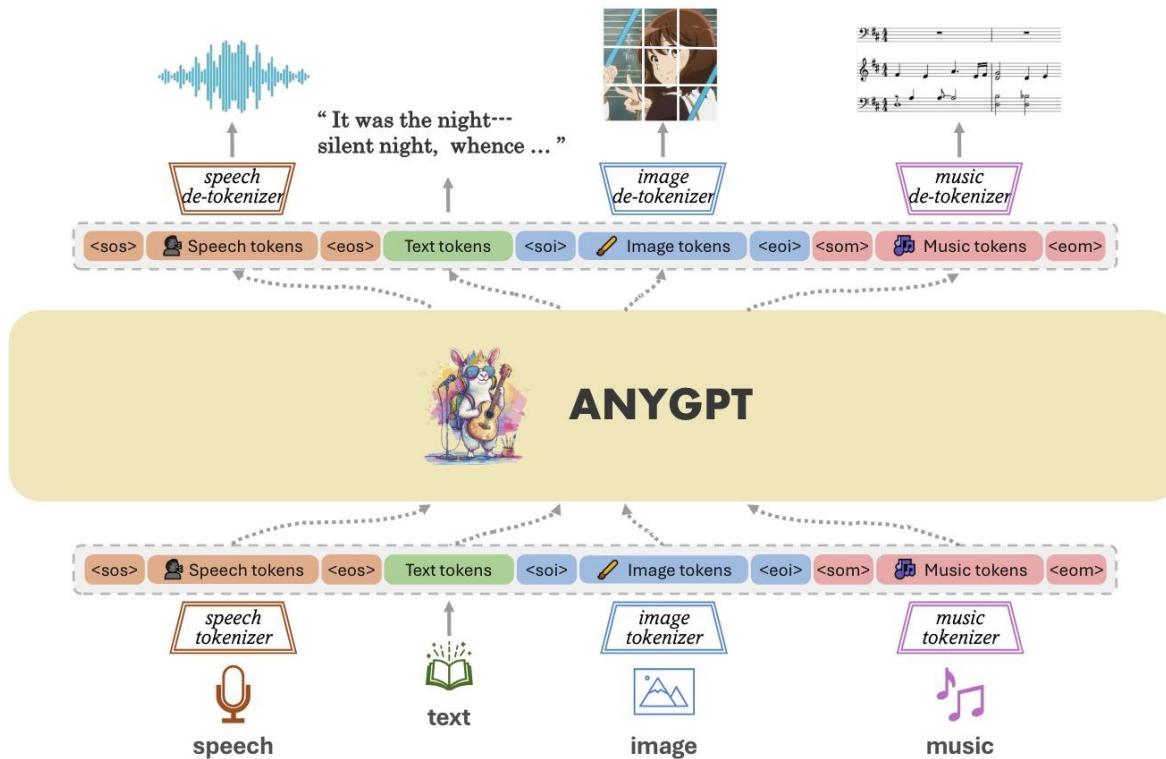


[1] LanguageBind: Extending Video-Language Pretraining to N-modality by Language-based Semantic Alignment. 2023

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

Multimodal Signal Tokenization

AnyGPT



[1] AnyGPT: Unified Multimodal LLM with Discrete Sequence Modeling. 2023

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

Multimodal Signal Tokenization

[Tokenization in codebook]

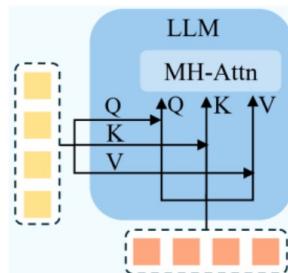
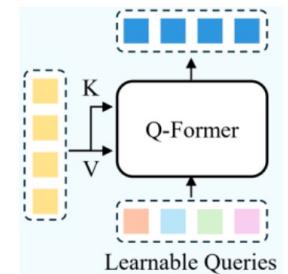
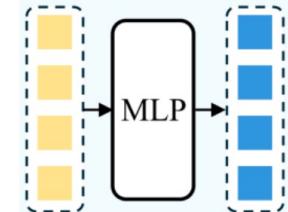
Represent multimodal signals as discrete tokens in a codebook

- **Advantages:** support unified multimodal signal understanding and generation in an auto-regressive next-token prediction framework
- More commonly used in image synthesis
 - Parti
 - Muse (parallel)
 - MaskGIT (parallel)
- Representative Multimodal LLMs
 - Gemini
 - CM3
 - VideoPoet

Methods to Connect Multimodal Representation with LLM

Projecting multimodal (e.g., image) representations into LLM semantic space

- **Q-Former:** BLIP-2, InstructBLIP, VisCPM, VisualGLM
- **Linear projection:** LLaVA, MiniGPT-4, NExT-GPT
- **Two-layer MLP:** LLaVA-1.5/NeXT, CogVLM, DeepSeek-VL, Yi-VL
- **Perceiver Resampler:** Flamingo, Qwen-VL, MiniCPM-V, LLaVA-UHD
- **C-Abstractor:** HoneyBee, MM1



Plan for this lecture

1. Intro to Multimodal Large Language Models (MLLM)
2. Architectures
3. Encodings
4. **Functionality**
 - Contrastive Learning Models
 - Generative Models
5. Multimodal Instruction Tuning
6. What's Next?

Multimodal LLM Functionality

	Modality (w/ Language)			
	Image	Video	Audio	3D
Input-side Perceiving	Flamingo, Kosmos-1, Blip2, mPLUG-Owl, Mini-GPT4, LLaVA, InstructBLIP, VPGTrans, CogVLM, Monkey, Chameleon, Otter, Qwen-VL, GPT-4v, SPHINX, Yi-VL, Fuyu, ...	VideoChat, Video-ChatGPT, Video-LLaMA, PandaGPT, MovieChat, Video-LLaVA, LLaMA-VID, Momentor, ...	AudioGPT, SpeechGPT, VIOLA, AudioPaLM, SALMONN, MU-LLaMA, ...	3D-LLM, 3D-GPT, LL3DA, SpatialVLM, PointLLM, PointBind, ...
	[Pixel-wise] GPT4RoI, LION, MiniGPT-v2, NExT-Chat, Kosmos-2, GLaMM, LISA, DetGPT, Osprey, PixelLM, ...	[Pixel-wise] PG-Video-LLaVA, Merlin, MotionEpic, ...	-	-
	Video-LLaVA, Chat-UniVi, LLaMA-VID		-	-
	Panda-GPT, Video-LLaMA, AnyMAL, Macaw-LLM, Gemini, VideoPoet, ImageBind-LLM, LLMBind, LLaMA-Adapter, ...			-
Perceiving + Generating	GILL, EMU, MiniGPT-5, DreamLLM, LLaVA-Plus, InternLM-XComposer2, SEED-LLaMA, LaVIT, Mini-Gemini, ...	GPT4Video, Video-LaVIT, VideoPoet, ...	AudioGPT, SpeechGPT, VIOLA, AudioPaLM, ...	-
	[Pixel-wise] Vitron		-	-
	NExT-GPT, Unified-IO 2, AnyGPT, CoDi-2, Modaverse, ViT-Lens, ...			-

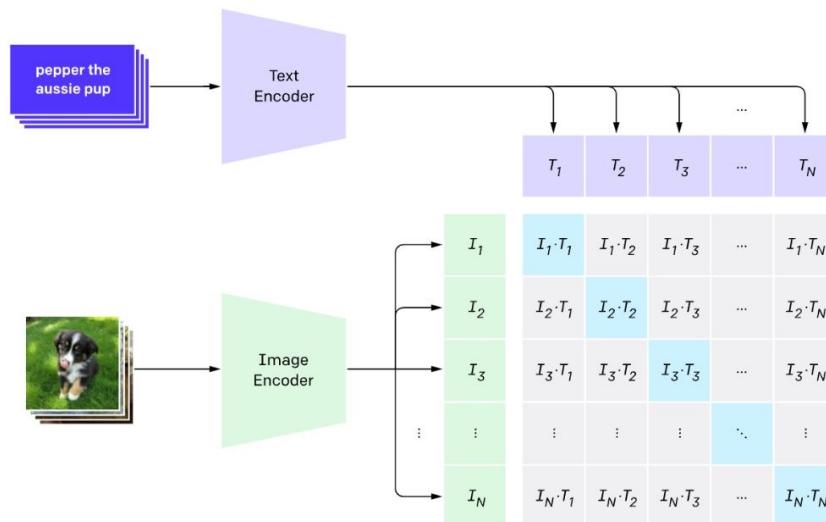
Plan for this lecture

1. Intro to Multimodal Large Language Models (MLLM)
2. Architectures
3. Encodings
4. Functionality
 - Contrastive Learning Models
 - Generative Models
5. Multimodal Instruction Tuning
6. What's Next?

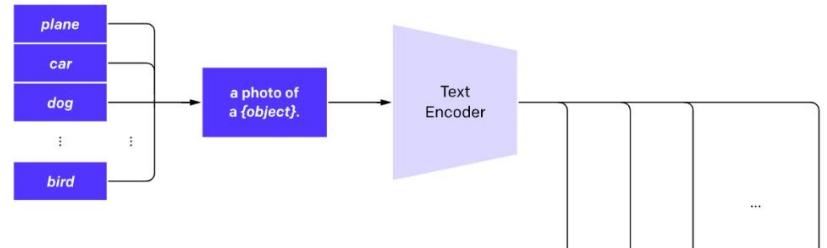
MLM Functionality: Image Perception – Contrastive Learning - CLIP (Radford et al. 2021)

Contrastive loss, but.. Transformers and *web data*!

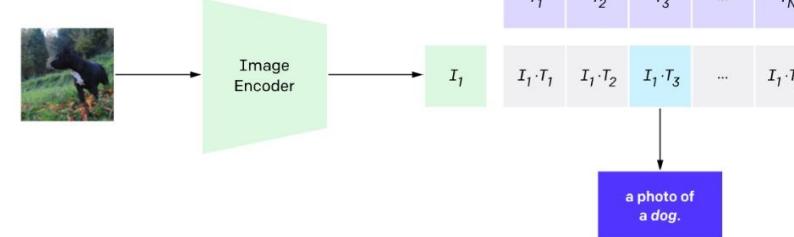
1. Contrastive pre-training



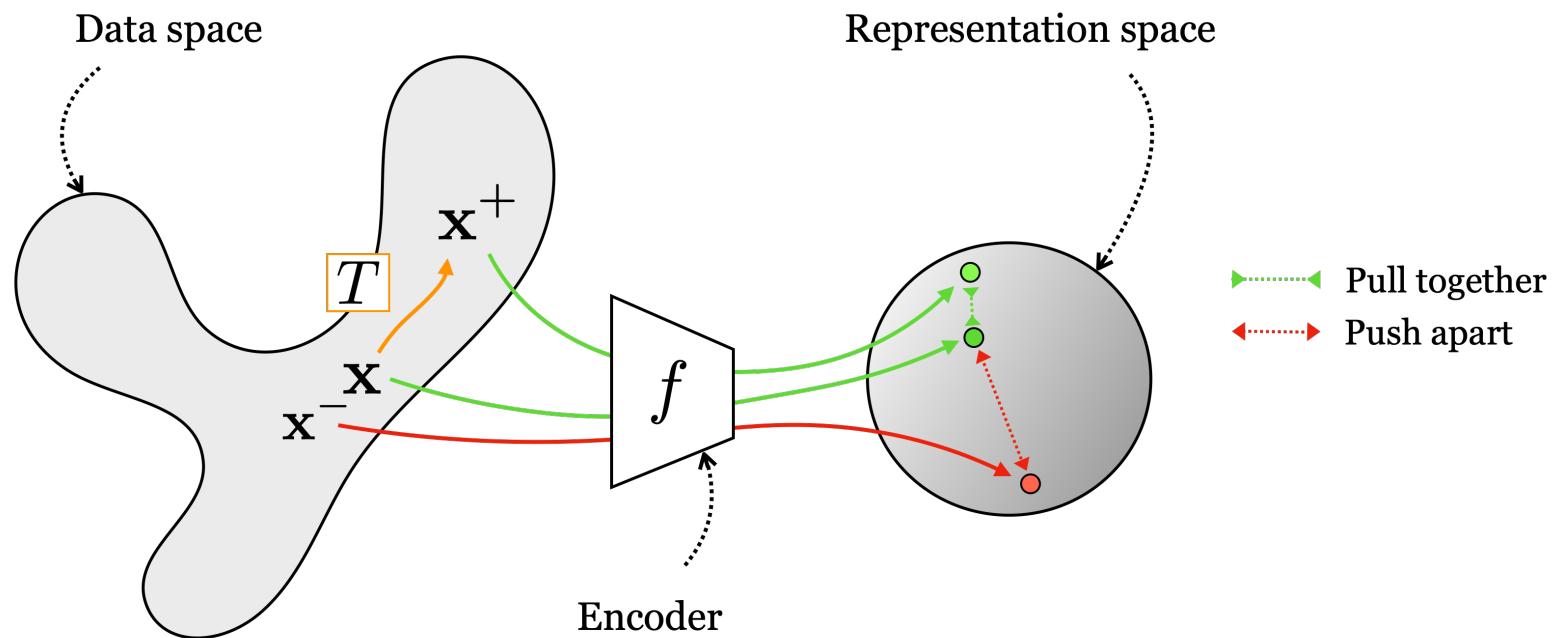
2. Create dataset classifier from label text



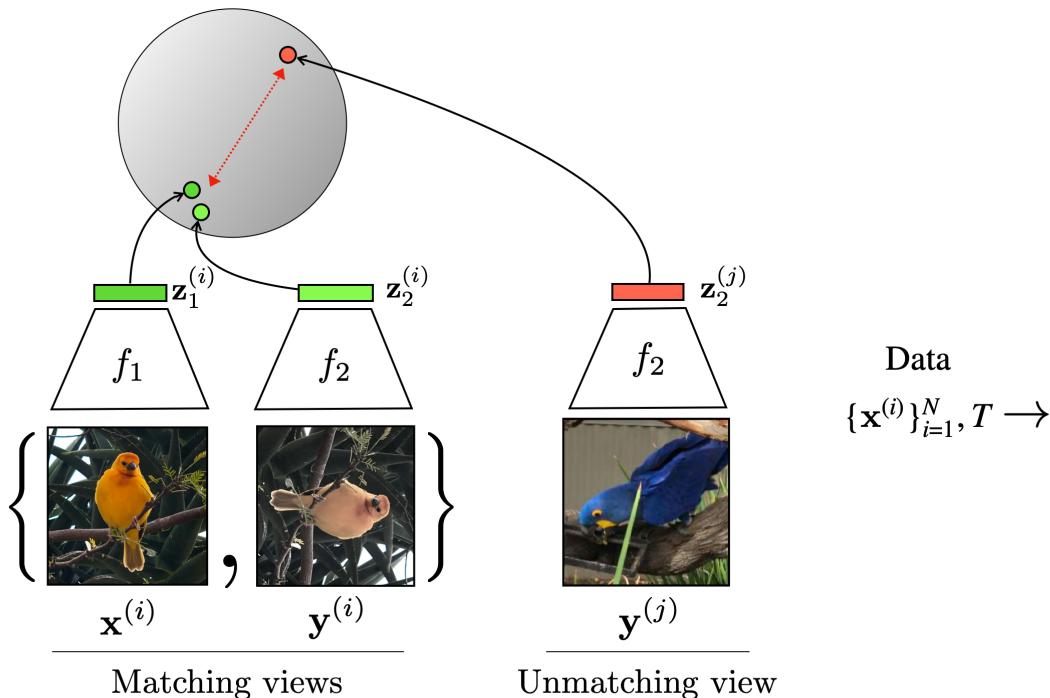
3. Use for zero-shot prediction



MLM Functionality: Image Perception – Contrastive Learning



MLM Functionality: Image Perception – Contrastive Learning



Contrastive learning (transformations)

Objective

$$\sum_{i,j} D(f(T(\mathbf{x}^{(i)})), f(\mathbf{x}^{(i)})) - D(f(\mathbf{x}^{(i)}), f(\mathbf{x}^{(j)}))$$

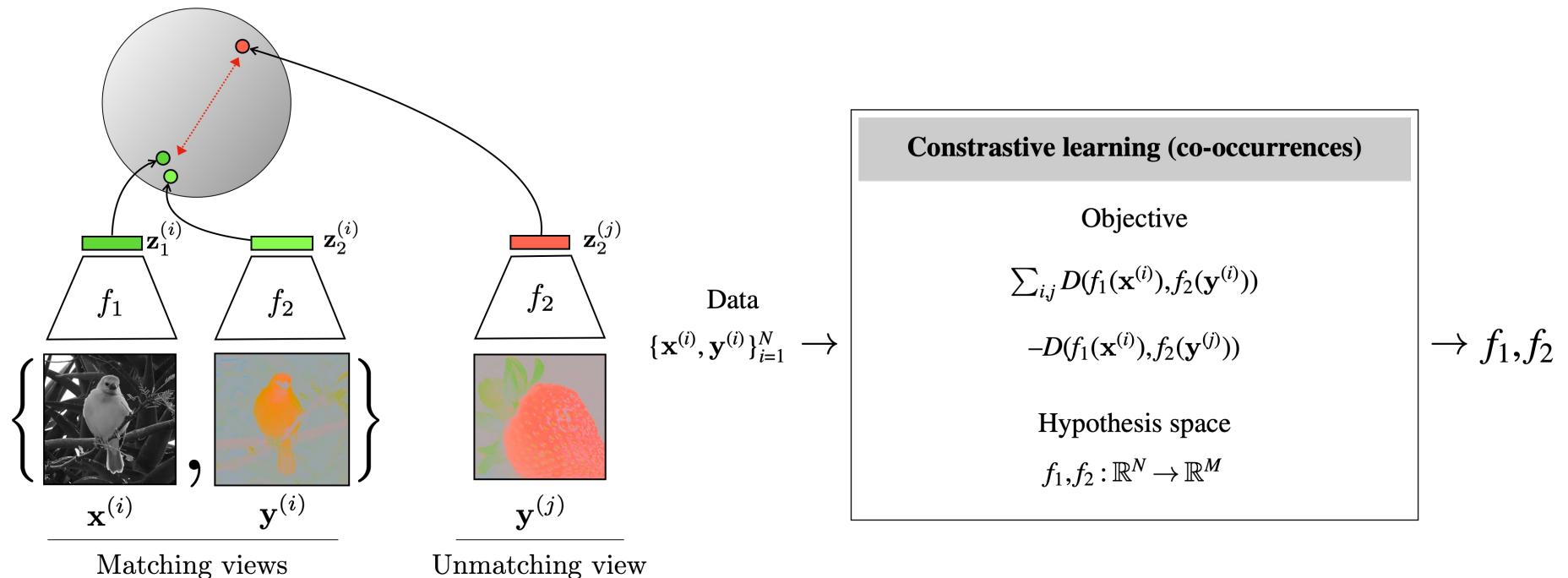
Hypothesis space

$$f : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

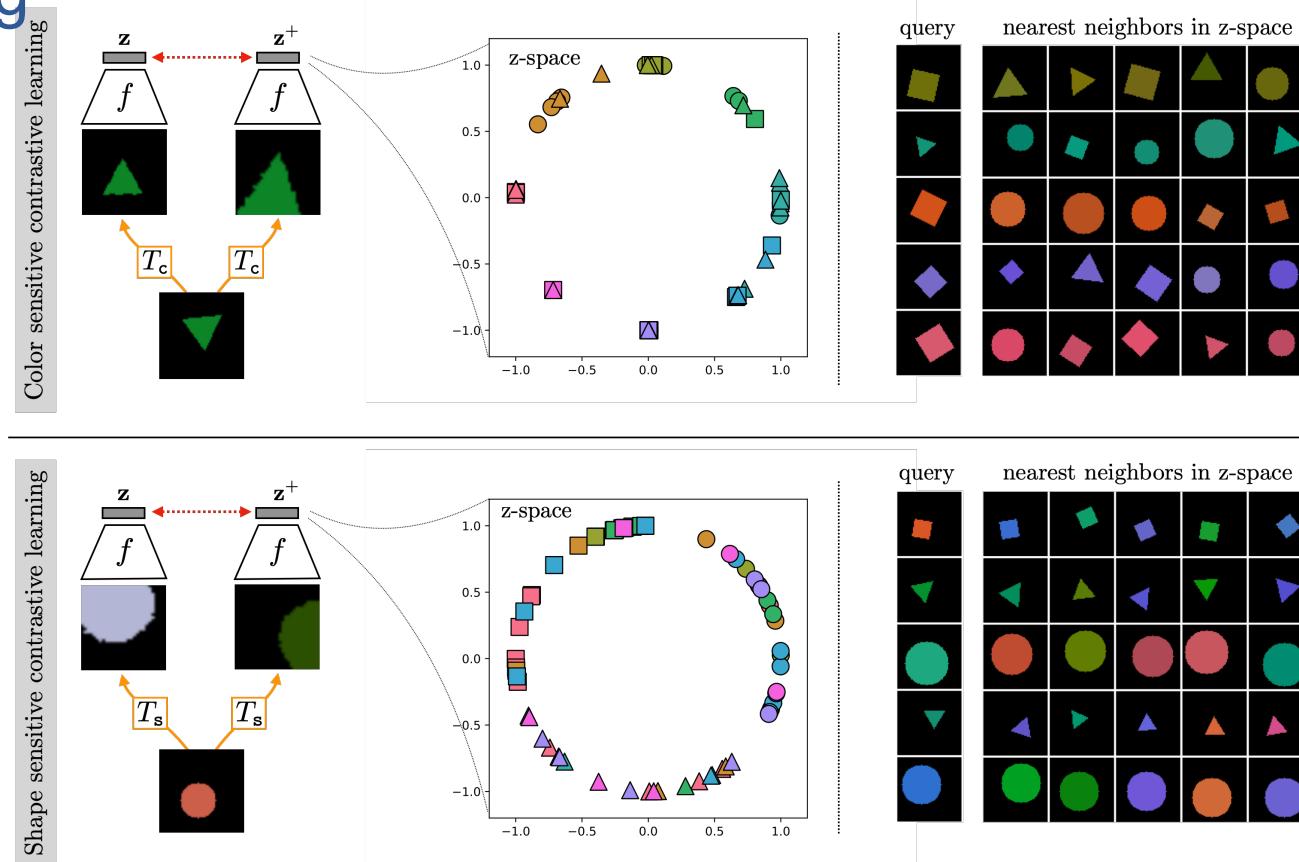
$\rightarrow f$

Data
 $\{\mathbf{x}^{(i)}\}_{i=1}^N, T \rightarrow$

MLM Functionality: Image Perception – Contrastive Learning



MLM Functionality: Image Perception – Contrastive Learning

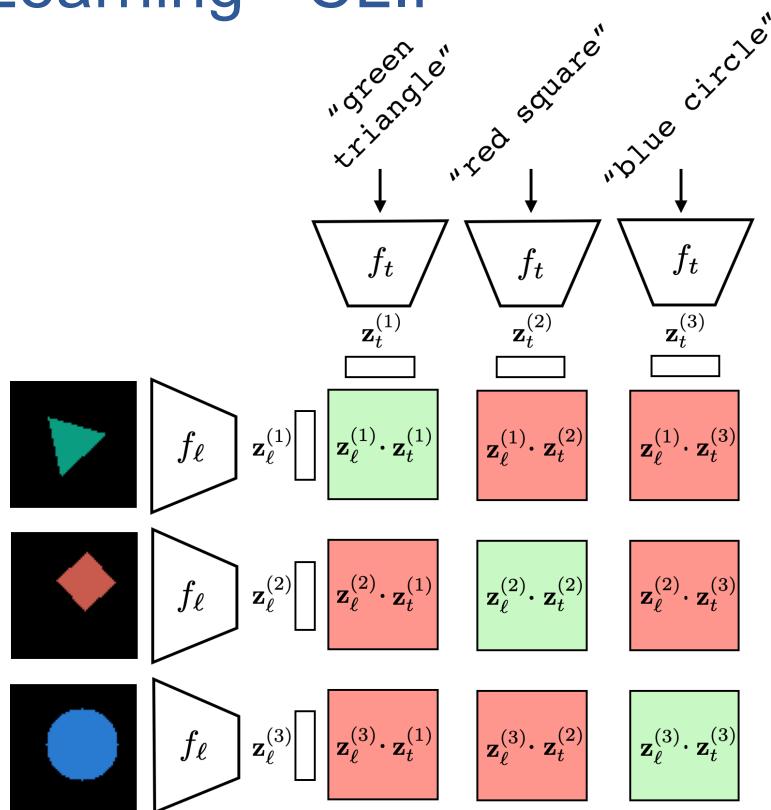


MLM Functionality: Image Perception – Contrastive Learning - CLIP

- Image classification: given an image, predict its class name
- Image captioning: given an image, predict its caption
- Contrastive learning: align image and text embeddings that describe the same thing

[Radford*, Kim* et al., ICML 2021]

MLM Functionality: Image Perception – Contrastive Learning - CLIP



- Positive pairs: image and its caption.
- Negative pairs: image and a different image's caption.
- Learn a representation in which positives are pulled together, negatives are pushed apart.

[Radford*, Kim* et al., ICML 2021]

MLM Functionality: Image Perception – Contrastive Learning - CLIP

Data
 $\{\mathbf{t}^{(i)}, \ell^{(i)}\}_{i=1}^N \rightarrow$

CLIP

Objective

$$\left(-\log \frac{e^{f_\ell(\ell)^T f_t(\mathbf{t}^+)/\tau}}{e^{f_\ell(\ell)^T f_t(\mathbf{t}^+)/\tau} + \sum_i e^{f_\ell(\ell)^T f_t(\mathbf{t}_i^-)/\tau}} + \right.$$

$$\left. -\log \frac{e^{f_t(\mathbf{t})^T f_\ell(\ell^+)/\tau}}{e^{f_t(\mathbf{t})^T f_\ell(\ell^+)/\tau} + \sum_i e^{f_t(\mathbf{t})^T f_\ell(\ell_i^-)/\tau}} \right) / 2$$

$\rightarrow f_\ell, f_t$

Hypothesis space

$$f_\ell : \mathbb{R}^{3 \times N \times M} \rightarrow \mathbb{S}^{d_z}$$

$$f_t : \mathbb{R}^{T \times d_t} \rightarrow \mathbb{S}^{d_z}$$

MLM Functionality: Image Perception – Improving CLIP with Descriptions

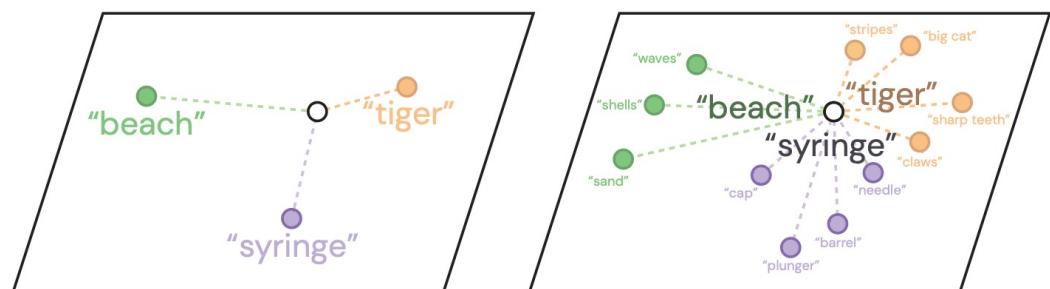
- Instead of directly querying class category names, can we use descriptions to improve the classification for vision language models?
 - Can we use language modality as the internal representation for vision task to enhance interpretability?



Our top prediction: Hen
and we say that because...

Average

- two legs
 - red, brown, or white feathers
 - a small body
 - a small head
 - two wings
 - a tail
 - a beak
 - a chicken



MLLM Functionality: Image Perception – Visual Classification via Description from Large Language Models ([Menon and Vondrick, ICLR 2023](#))

$$s(c, x) = \frac{1}{|D(c)|} \sum_{d \in D(c)} \phi(d, x)$$

$s(c, x)$: computed as the addition of all the descriptors pertains to image

- x: image
- d: descriptor
- D(c): descriptors for class c
- phi: dot product using CLIP

```
for i, (k, v) in enumerate(description_encodings.items()): # You can also vectorize this; it wasn't much

    dot_product_matrix = image_encodings @ v.T

    image_description_similarity[i] = dot_product_matrix
    image_description_similarity_cumulative[i] = aggregate_similarity(image_description_similarity[i])
```

MLLM Functionality: Image Perception – Visual Classification via Description from Large Language Models ([Menon and Vondrick, ICLR 2023](#))

Generating Descriptors from Large Language Models (LLMs)

Q: What are useful features for distinguishing a {category name} in a photo?

A: There are several useful visual features to tell there is a {category name} in a photo:

 → Adding “-” help elicit LLMs to output in a bulleted list

Prompt
Structure

Q: What are useful visual features for distinguishing a lemur in a photo?

A: There are several useful visual features to tell there is a lemur in a photo:

- four-limbed primate
- black, grey, white, brown, or red-brown
- wet and hairless nose with curved nostrils
- long tail
- large eyes
- furry bodies
- clawed hands and feet

MLLM Functionality: Image Perception – Visual Classification via Description from Large Language Models ([Menon and Vondrick, ICLR 2023](#))

Classification by Description

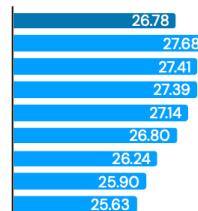
$$\arg \max_{c \in C} s(c, x)$$

$$s(c, x) = \frac{1}{|D(c)|} \sum_{d \in D(c)} \phi(d, x) \rightarrow \text{Computed by CLIP similarity}$$



Our top prediction: **Hen**
and we say that because...

- Average
- ↳ two legs
- ↳ red, brown, or white feathers
- ↳ a small body
- ↳ a small head
- ↳ two wings
- ↳ a tail
- ↳ a beak
- ↳ a chicken



CLIP's top prediction: **Dalmatian**
but we don't say that because...

MLLM Functionality: Image Perception – Visual Classification via Description from Large Language Models ([Menon and Vondrick, ICLR 2023](#))

Generating Descriptors from Large Language Models (LLMs)

- Descriptors are generated by prompting GPT3
 - Category Specific
 - Or general description

School bus



Shoe store



Volcano



Barber shop



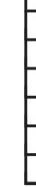
Cheeseburger



Violin



Pirate ship



MLLM Functionality: Image Perception – Visual Classification via Description from Large Language Models ([Menon and Vondrick, ICLR 2023](#))

Classification by Description (Results)

Architecture for ϕ	ImageNet			ImageNetV2			CUB			EuroSAT		
	Ours	CLIP	Δ	Ours	CLIP	Δ	Ours	CLIP	Δ	Ours	CLIP	Δ
ViT-B/32	62.97	58.46	4.51	55.52	51.90	3.62	52.57	51.95	0.62	48.94	43.84	5.10
ViT-B/16	68.03	64.05	3.98	61.54	57.88	3.66	57.75	56.35	1.40	48.82	43.36	5.46
ViT-L/14	75.00	71.58	3.42	69.3	65.33	3.97	63.46	63.08	0.38	48.66	41.48	7.18
ViT-L/14@336px	76.16	72.97	3.19	70.32	66.58	3.74	65.257	63.41	1.847	48.74	44.80	3.94
Places365			Food101			Oxford Pets			Describable Textures			
ViT-B/32	39.90	37.37	2.52	83.63	79.31	4.32	83.46	79.94	3.52	44.26	41.38	2.87
ViT-B/16	40.34	38.27	2.07	88.50	85.61	2.90	86.92	81.88	5.04	45.59	43.72	1.86
ViT-L/14	40.55	39.00	1.55	92.44	91.79	0.65	92.23	88.25	3.98	54.36	51.33	3.03
ViT-L/14@336px	41.18	39.58	1.59	93.26	92.23	1.03	91.69	88.20	3.49	54.95	52.39	2.55

Model Variants. We base ViT configurations on those used for BERT (Devlin et al., 2019), as summarized in Table 1. The “Base” and “Large” models are directly adopted from BERT and we add the larger “Huge” model. In what follows we use brief notation to indicate the model size and the input patch size: for instance, ViT-L/16 means the “Large” variant with 16×16 input patch size. Note that the Transformer’s sequence length is inversely proportional to the square of the patch size, thus models with smaller patch size are computationally more expensive.

Dosovitskiy et al., ICLR 2021

MLLM Functionality: Image Perception – Visual Classification via Description from Large Language Models ([Menon and Vondrick, ICLR 2023](#))

Capability in acquiring and utilizing novel information

- Add two new categories to the validation dataset of ImageNet

Query	Descriptors	Recall: 100%			Recall: 10%
		Ours	CLIP	Recall: 0%	
Ever Given	a large container ship red, white, and green the name "EVER GIVEN" written on the side of the ship a stack of containers on the deck of the ship				
Wordle game	a grid of letter tiles different colors for different letters				

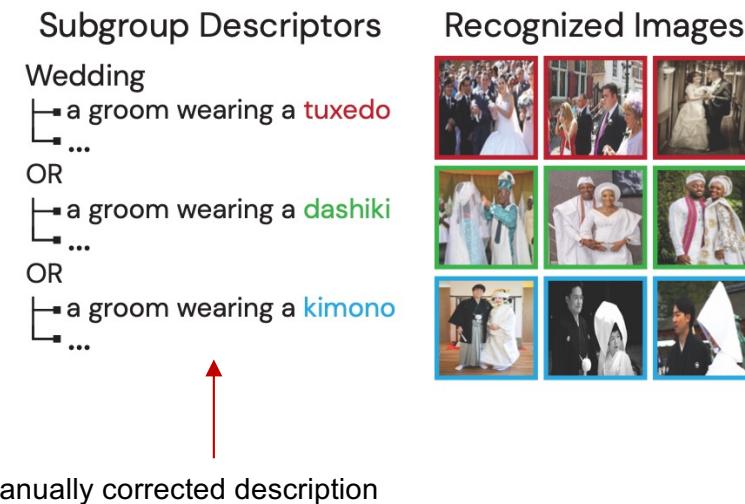
↑

Descriptors generated by GPT 3

MLLM Functionality: Image Perception – Visual Classification via Description from Large Language Models ([Menon and Vondrick, ICLR 2023](#))

Correcting failures induced by bias

- Both foundational models (CLIP and GPT 3) have bias for certain categories ---- e.g. “Wedding”



Sub-group	Ours	CLIP
Western African	100%	40%
Chinese	100%	20%
Japanese	100%	0%
North Indian	100%	60%

MLLM Functionality: Image Perception – Visual Classification via Description from Large Language Models ([Menon and Vondrick, ICLR 2023](#))

Classification by Description (Results)

Analyzing the failure modes

- Failure in descriptor creation



Jackfruit, which (has/is/etc)

- large, round fruit
- green or yellow skin
- white flesh with black seeds
- sweet and sticky taste
- strong smell

Not visual descriptors

Vespa, which (has/is/etc)

- a type of wasp
- black and yellow stripes
- a stinger
- two pairs of wings
- six legs
- a narrow waist

From GPT 3



Hair spray, which (has/is/etc)

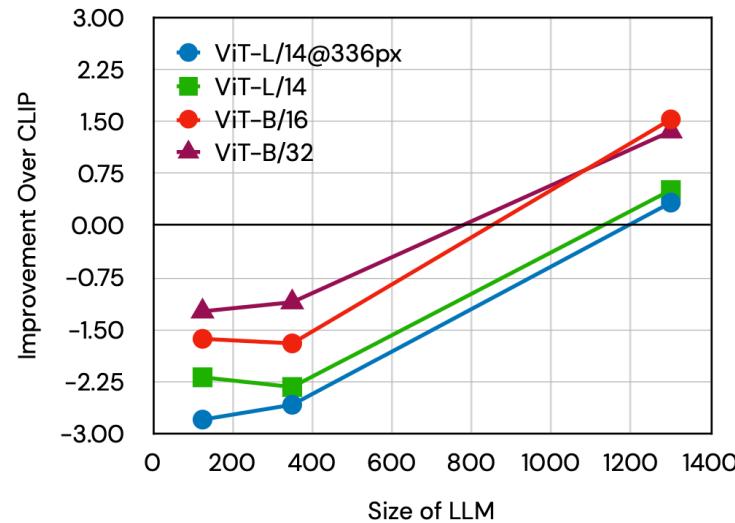
- aerosable product
- aerosable product
- aerosable product

MLLM Functionality: Image Perception – Visual Classification via Description from Large Language Models ([Menon and Vondrick, ICLR 2023](#))

Classification by Description (Results)

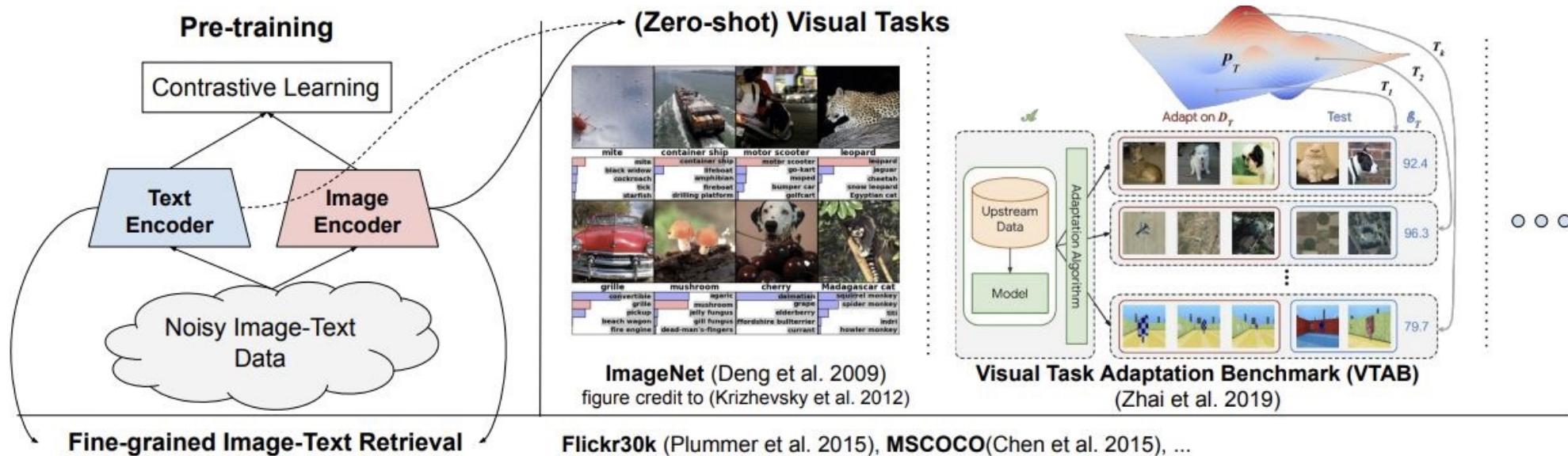
Influences of language model choices

- Small LLMs degrade the performance



MLM Functionality: Image Perception – Contrastive Learning - ALIGN (Jia et al., 2021)

Same idea of CLIP, but EVEN MORE data (JFT at 1.8B image-text pairs vs CLIP's 300m).



MLM Functionality: Image Perception

FLAVA (Singh et al., 2021)

Holistic approach to multimodality.

One foundation model spanning V&L, CV and NLP.

Jointly pretrained on:

- unimodal text data (CCNews + BookCorpus)
- unimodal image data (ImageNet)
- public paired image-text data (70M)

All data/models are publicly released.



MLLM Functionality: Image Perception - FLAVA

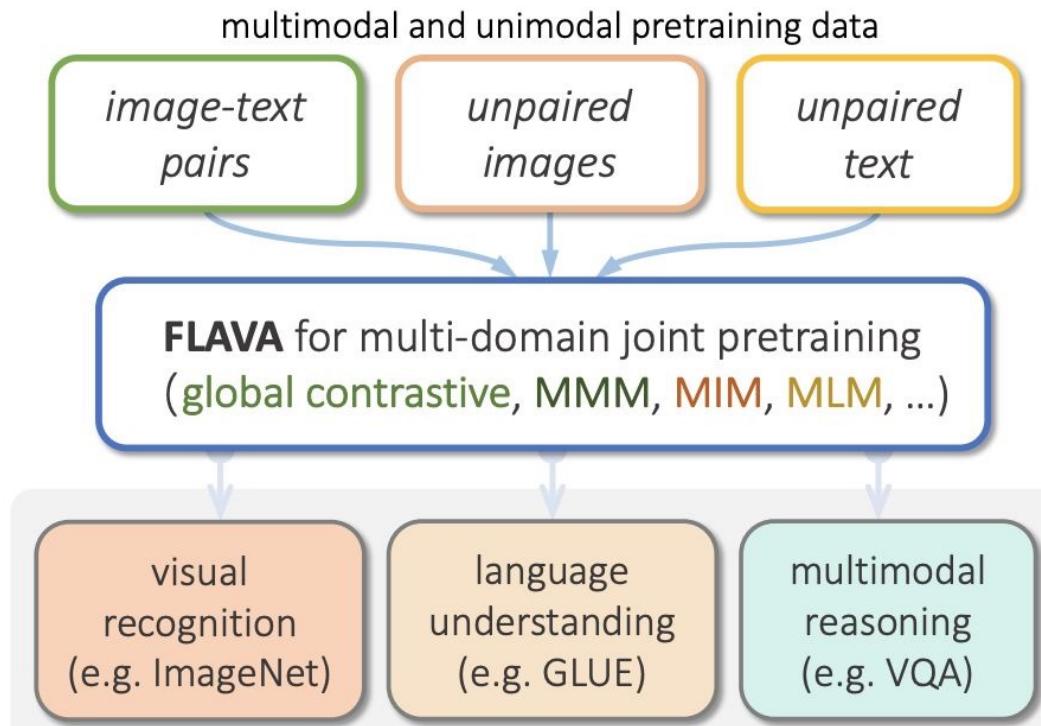
The PMD dataset

- 70M image-text pairs from public sources

COCO	Visual Genome	SBU captions	Localized narratives	WIT	RedCaps	CC12M	YFCC filtered
							
A close up view of a pizza sitting on a table with a soda in the back.	a lenovo laptop rebooting	Front view of basket 13, from the sidewalk in front of the basket.	The woman is touching a utensil in front of her on the grill stand.	Typocerus balteatus, Subfamily: Flower Longhorns	Deigdoh falls in india	Jumping girl in a green summer dress stock illustration	In the kitchen at the Muse Nissim de Camondo

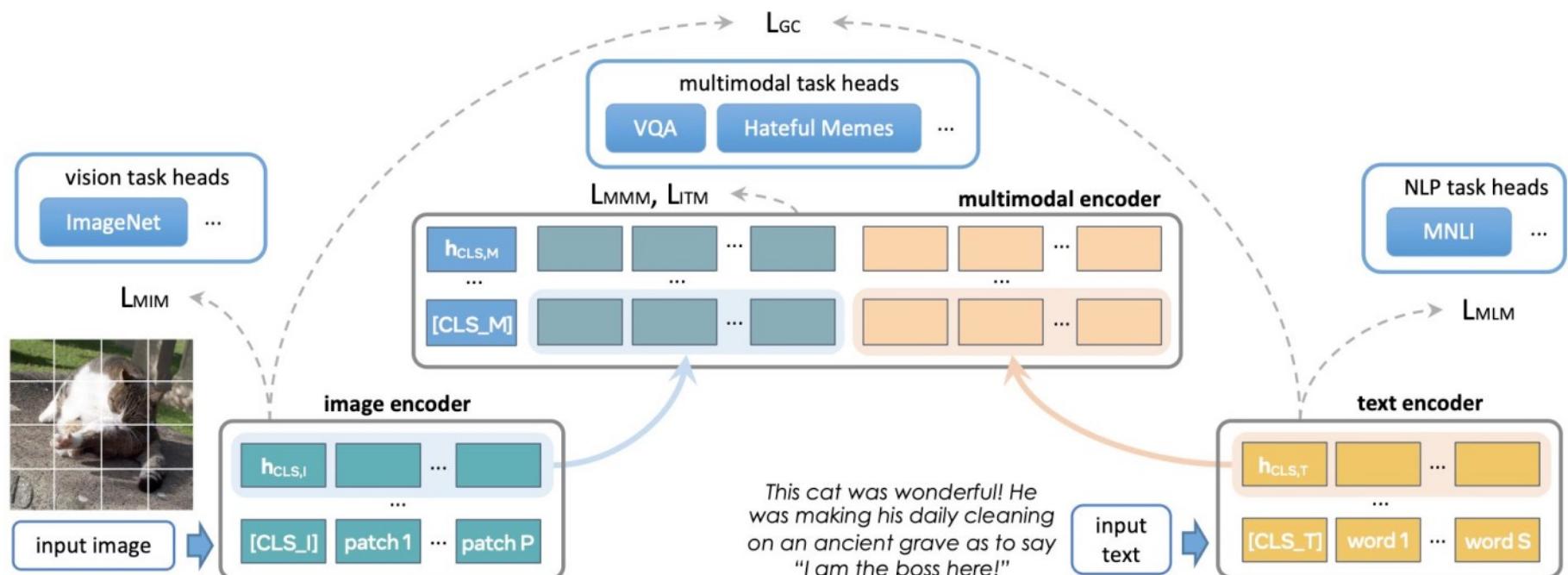
MLM Functionality: Image Perception - FLAVA

Problems to Solve



MLM Functionality: Image Perception - FLAVA

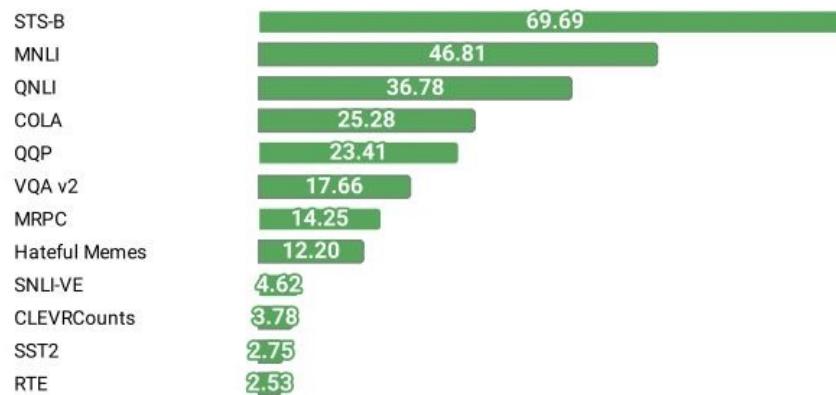
How does FLAVA work?



MLM Functionality: Image Perception - FLAVA

How does FLAVA work?

On average, over 35 tasks,
FLAVA obtains impressive
performance



CIFAR10	-2.06
Food101	-3.05
CIFAR100	-3.42
FGVCAircraft	-4.08
OxfordPets	-6.83
FER 2013	-7.24
GTSRB	-9.14
COCO-Ret	-9.74
Flickr30k-Ret	-14.50
St. Cars	-15.05
SST	-17.57

MLM Functionality: Image Perception - FLAVA

How does FLAVA work?

On average, over 35 tasks,
FLAVA obtains impressive
performance

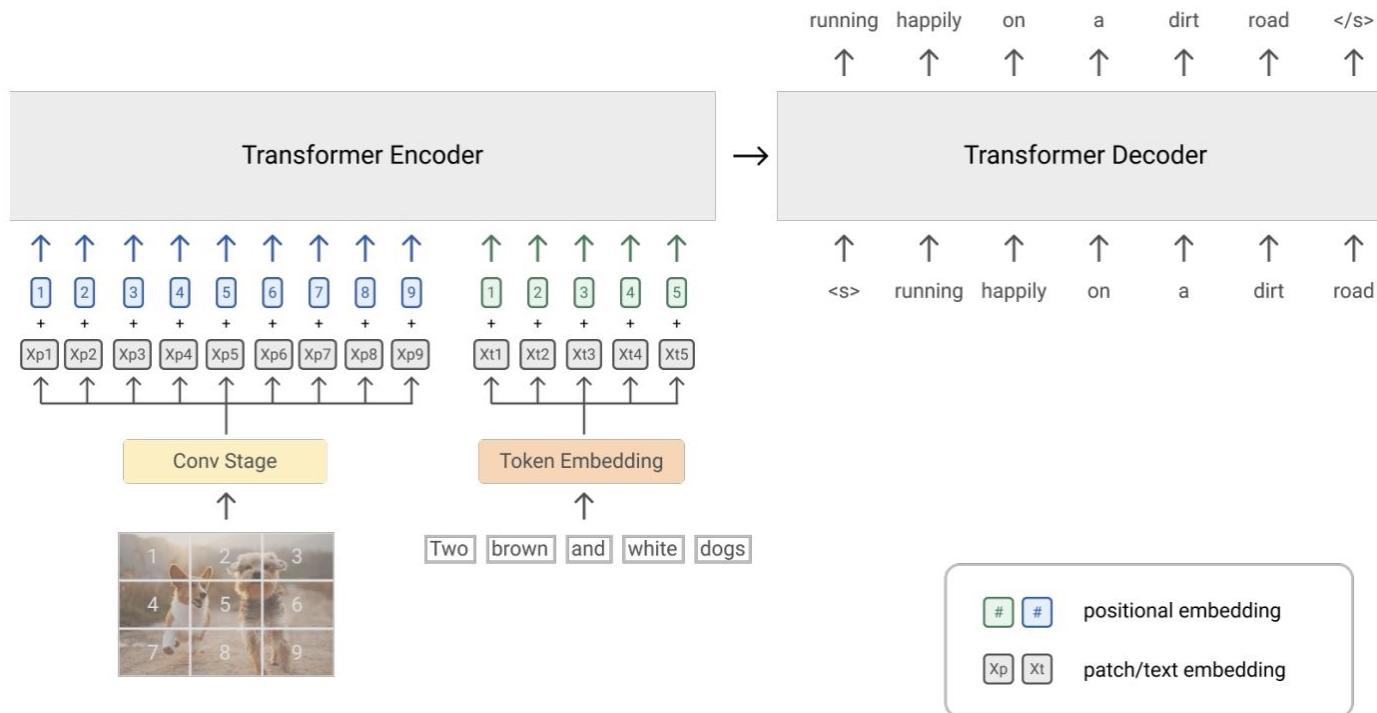
Datasets	Eval method	PMD	PMD	PMD	PMD	(PMD+IN-1k+CCNews+BC)	PMD	400M [83]
MNLI [111]	fine-tuning	—	73.23	70.99	76.82	78.06	80.33	32.85
CoLA [110]	fine-tuning	—	39.55	17.58	38.97	44.22	50.65	11.02
MRPC [29]	fine-tuning	—	73.24	76.31	79.14	78.91	84.16	68.74
QQP [40]	fine-tuning	—	86.68	85.94	88.49	98.61	88.74	59.17
SST-2 [97]	fine-tuning	—	87.96	86.47	89.33	90.14	90.94	83.49
QNLI [88]	fine-tuning	—	82.32	71.85	84.77	86.40	87.31	49.46
RTE [7, 25, 36, 40]	fine-tuning	—	50.54	51.99	51.99	54.87	57.76	53.07
STS-B [1]	fine-tuning	—	78.89	57.28	84.29	83.21	85.67	13.70
NLP Avg.		—	71.55	64.80	74.22	75.55	78.19	46.44
ImageNet [90]	linear eval	41.79	—	74.09	74.34	73.49	75.54	72.95
Food101 [11]	linear eval	53.30	—	87.77	87.53	87.39	88.51	85.49
CIFAR10 [58]	linear eval	76.20	—	93.44	92.37	92.63	92.87	91.25
CIFAR100 [58]	linear eval	55.57	—	78.37	78.01	76.49	77.68	74.40
Cars [56]	linear eval	14.71	—	72.12	72.07	66.81	70.87	62.84
Aircraft [74]	linear eval	13.83	—	49.74	48.90	44.73	47.31	40.02
DTD [20]	linear eval	55.53	—	76.86	76.91	75.80	77.29	73.40
Pets [79]	linear eval	34.48	—	84.98	84.93	82.77	84.82	79.61
Caltech101 [32]	linear eval	67.36	—	94.91	95.32	94.95	95.74	93.76
Flowers102 [76]	linear eval	67.23	—	96.36	96.39	95.58	96.37	94.94
MNIST [60]	linear eval	96.40	—	98.39	98.58	98.70	98.42	97.38
STL10 [21]	linear eval	80.12	—	98.06	98.31	98.32	98.89	97.29
EuroSAT [41]	linear eval	95.48	—	97.00	96.98	97.04	97.26	95.70
GTSRB [100]	linear eval	63.14	—	78.92	77.93	77.71	79.46	76.34
KITTI [35]	linear eval	86.03	—	87.83	88.84	88.70	89.04	84.89
PCAM [106]	linear eval	85.10	—	85.02	85.51	85.72	85.31	83.99
UCF101 [98]	linear eval	46.34	—	82.69	82.90	81.42	83.32	77.85
CLEVR [52]	linear eval	61.51	—	79.35	81.66	80.62	79.66	73.64
FER 2013 [38]	linear eval	50.98	—	59.96	60.87	58.99	61.12	57.04
SUN397 [113]	linear eval	52.45	—	81.27	81.41	81.05	82.17	79.96
SST [83]	linear eval	57.77	—	56.67	59.25	56.40	57.11	56.84
Country211 [83]	linear eval	8.87	—	27.27	26.75	27.01	28.92	25.12
Vision Avg.		57.46	—	79.14	79.35	78.29	79.44	76.12
VQAv2 [39]	fine-tuning	—	—	67.13	71.69	71.29	72.49	59.81
SNLI-VE [14]	fine-tuning	—	—	73.27	78.36	78.14	78.89	73.53
Hateful Memes [53]	fine-tuning	—	—	55.58	70.72	77.45	76.09	56.59
Flickr30K [81]	TR R@1	zero-shot	—	—	68.30	69.30	64.50	67.70
Flickr30K [81]	TR R@5	zero-shot	—	—	93.50	92.90	90.30	94.00
Flickr30K [81]	IR R@1	zero-shot	—	—	60.56	63.16	60.04	65.22
Flickr30K [81]	IR R@5	zero-shot	—	—	86.68	87.70	86.46	89.38
COCO [66]	TR R@1	zero-shot	—	—	43.08	43.48	39.88	42.74
COCO [66]	TR R@5	zero-shot	—	—	75.82	76.76	72.84	76.76
COCO [66]	IR R@1	zero-shot	—	—	37.59	38.46	34.95	38.38
COCO [66]	IR R@5	zero-shot	—	—	67.28	67.68	64.63	67.47

Plan for this lecture

1. Intro to Multimodal Large Language Models (MLLM)
2. Architectures
3. Encodings
4. Functionality
 - Contrastive Learning Models
 - Generative Models
5. Multimodal Instruction Tuning
6. What's Next?

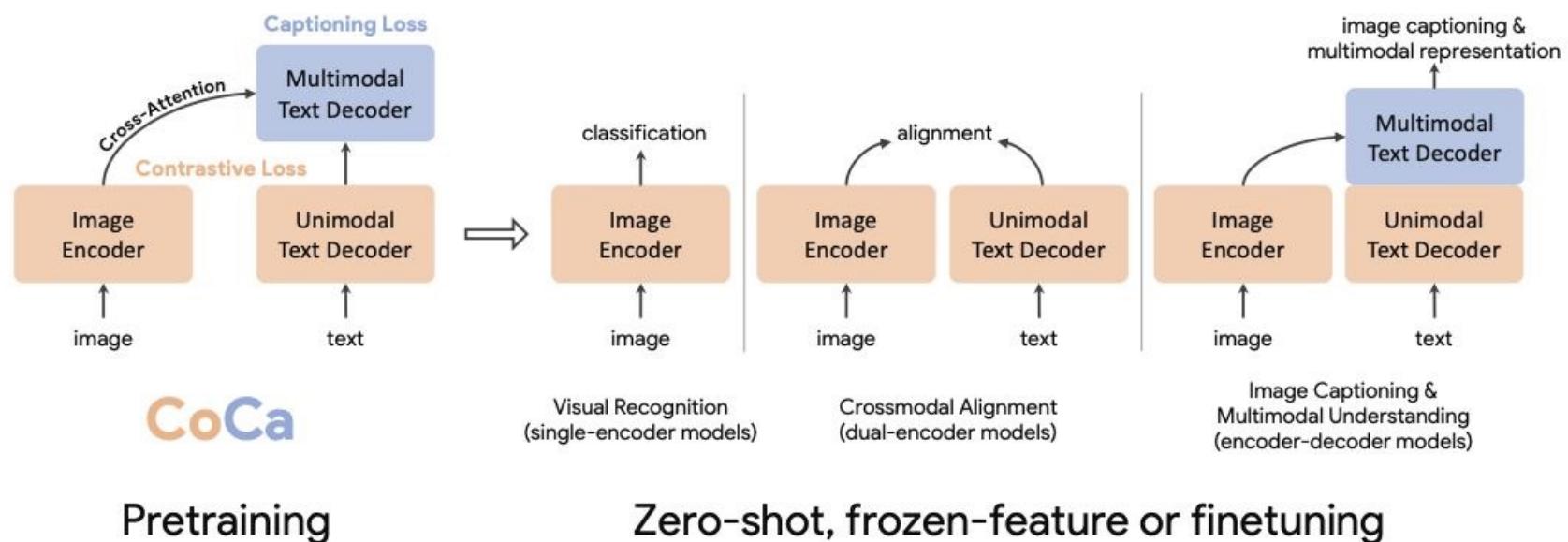
MLM Functionality: Image Perception - SimVLM (Wang et al., 2022)

Slowly moving from contrastive/discriminative to generative.



MLM Functionality: Image Perception - CoCa Contrastive Captioner (Yu et al., 2022)

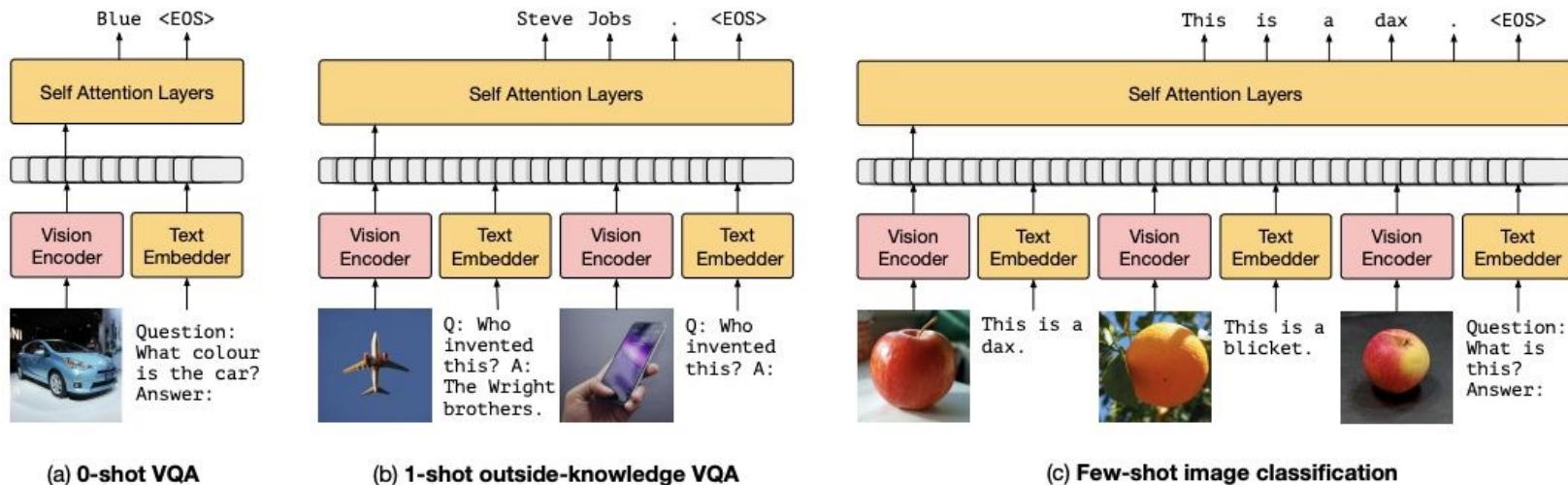
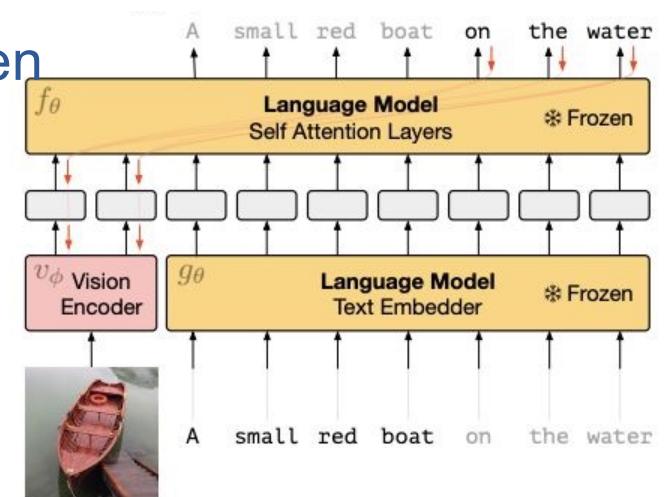
Best of both (contrastive and generative) worlds.



MLLM Functionality: Image Perception - Frozen (Tsimpoukelli, Menick, Cabi, et al., 2021)

Kind of like MMBT* but with a better LLM (T5) and a better vision encoder (NF-ResNet).

Multi-Modal Few-Shot Learners!

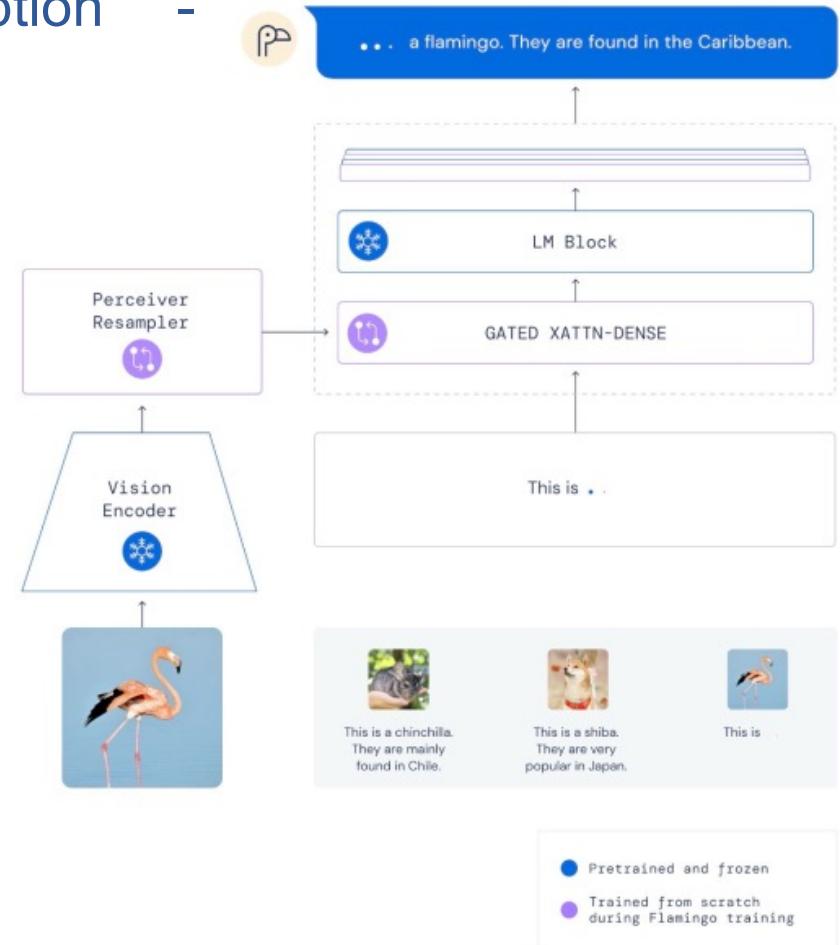
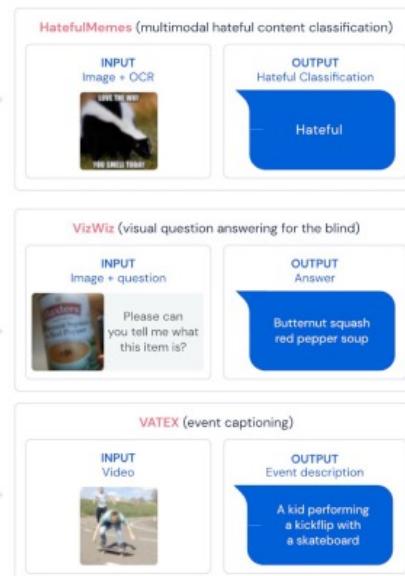
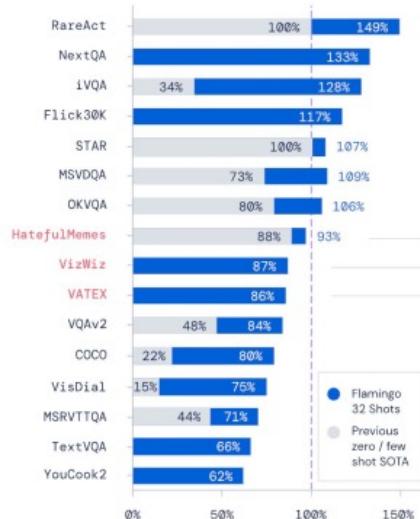


MLM Functionality: Image Perception - Flamingo (Alayrac et al., 2022)

80b param model based on Chinchilla.

Multi-image.

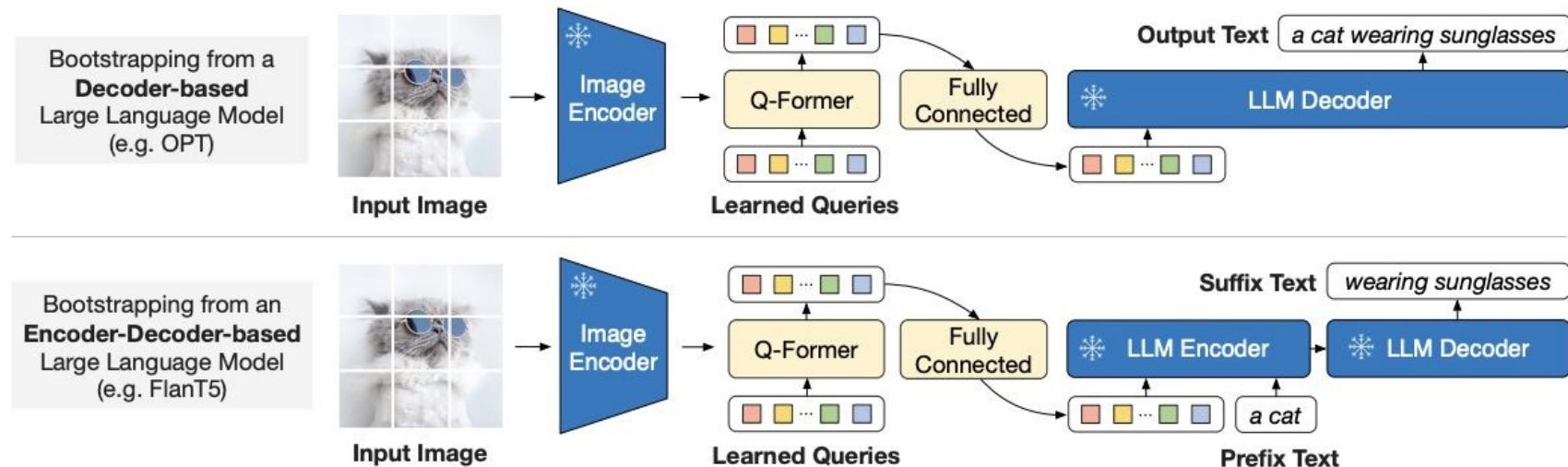
Performance relative to SOTA



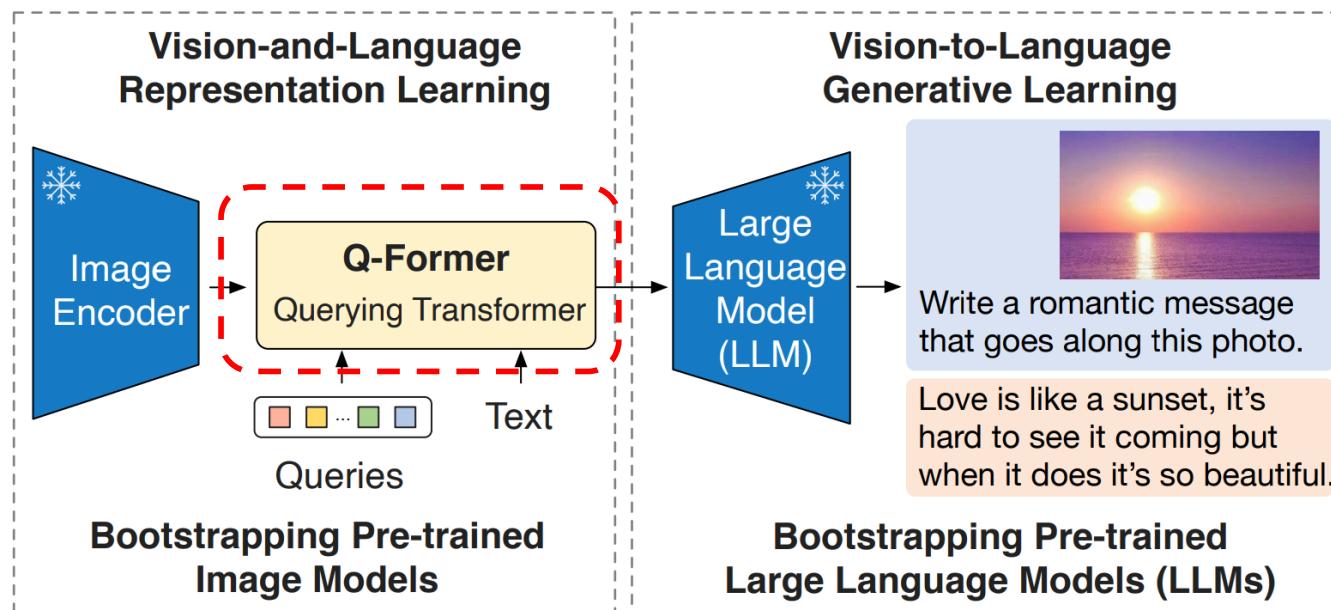
MLLM Functionality: Image Perception - BLIP/BLIP2 (Li et al., 2023)

Freeze it all (CLIP-ViT / OPT decoder / FlanT5 encoder-decoder)

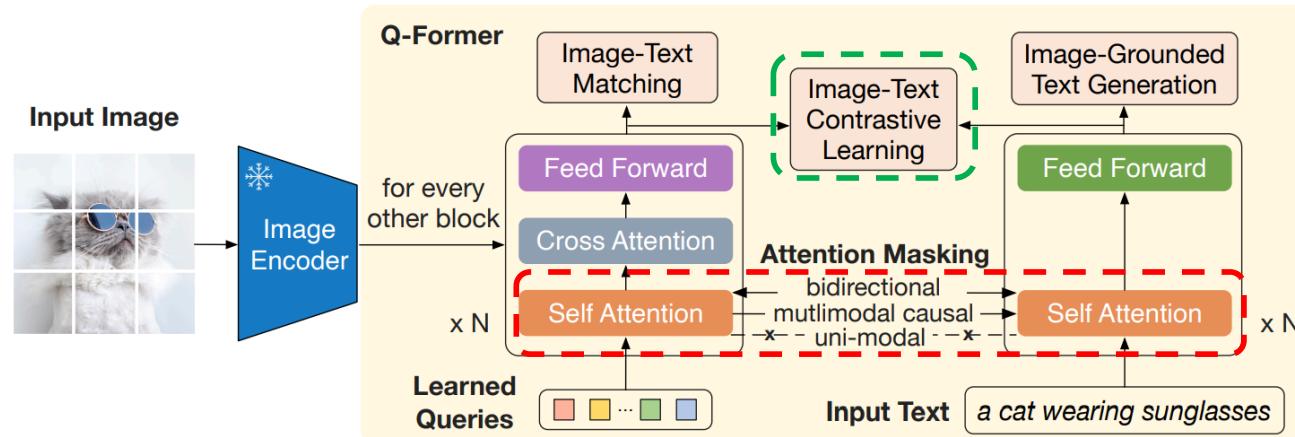
BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models



BLIP2: Bootstrapping language-image pre-training with frozen image encoders and large language models



BLIP2 Architecture: Q-Former



- Extracts fixed # of features from image encoder
- Has image and text transformers **with same self-attention layers**
- Learnable query embeddings (**Z**) are inputs to image transformer
 - 32x768; can interact with each other, text, and frozen image features
 - **Goal:** Extract visual info most relevant to the text
- Initialized with $\text{BERT}_{\text{base}}$ weights
- 188M parameters

BLIP2 - Stage 1 Training: Representation Learning

- Three objectives are jointly optimized using different self-attention masking strategies to control query-text interaction

Q: query token positions; **T:** text token positions.
■ masked □ unmasked

	Q	T
Q	□ □	□ □
T	□ □	□ □

Bi-directional
Self-Attention Mask
Image-Text
Matching

	Q	T
Q	□ □	■ ■
T	□ □	■ □

Multi-modal Causal
Self-Attention Mask
Image-Grounded
Text Generation

	Q	T
Q	□ □	■ ■
T	■ ■	□ □

Uni-modal
Self-Attention Mask
Image-Text
Contrastive Learning

Goal: Fine-grained alignment
Task: Binary classification if image-text pair is matching
Masking: All queries/text can attend to each other

Goal: Generate text conditioned on image
Task: Decode text
Masking: Queries can attend to each other but not the text tokens. Text can attend to queries and previous text tokens.

Goal: Alignment
Task: Contrastive learning with in-batch negatives (original BLIP uses momentum queue)
Masking: Text only attends to text and queries to queries to avoid info leak

BLIP2 - Stage 2: Generative Learning

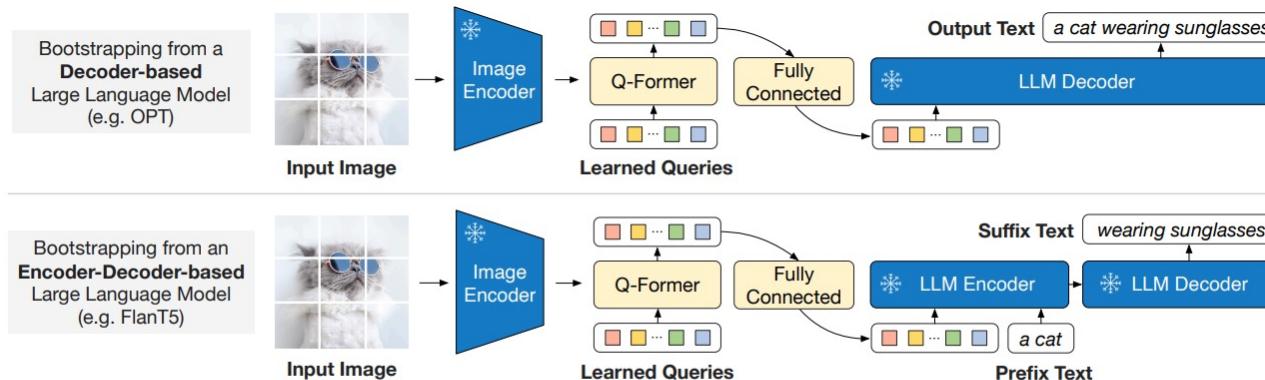


Figure 3. BLIP-2’s second-stage vision-to-language generative pre-training, which bootstraps from frozen large language models (LLMs). **(Top)** Bootstrapping a decoder-based LLM (e.g. OPT). **(Bottom)** Bootstrapping an encoder-decoder-based LLM (e.g. FlanT5). The fully-connected layer adapts from the output dimension of the Q-Former to the input dimension of the chosen LLM.

- Query embeddings \mathbf{Z} projected into LLM embedding space and prepended to input text embeddings
 - A soft visual prompting mechanism for the LLM
- Different loss forms are used for different LM architectures
 - **Decoder** – language modeling loss over the queries
 - **Encoder-decoder** – prefix pretrained with LM loss; suffix used as generation target

Kyle Buettner

BLIP2 - Instructed Zero-Shot Image-to-Text Generation (1/2)

 <p>Explain the advantages of this product.</p> <p>The audi e-tron quattro concept is a plug-in hybrid electric sports car that has a range of 310 miles and can accelerate from 0 to 60 mph in just 4 seconds.</p>	 <p>Tell me something about the history of this place.</p> <p>The great wall of china was built in 221 bc by emperor qin shihuang to protect the emperor's capital from invaders from the north.</p>	 <p>Write down the facts that you know about this flower.</p> <p>It is a genus of flowering plants in the orchid family, orchidaceae. It is native to eastern asia and south east asia. It is a perennial flowering plant that grows up to a height of 1 m.</p>
 <p>Is this photo unusual?</p> <p>Yes, it's a house that looks like it's upside down.</p> <p>How could someone get out of the house?</p> <p>It has a slide on the side of the house.</p>	 <p>What are shown in the photo?</p> <p>A man and a chicken.</p> <p>What does the man feel and why?</p> <p>He is scared of the chicken because it is flying at him.</p>	 <p>What are the ingredients I need to make this?</p> <p>Pepperoni, mozzarella cheese, pizza sauce, olive oil, salt, pepper, basil.</p> <p>What is the first step?</p> <p>Place the pizza dough on a baking sheet, brush with olive oil, sprinkle with salt, pepper, and basil.</p>

BLIP2 - Instructed Zero-Shot Image-to-Text Generation (2/2)

 <p>Which city is this? 8 Singapore. 8 Why do you think so? 8 The city has a statue of a merlion. 8</p>	 <p>What happened at the end of this movie? 8 The titanic sank. 8 Did Leonardo Dicaprio's character survive? 8 No, he drowned. 8</p>	 <p>What is in the photo? 8 A pizza that looks like a cat. 8 What is the nose made of? 8 A slice of pepperoni. 8</p>
 <p>Write a romantic message that goes along with this photo. 8 Love is like a sunset, it's hard to see it coming but when it does it's so beautiful. 8</p>	 <p>Tell us about the photo you took for Darren and Jade. 8 Darren and Jade are a couple who have a passion for horses, so we decided to take a photo of them with a horse in the desert on their wedding day. 8</p>	 <p>Write a conversation between the two animals. cat: hey dog, can i ride on your back? dog: sure, why not? cat: i'm tired of walking in the snow. 8</p>

BLIP2 - Example Issues

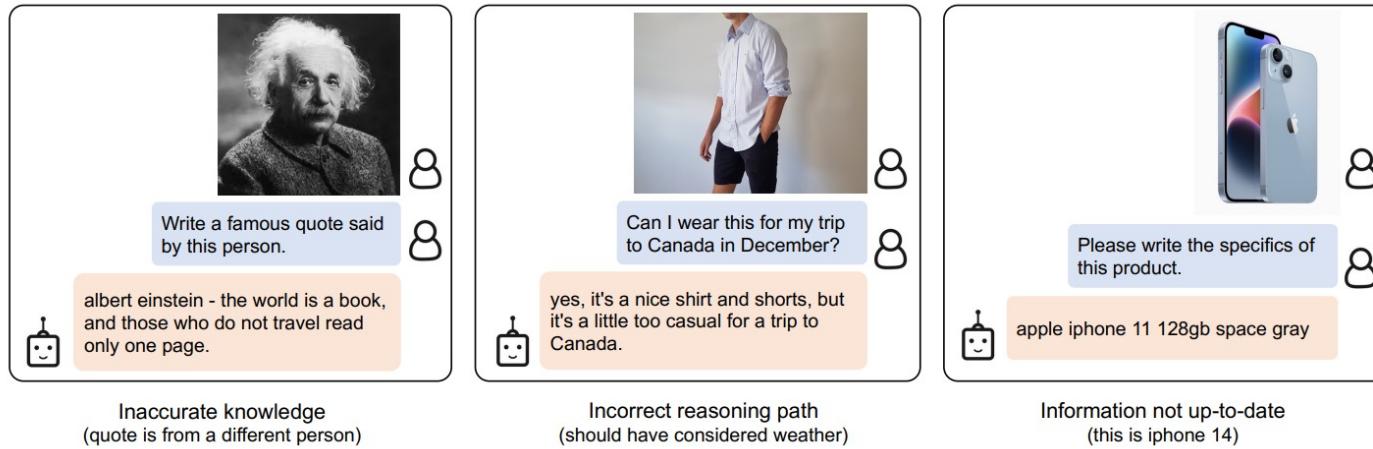
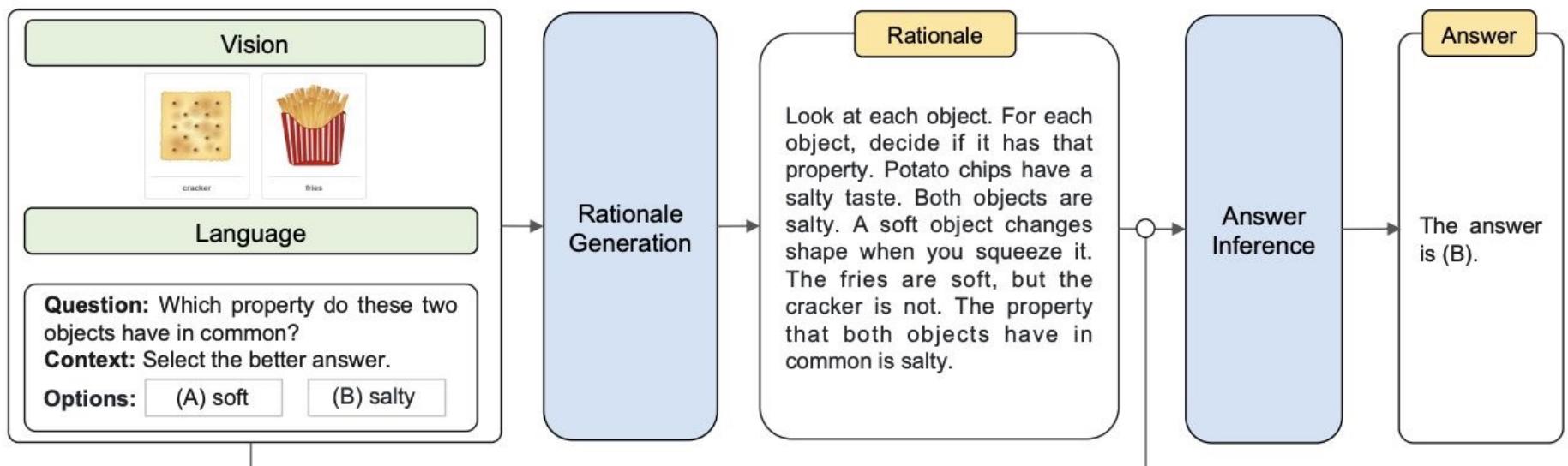


Figure 6. Incorrect output examples for instructed zero-shot image-to-text generation using a BLIP-2 model w/ ViT-g and FlanT5_{XXL}.

MLLM Functionality: Image Perception - Multimodal “Chain of Thought” (Zhang et al., 2023)

Providing a rationale helps give the right answer.



MLLM Functionality: Image Perception - KOSMOS-1 (Huang et al., 2023)

LLMs => MLLMs == Foundation Model

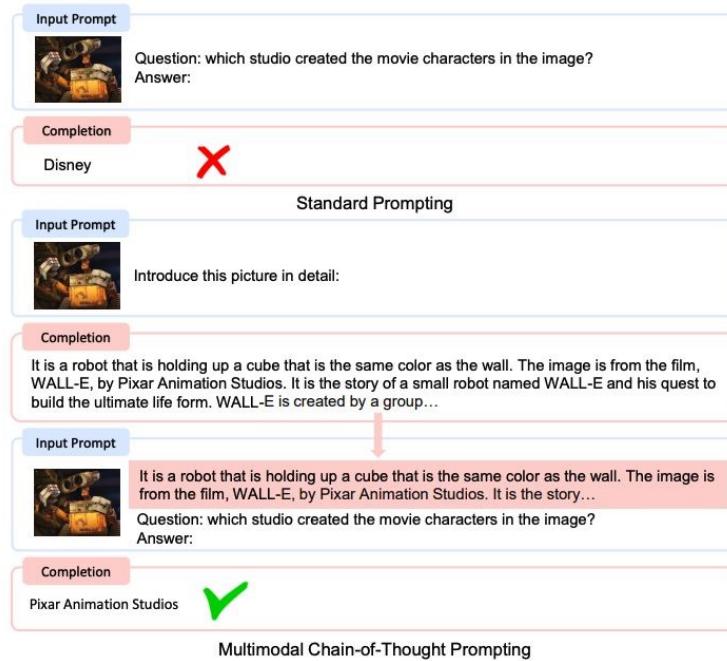
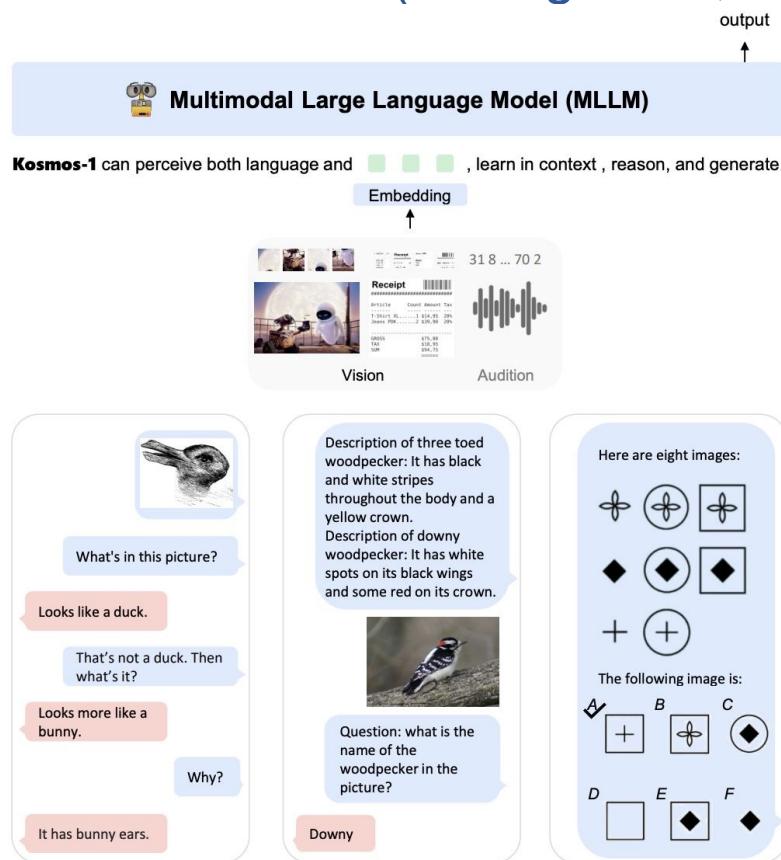


Figure 5: Multimodal Chain-of-Thought prompting enables KOSMOS-1 to generate a rationale first, then to tackle complex question-answering and reasoning tasks.



MLM Functionality: Image Perception - LLaVA: Large Language and Vision Assistant (Liu NeurIPS'23)

- Instruction tuning in multimodal space
- Contributions
 - 1) Data creation strategy to create instruction-following multimodal data (from image-text pairs)
 - 2) Large multimodal model – LLaVA – open-set visual encoder of CLIP connected with language decoder LLaMA, finetuned end-to-end
 - State-of-the-art performance on ScienceQA dataset
 - 3) Open-source assets – multimodal instruction data, codebase for data generation/training, checkpoint, visual chat demo

MLM Functionality: Image Perception - LLaVA

GPT-Assisted Visual Instruction Data Generation

- Amount of multimodal instruction-following data is limited, but **image-text pairs** are widely available
 - Conceptual Captions, LAION
- Approach: Use ChatGPT/GPT-4 to create instruction data
 - Create set of questions X_q with intent to instruct assistant to describe image content
 - Input: Image X_v , Caption X_c
 - Use simple/cheap idea to expand $\langle X_v, X_c \rangle$
 - $X_q X_v <\text{STOP}> \backslash n$ Assistant: $X_c <\text{STOP}> \backslash n$.
 - But lacks diversity and in-depth reasoning...

MLM Functionality: Image Perception - LLaVA

GPT-Assisted Visual Instruction Data Generation

- To expand data, use two symbolic representations for image and input into LLM (ChatGPT/GPT-4)
 - **Captions**
 - **Bounding boxes** for each object in the scene
- Use these (from COCO images) to generate 3 types of instruction-following data with LLMs
 - **Conversation** – QA about object types, counts, actions, locations, etc.
 - **Detail description** – detailed/comprehensive text
 - **Complex reasoning** – more complex QA
- For each type, a few manually designed examples are used to seed in-context learning
 - *Only human annotations in data collection*
- 158K unique samples created overall

Kyle Buettner

MLM Functionality: Image Perception - LLaVA

GPT-Generated “Brief” Instructions

Instructions for brief image description. The list of instructions used to briefly describe the image content are shown in Table 8. They present the same meaning with natural language variance.

- "Describe the image concisely."
- "Provide a brief description of the given image."
- "Offer a succinct explanation of the picture presented."
- "Summarize the visual content of the image."
- "Give a short and clear explanation of the subsequent image."
- "Share a concise interpretation of the image provided."
- "Present a compact description of the photo's key features."
- "Relay a brief, clear account of the picture shown."
- "Render a clear and concise summary of the photo."
- "Write a terse but informative summary of the picture."
- "Create a compact narrative representing the image presented."

Table 8: The list of instructions for brief image description.

MLM Functionality: Image Perception - LLaVA

GPT-Generated “Detailed” Instructions

Instructions for detailed image description. The list of instructions used to describe the image content in detail are shown in Table 9. They present the same meaning with natural language variance.

- "Describe the following image in detail"
- "Provide a detailed description of the given image"
- "Give an elaborate explanation of the image you see"
- "Share a comprehensive rundown of the presented image"
- "Offer a thorough analysis of the image"
- "Explain the various aspects of the image before you"
- "Clarify the contents of the displayed image with great detail"
- "Characterize the image using a well-detailed description"
- "Break down the elements of the image in a detailed manner"
- "Walk through the important details of the image"
- "Portray the image with a rich, descriptive narrative"
- "Narrate the contents of the image with precision"
- "Analyze the image in a comprehensive and detailed manner"
- "Illustrate the image through a descriptive explanation"
- "Examine the image closely and share its details"
- "Write an exhaustive depiction of the given image"

Table 9: The list of instructions for detailed image description.

MLM Functionality: Image Perception - LLaVA

Example Context to Prompt LLMs

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], person: [0.63, 0.222, 0.686, 0.516], person: [0.444, 0.233, 0.487, 0.34], backpack: [0.384, 0.696, 0.485, 0.914], backpack: [0.755, 0.413, 0.846, 0.692], suitcase: [0.758, 0.413, 0.845, 0.69], suitcase: [0.1, 0.497, 0.173, 0.579], bicycle: [0.282, 0.363, 0.327, 0.442], car: [0.786, 0.25, 0.848, 0.322], car: [0.783, 0.27, 0.827, 0.335], car: [0.86, 0.254, 0.891, 0.3], car: [0.261, 0.101, 0.787, 0.626]

MLM Functionality: Image Perception - LLaVA

Example Responses from LLM

Response type 1: conversation

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

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

Question: Where is the vehicle parked?

Answer: The vehicle is parked in an underground parking area, likely in a public garage.

Question: What are the people in the image doing?

Answer: The people are trying to fit all of their luggage into the SUV, likely preparing for a trip.

Response type 2: detailed description

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

In the parking area, various luggage items are scattered around the vehicle. There are two backpacks, one located near the left rear wheel and the other closer to the right side of the vehicle. Additionally, there are two suitcases, one on the right side of the car and another further away near the center of the parking area. A bicycle can also be seen on the left side of the vehicle.

Other cars are parked around the main SUV, with one car positioned behind it and slightly to the left, another behind and slightly to the right, and the third car further behind on the right side.

Response type 3: complex reasoning

Question: What challenges do these people face?

Answer: In the image, a group of people is standing outside a black SUV in a parking area, surrounded by various pieces of luggage, including suitcases and backpacks. They are facing the challenge of fitting all their luggage into the black SUV. There are multiple suitcases and backpacks to be packed, which suggests that the group has a significant amount of belongings to accommodate. They might have to strategize and arrange the luggage efficiently to ensure that everything fits properly into the vehicle. Additionally, they need to consider the comfort of the passengers and visibility while driving, so the placement of the luggage must not obstruct the driver's view or make the passengers uncomfortable during the trip.

MLM Functionality: Image Perception - LLaVA

Conversation Generation

```
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.

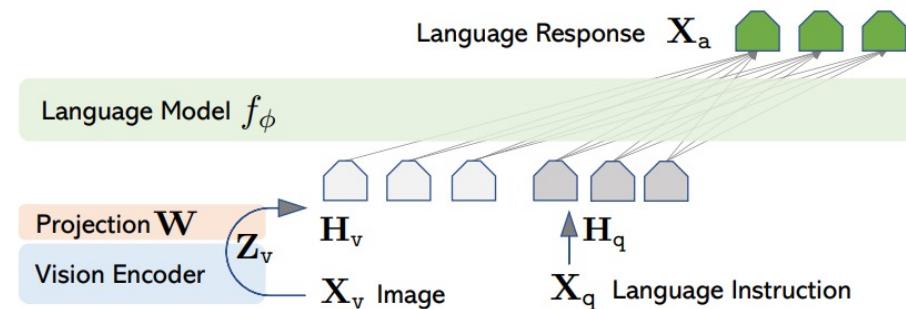
Also include complex questions that are relevant to the content in the image, for example, asking about background knowledge of the objects in the image, asking to discuss about events happening in the image, etc. Again, do not ask about uncertain details. Provide detailed answers when answering complex 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)})
```

MLM Functionality: Image Perception - LLaVA

LLaVA Model

- How can visual instruction data be used?
- LLM = LLaMA
- Vision encoder = CLIP ViT-L/14
 - Features linearly projected into word embedding space (layer trainable)
 - Lightweight vs. gated cross-attention of Flamingo/Q-Former in BLIP-2



Kyle Buettner

Figure 1: LLaVA network architecture.

MLM Functionality: Image Perception - LLaVA

LLaVA Training

- For each image X_v , multi-turn conversation data is generated $\rightarrow (X_q^1, X_a^1, \dots, X_q^T, X_a^T)$; $T = \# \text{ of turns}$
- Goal is to learn probability of generating answers based on previous conversation text and image

```

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

```

Table 2: The input sequence used to train the model. Only two conversation turns are illustrated here; in practice, the number of turns varies based on the instruction-following data. In our current implementation, $X_{\text{system-message}} = \text{A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions. and } <\text{STOP}> = \#\#\#.$ The model is trained to predict the assistant answers and where to stop, and thus only green sequence/tokens are used to compute the loss in the auto-regressive model.

$$X_{\text{instruct}}^t = \begin{cases} \text{Random choose } [X_q^1, X_v] \text{ or } [X_v, X_q^1], & \text{the first turn } t = 1 \\ X_q^t, & \text{the remaining turns } t > 1 \end{cases}$$

$$p(X_a | X_v, X_{\text{instruct}}) = \prod_{i=1}^L p_{\theta}(x_i | X_v, X_{\text{instruct}, <i}, X_{a, <i})$$

MLM Functionality: Image Perception - LLaVA

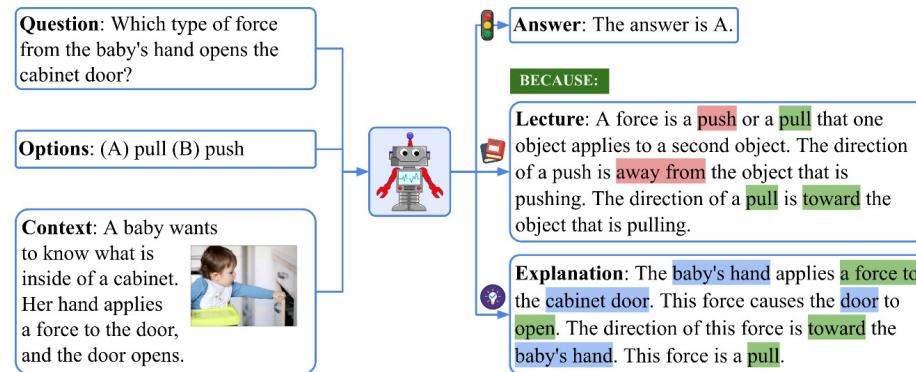
Two-Stage Instruction Tuning

- Stage 1: Pretraining for Feature Alignment
 - Conceptual Captions 3M filtered to 595K image-text pairs for efficiency
 - Converted to instruction-following data using simple expansion strategy
 - Each sample treated as single-turn conversation
 - Question X_q randomly sampled, X_a original caption
 - Visual encoder and LLM weights frozen, projection layer trained
 - *“Training a compatible visual tokenizer for the frozen LLM”*
- Stage 2: Finetuning End-to-End
 - Visual encoder weights frozen, projection layer and LLM updated
 - Use cases
 - Multimodal chatbot – 158K unique language-image instruction-based data
 - Science QA
 - Context can be image/language
 - Answer from multiple choices, along with reasoning

MLM Functionality: Image Perception - LLaVA

Evaluation: ScienceQA (Lu et al., NeurIPS 2022)

- 21k multimodal multiple-choice questions



Biology Genes to traits Classification Adaptations Traits and heredity Ecosystems Classification Scientific names Heredity Ecological interactions Cells Plants Animals Plant reproduction	Physics Materials Magnets Velocity and forces Force and motion Particle motion and energy Heat and thermal energy States of matter Kinetic and potential energy Mixture	Geography State capitals Geography Maps Oceania: geography Physical Geography The Americas: geography Oceans and continents Cities States	History Colonial America English colonies in North America The American Revolution	Civics Social skills Government The Constitution
Earth Science Weather and climate Rocks and minerals Astronomy Fossils Earth events Plate tectonics	Chemistry Solutions Physical and chemical change Atoms and molecules Chemical reactions	Writing Strategies Supporting arguments Sentences, fragments, and run-ons Word usage and nuance Creative techniques	Vocabulary Categories Shades of meaning Comprehension strategies Context clues	Verbs Verb tense
	Engineering Designing experiments Engineering practices	Audience, purpose, and tone Pronouns and antecedents Persuasive strategies Editing and revising Visual elements Opinion writing	Capitalization Formatting	
	Units and Measurement Weather and climate		Punctuation Fragments	
			Phonology Rhyming	
			Figurative Language Literary devices	Reference Research skills

MLM Functionality: Image Perception - LLaVA

Evaluation: Multimodal Chatbot

- Example from GPT-4 paper
- Describing image vs. intent
- Small (~80K unique image) data of LLaVA effective

Visual input example, Extreme Ironing:

User LLaVA	
User LLaVA	<p>Source: https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg</p> <p>What is unusual about this image?</p> <p>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 at 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.</p> <p>[Start a new conversation, and clear the history]</p> <p>What's happening in the scene?</p> <p>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.</p>
User GPT-4 [32]	<p>What is unusual about this image?</p> <p>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.</p>
User BLIP-2	<p>Can you explain this meme in detail?</p> <p>a man is sitting on the back of a yellow cab</p>
User OpenFlamingo	<p>Can you explain this meme in detail?</p> <p>The man is drying his clothes on the hood of his car.</p>

Table 4: Example prompt demonstrating LLaVA and GPT-4's visual input capability. The prompt requires image understanding.

MLM Functionality: Image Perception - LLaVA

Quantitative Evaluation

- Use GPT-4 to measure the quality of model's responses
- Randomly sample 30 images from COCO val
- Generate each question type; GPT-4 serves as reference
- After getting GPT-4/LLaVA predictions, question/visual info/responses fed into GPT-4
- GPT-4 evaluates helpfulness, relevance, accuracy, and level of details; scale 1->10; results explained

	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 3: Relative scores for different settings *w.r.t.* GPT-4 (text-only) on 30 randomly sampled images from COCO Val 2014. Each image is associated one short question, one detailed question, and one complex reasoning question, resulting in a total of 90 questions. Following the same setting as our data generation pipeline, GPT-4 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.

MLM Functionality: Image Perception - LLaVA

LLaVA in Action



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.

I would like to do the fruit salad. Show me the recipe.

Certainly! Here's a simple fruit salad recipe using the items in your refrigerator:

Ingredients:

- 2 cups fresh strawberries, sliced
- 1 cup fresh blueberries
- 1 large carrot, grated
- Juice of half a lemon
- A handful of chopped fresh parsley or mint
- A few ice cubes
- Optional: honey or maple syrup (adjust to taste)

Instructions:

1. In a large bowl, combine the sliced strawberries, blueberries, and carrot. Gently toss them together to mix.
2. Drizzle the juice of half a lemon over the fruit and vegetables to prevent them from oxidizing and add a refreshing citrus flavor.
3. Sprinkle the chopped fresh parsley or mint over the fruit and gently toss once more.
4. If desired, add a drizzle of honey or maple syrup for a touch of sweetness. Adjust the amount according to your taste preferences.
5. Cover the bowl and refrigerate the fruit salad for at least 30 minutes or up to a few hours. This allows the flavors to meld and creates a more refreshing salad.
6. Serve the fruit salad in cups or bowls, garnished with a sprinkle of fresh parsley or mint, if desired.

This fruit salad recipe is perfect for a healthy snack or side dish, and with the variety of ingredients available in your refrigerator, it should be easy to put together quickly.

Kyle Buettner

MLM Functionality: Image Perception - LLaVA

LLaVA in Action



What are the things I should be cautious about when I visit here?



When visiting this peaceful location with a pier extending over a serene lake, surrounded by mountains, there are several things to be cautious about. First, pay attention to the weather conditions, as sudden changes in weather, especially strong winds or storms, can cause the water to become rough and pose risks to the stability of the pier. Second, be mindful of the structural integrity of the pier; avoid overloading it with too many people or excessive weight, as this could potentially damage the pier or cause it to collapse. Additionally, be aware of the surrounding wildlife and respect their space; do not harm or disturb them in any way. Finally, if you decide to go fishing, ensure that you adhere to any local fishing regulations and use appropriate fishing techniques to keep the lake ecosystem healthy and maintain the balance of the food chain.



Compose an engaging travel blog post about my trip to this place, highlighting the beautiful sceneries, joyful experiences, and include possible anecdotes.



Traveling to this picturesque location with a serene lake and mountains in the background was a refreshing and rejuvenating experience. The pier extending over the calm water provided an excellent vantage point from which to appreciate the breathtaking scenery. As I stood on the pier, I couldn't help but feel a deep connection with nature, taking a moment to soak in the peaceful atmosphere.

One of the highlights of my trip was undoubtedly the fishing experience. The area is teeming with life, and I was thrilled to land a sizeable catch, despite the modest size of the lake. The joy of fishing in such a beautiful setting was truly rewarding, and it allowed me to take a break from my busy life and truly immerse myself in the tranquility of nature.

While exploring the surrounding area, I came across a family of birds that had built a nest close to the pier, adding to the charm of the location. Interacting with the wildlife and observing their behavior was an enriching experience that I'll cherish for years to come.

In conclusion, this idyllic destination provided me with an opportunity to unwind and recharge, surrounded by mesmerizing natural beauty and the gentle sounds of water and wildlife. If you're looking for a serene escape from the hustle and bustle of everyday life, this pier overlooking the lake is the perfect place to find solace and reconnect with nature.

Kyle Buettner

MLM Functionality: Image Perception - LLaVA

LLaVA in Action



In the image, a young woman is leaning on a man in front of a beautiful sunset, with both of them standing on a boat. They appear to be enjoying a romantic moment together. The scene suggests that they could be depicted as the characters Jack and Rose from the popular movie Titanic. In the movie, their romance blossoms aboard the fictional ocean liner Titanic, with the pair ultimately sharing a passionate kiss as the ship sinks into the ocean. The ending of the movie is known for its emotional and heartfelt scenes, where the characters are forced to say goodbye to each other as they are separated by the sinking ship. The image evokes a sense of romance and nostalgia, paying tribute to the iconic love story portrayed in the movie Titanic.



What's the ending of this movie?

Figure 4: LLaVA relates the movie scenes to the textural knowledge from the pretrained LLM.

Kyle Buettner

MLM Functionality: Image Perception - LLaVA

LLaVA in Action

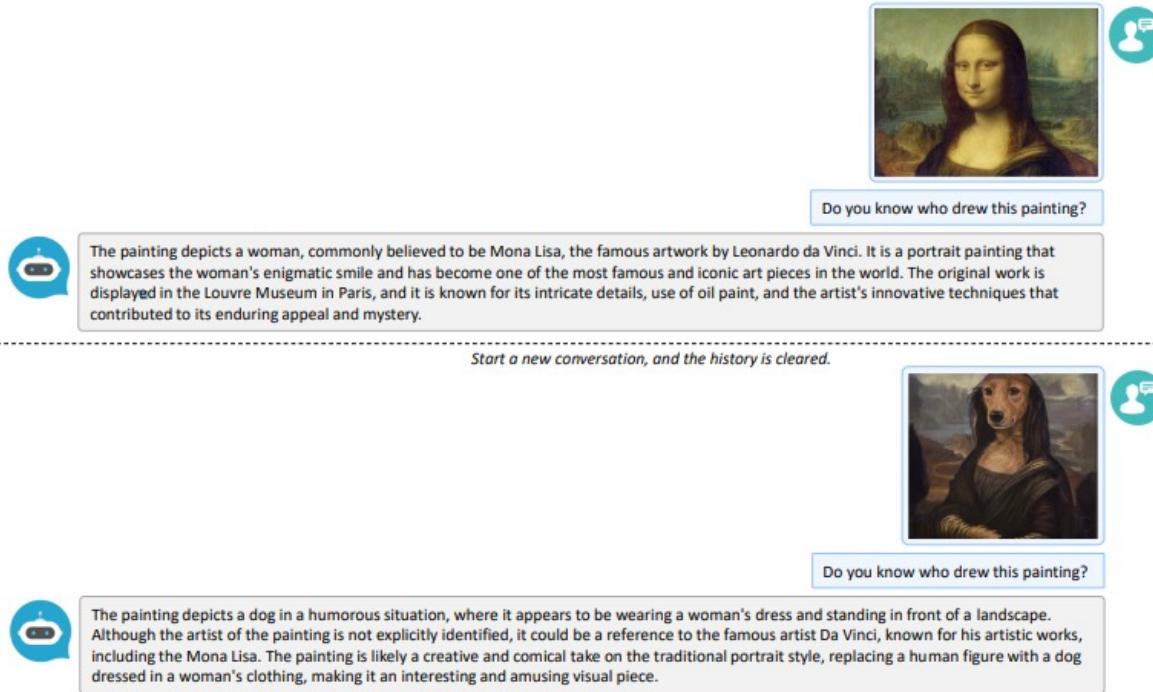
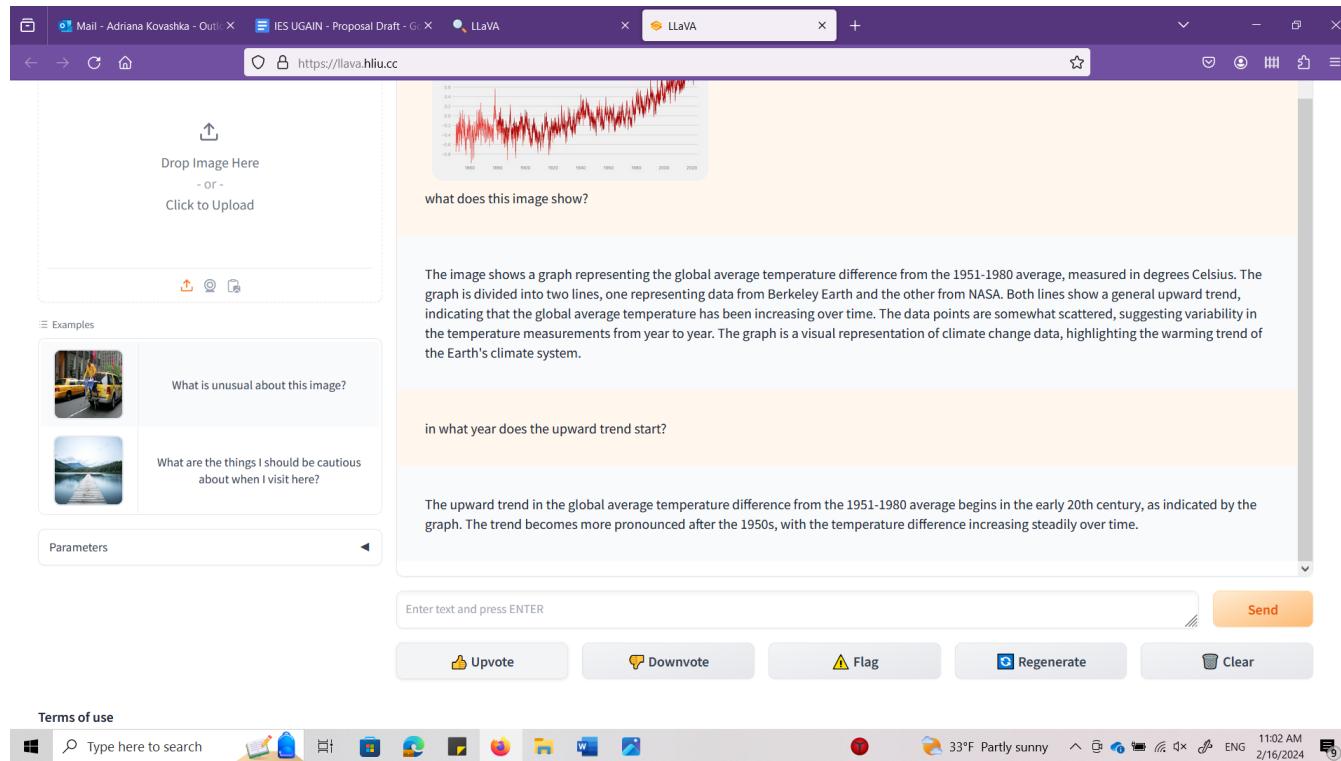


Figure 5: LLaVA recognizes the famous art work, Mona Lisa, by Leonardo da Vinci. When we start a new conversation, it also explains the humorous artwork created on the web, mimicking the Mona Lisa.

Kyle Buettner

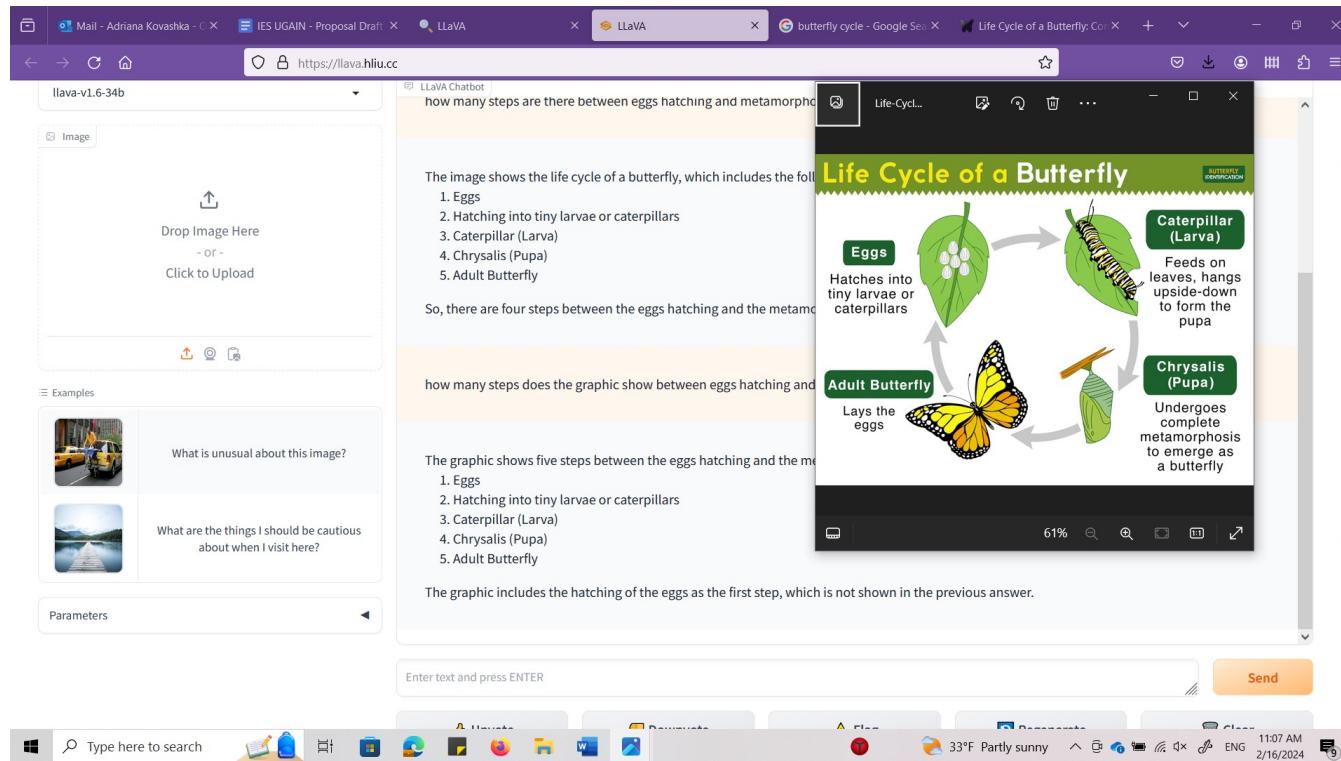
MLM Functionality: Image Perception - LLaVA

LLaVA in Action



MLM Functionality: Image Perception - LLaVA

LLaVA in Action



MLLM Functionality: Image Perception - LLaVA

LLaVA in Action

The diagram illustrates the hydrological cycle with the following labeled components:

- Transpiration:** Arrows pointing upwards from trees.
- Evaporation:** Arrows pointing upwards from the ground.
- Condensation:** A cloud with rain falling from it.
- Precipitation:** Rain falling from clouds.
- Snowmelt Runoff:** Red arrows pointing down from a snow-covered mountain into a river.
- Surface Runoff:** Red arrows pointing down from a grassy area into a river.

LLaVA Chatbot Response:

Based on the information provided in the image, trees are not strictly necessary for precipitation to happen. The image shows the water cycle with various processes such as evaporation, condensation, precipitation, and transpiration. While transpiration is one of the processes depicted, it is not the only factor that contributes to precipitation.

Precipitation is shown as a result of condensation, which occurs when water vapor in the atmosphere condenses into droplets or ice crystals. This process can happen independently of the presence of trees. The image does not depict any requirement for trees to be present for precipitation to occur.

Therefore, while trees can contribute to the water cycle through transpiration, they are not a strict requirement for precipitation to happen.

according to the figure, what process other than transpiration from trees can fuel condensation?

According to the figure, besides transpiration from trees, the process of evaporation from the ground and other surfaces can also fuel condensation. The image shows arrows pointing upwards from the ground, indicating evaporation. This process involves the conversion of liquid water into water vapor, which can then rise into the atmosphere and contribute to the formation of clouds through condensation.

Evaporation from the ground can occur due to various factors such as sunlight, wind, and temperature. As the water on the ground surface heats up, it evaporates into the air, increasing the amount of water vapor in the atmosphere. This water vapor can then condense into clouds, leading to precipitation.

In summary, the image shows that both transpiration from trees and evaporation from the ground can contribute to the formation of clouds through condensation, which is a key process in the water cycle leading to precipitation.

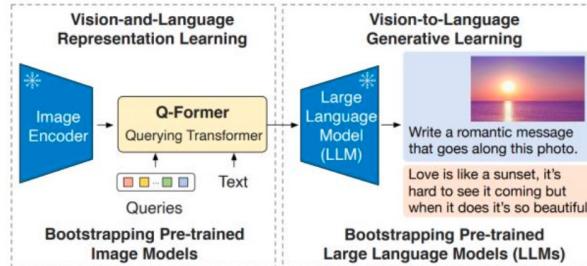
Enter text and press ENTER Send

Windows taskbar: Type here to search, File Explorer, Microsoft Edge, File, Documents, File, Microsoft Word, Microsoft Excel, Microsoft Powerpoint, Microsoft Access, 33°F Partly sunny, 11:21 AM, 2/16/2024

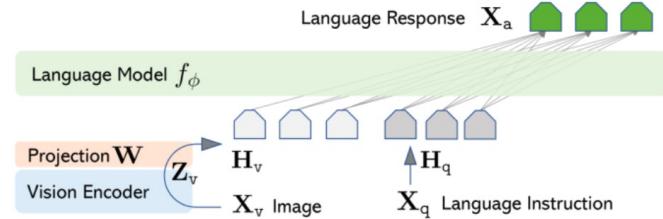
<https://llava.hliu.cc/>

MLLM Functionality: Image Perception

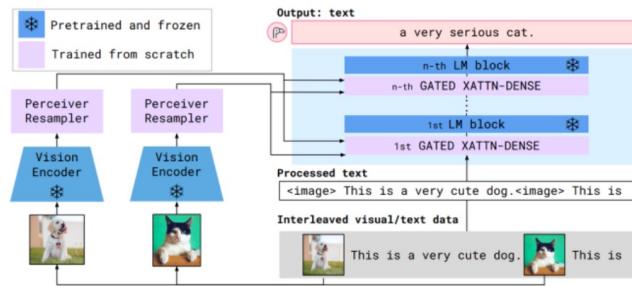
Blip2



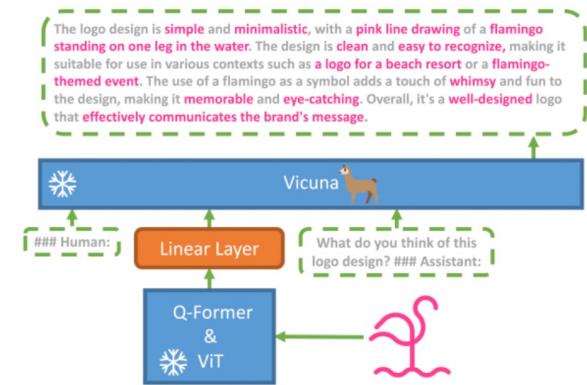
LLaVA



Flamingo



Mini-GPT4



[1] Flamingo: a Visual Language Model for Few-Shot Learning. 2022

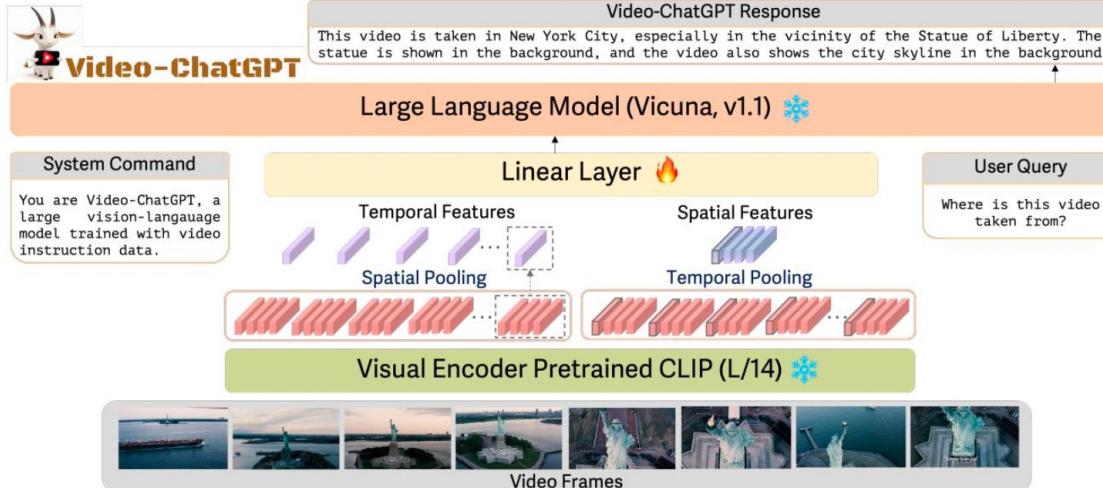
[2] BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. 2023

[3] Visual Instruction Tuning. 2023

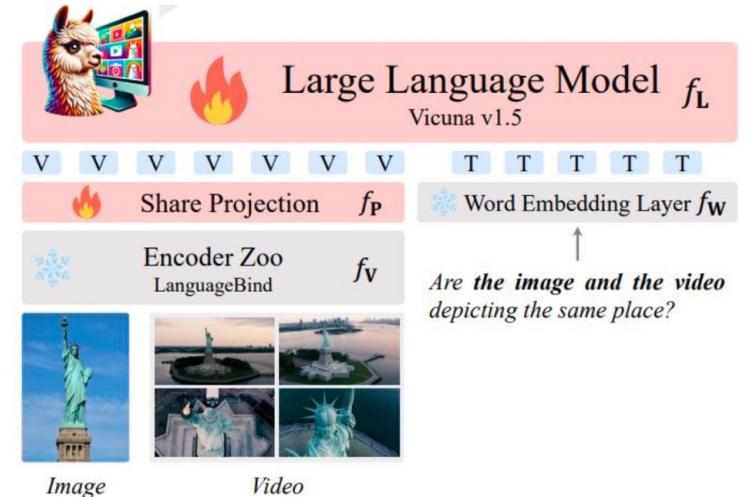
[4] A Survey on Multimodal Large Language Models. <https://github.com/BradyFU/Awesome-Multimodal-Large-Language-Models> , 2023.

MLLM Functionality: Video Perception

+ Video-ChatGPT



+ Video-LLaVA



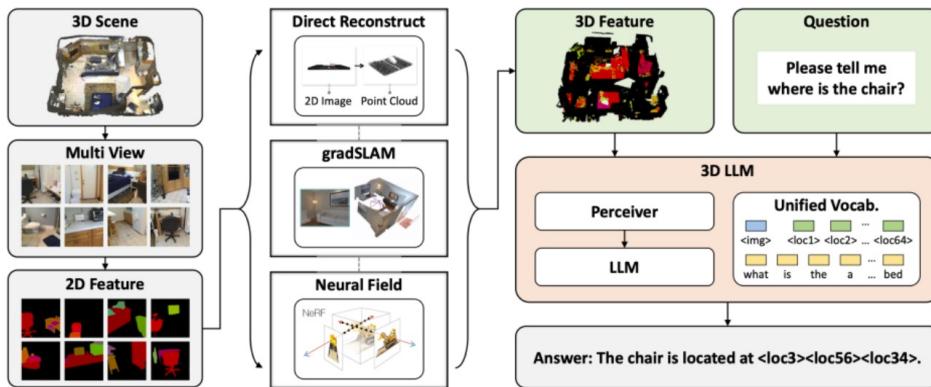
[1] Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. 2023

[2] Video-LLaVA: Learning United Visual Representation by Alignment Before Projection. 2023

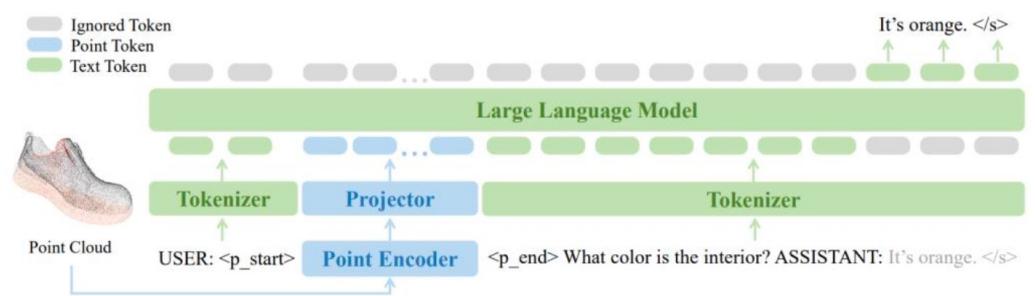
[3] Video Understanding with Large Language Models: A Survey. <https://github.com/yunlong10/Awesome-LLMs-for-Video-Understanding>, 2023

MLLM Functionality: 3D Perception

+ 3D-LLM



+ PointLLM

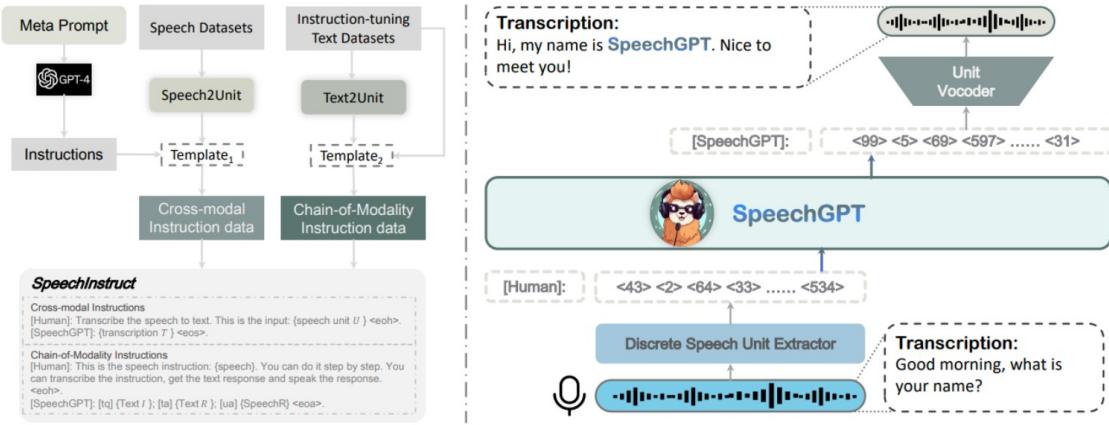


[1] 3D-LLM: Injecting the 3D World into Large Language Models. 2023

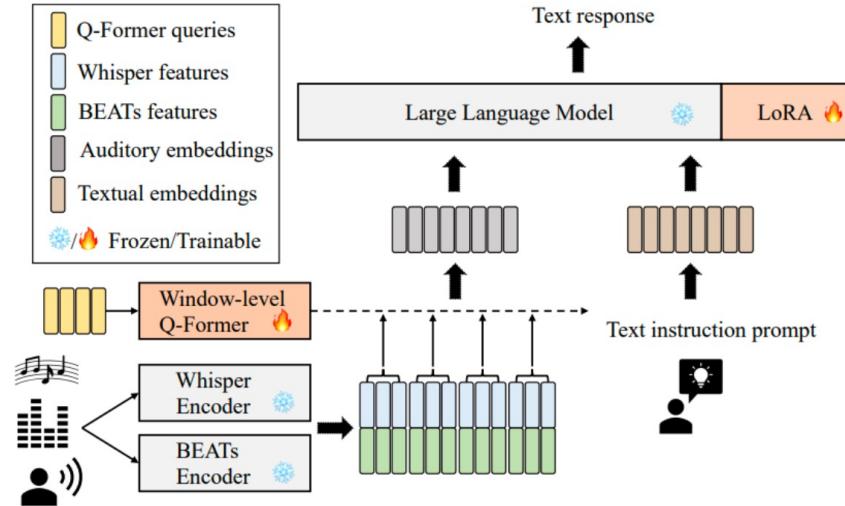
[2] PointLLM: Empowering Large Language Models to Understand Point Clouds. 2023

MLLM Functionality: Audio Perception

SpeechGPT



SALMONN



[1] SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities. 2023

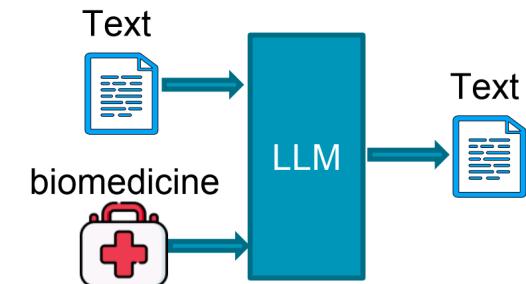
[2] SALMONN: Towards Generic Hearing Abilities for Large Language Models. 2023

[3] Sparks of Large Audio Models: A Survey and Outlook. <https://github.com/EmulationAI/awesome-large-audio-models> , 2023

MLLM Functionality: BioMedical Perception

+ Bio-/Medical & Healthcare

- | | | |
|----------------|---------------|-------------|
| + BioGPT | + DoctorGLM | + MedAlpaca |
| + DrugGPT | + BianQue | + AlpaCare |
| + BioMedLM | + ClinicalGPT | + Zhongjing |
| + OphGLM | + Qilin-Med | + PMC-LLaMA |
| + GatorTron | + ChatDoctor | + CPLLM |
| + GatorTronGPT | + BenTsao | + MedPaLM 2 |
| + MEDITRON | + HuatuoGPT | + BioMedGPT |

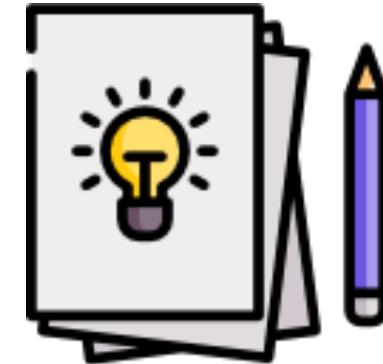


- [1] BioGPT: Generative Pre-trained Transformer for Biomedical Text Generation and Mining. 2022
 - [2] DrugGPT: A GPT-based Strategy for Designing Potential Ligands Targeting Specific Proteins. 2023
 - [3] MEDITRON-70B: Scaling Medical Pretraining for Large Language Models. 2023
 - [4] HuaTuo: Tuning LLaMA Model with Chinese Medical Knowledge. 2023
 - [5] AlpaCare: Instruction-tuned Large Language Models for Medical Application. 2023
 - [6] A Survey of Large Language Models in Medicine: Progress, Application, and Challenge, <https://github.com/AI-in-Health/MedLLMsPracticalGuide>. 2023.
- Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

MLLM Functionality: Generation

Often, MLLMs need to not only understand the input multimodal information, but also to generate information in that modality.

- Image Captioning
- Visual Question Answering
- Text-to-Vision Synthesis
- Vision-to-Vision Translation
- Scene Text Recognition
- Scene Text Inpainting
- ...

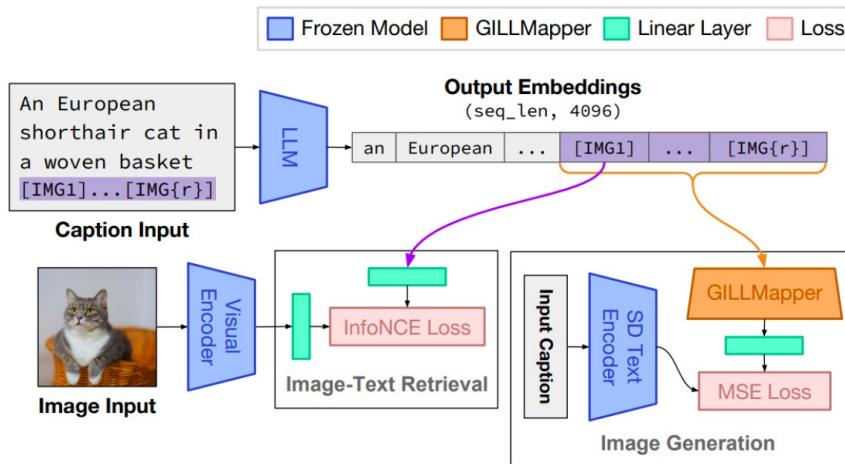


MLLM Functionality: Generation

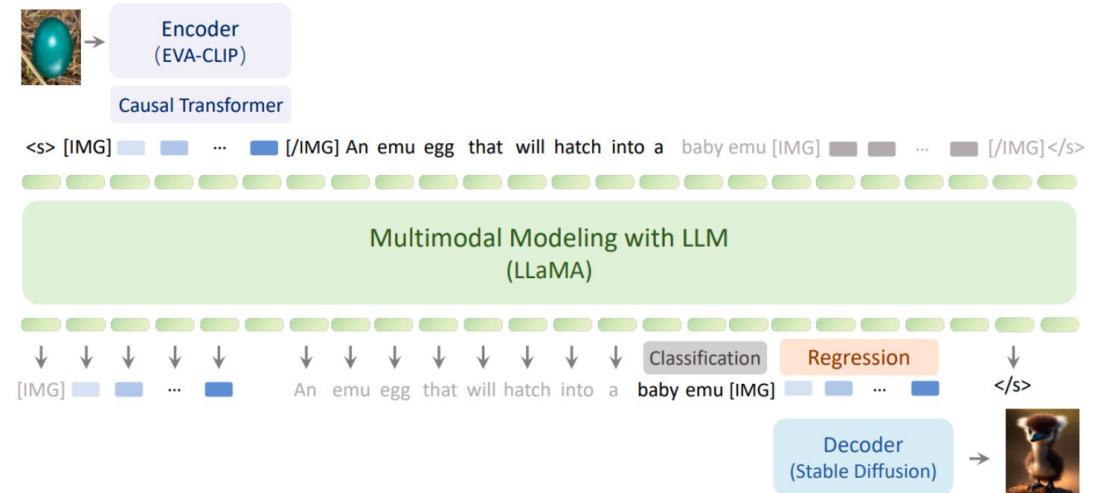
	Modality (w/ Language)			
	Image	Video	Audio	3D
Input-side Perceiving	Flamingo, Kosmos-1, Blip2, mPLUG-Owl, Mini-GPT4, LLaVA, InstructBLIP, VPGTrans, CogVLM, Monkey, Chameleon, Otter, Qwen-VL, GPT-4v, SPHINX, Yi-VL, Fuyu, ...	VideoChat, Video-ChatGPT, Video-LLaMA, PandaGPT, MovieChat, Video-LLaVA, LLaMA-VID, Momentor, ...	AudioGPT, SpeechGPT, VIOLA, AudioPaLM, SALMONN, MU-LLaMA, ...	3D-LLM, 3D-GPT, LL3DA, SpatialVLM, PointLLM, PointBind, ...
	[Pixel-wise] GPT4RoI, LION, MiniGPT-v2, NErT-Chat, Kosmos-2, GLaMM, LISA, DetGPT, Osprey, PixelLM, ...	[Pixel-wise] PG-Video-LLaVA, Merlin, MotionEpic, ...	-	-
	Video-LLaVA, Chat-UniVi, LLaMA-VID		-	-
	Panda-GPT, Video-LLaMA, AnyMAL, Macaw-LLM, Gemini, VideoPoet, ImageBind-LLM, LLMBind, LLaMA-Adapter, ...			-
Perceiving + Generating	GILL, EMU, MiniGPT-5, DreamLLM, LLaVA-Plus, InternLM-XComposer2, SEED-LLaMA, LaVIT, Mini-Gemini, ...	GPT4Video, Video-LaVIT, VideoPoet, ...	AudioGPT, SpeechGPT, VIOLA, AudioPaLM, ...	-
	[Pixel-wise] Vitron		-	-
	NExT-GPT, Unified-IO 2, AnyGPT, CoDi-2, Modaverse, ViT-Lens, ...			-

MLLM Functionality: Image Generation

+ GILL



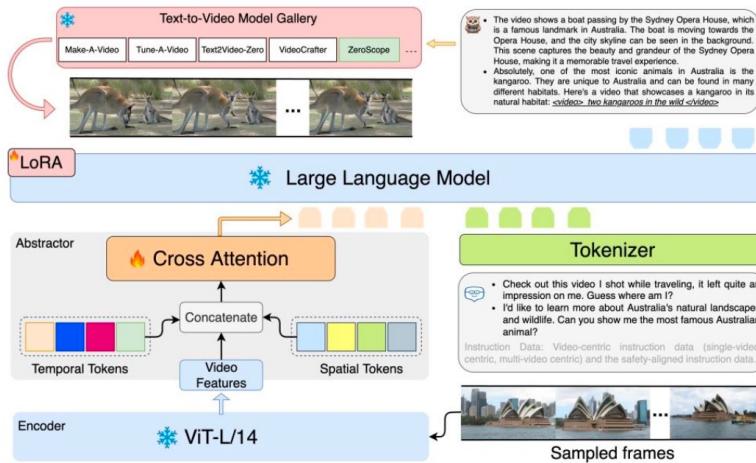
+ EMU



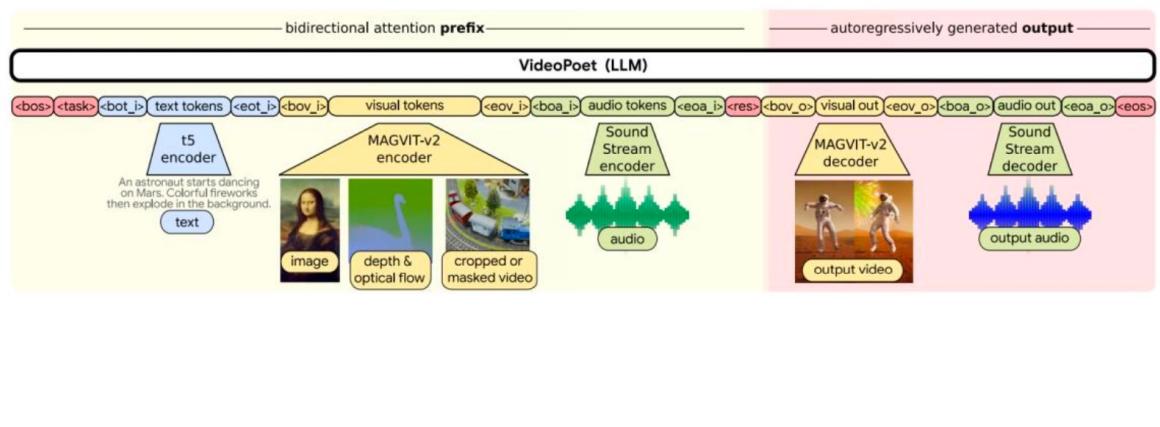
- [1] Generating Images with Multimodal Language Models. 2023
- [2] Generative Pretraining in Multimodality. 2023

MLLM Functionality: Video Generation

GPT4Video



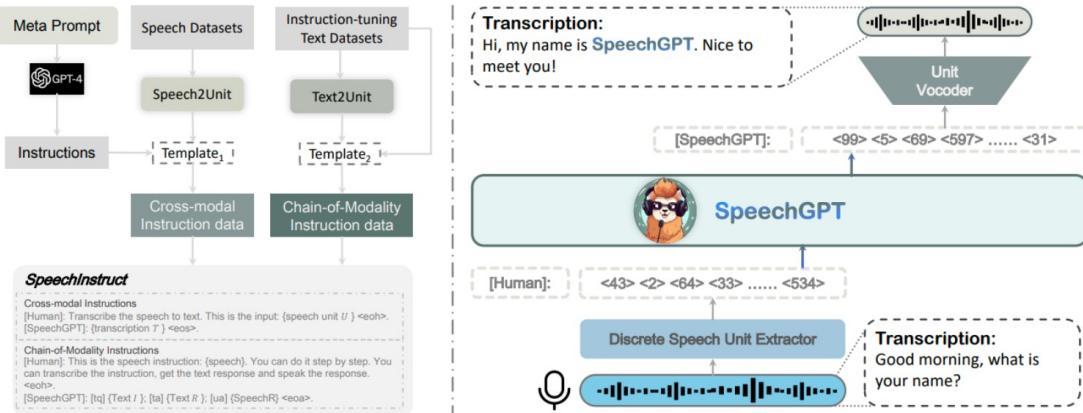
VideoPoet



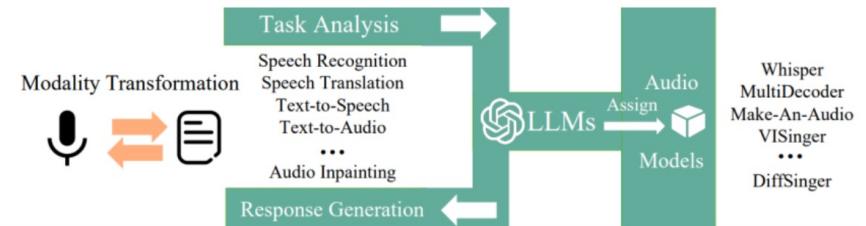
- [1] GPT4Video: A Unified Multimodal Large Language Model for Instruction-Followed Understanding and Safety-Aware Generation. 2023
- [2] VideoPoet: A Large Language Model for Zero-Shot Video Generation. 2023

MLLM Functionality: Audio Generation

SpeechGPT



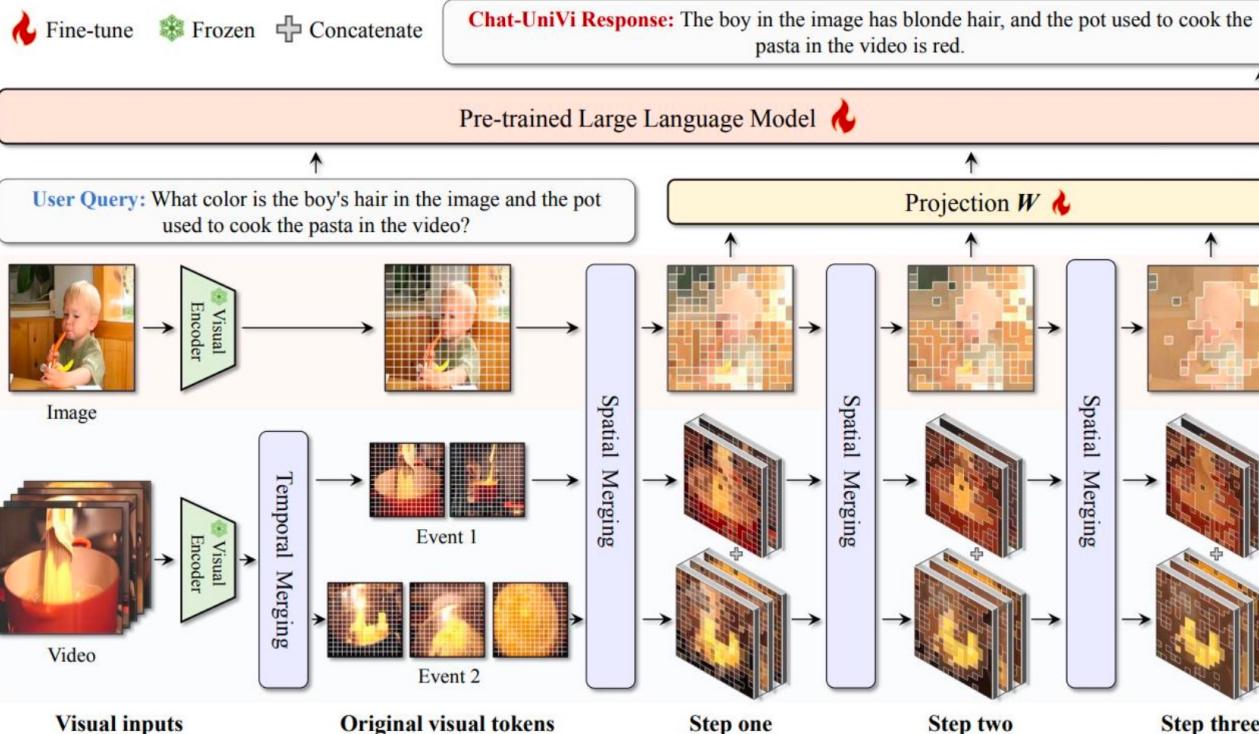
AudioGPT



- [1] SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities. 2023
- [2] AudioGPT: Understanding and Generating Speech, Music, Sound, and Talking Head. 2023

MLLM Functionality: Text+Image+Video Generation

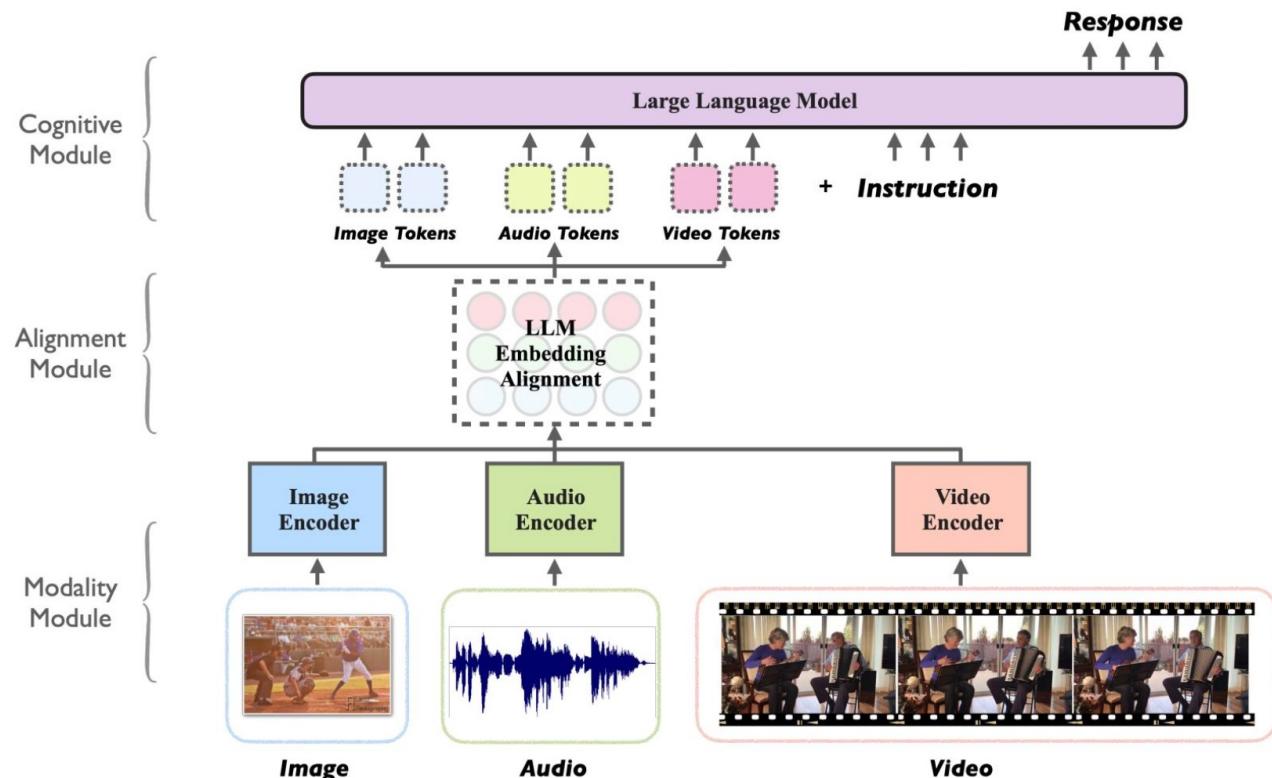
 Chat-UniVi



[1] Chat-UniVi: Unified Visual Representation Empowers Large Language Models with Image and Video Understanding. 2023

MLLM Functionality: Text+Image+Video+Audio Generation

Macaw-LLM



[1] Macaw-LLM: Multi-Modal Language Modeling with Image, Audio, Video, and Text Integration. 2023

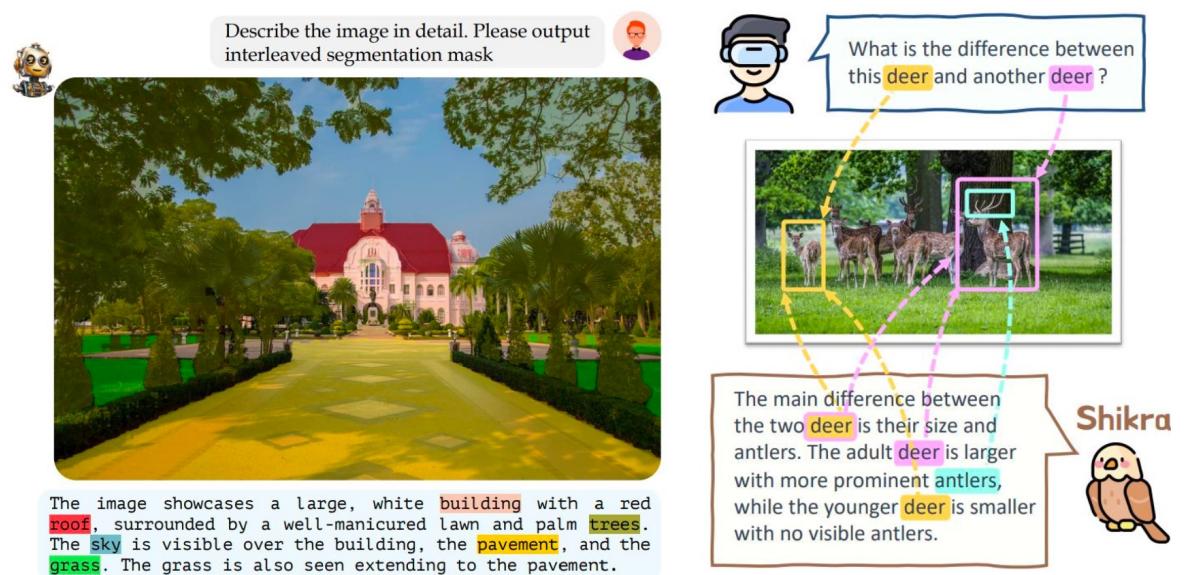
Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

MLLM Functionality: Fine-grained Capabilities

[Pixel-level Vision MLLM]

The vision MLLMs described above generally only support coarse-grained, instance-level visual understanding. This can lead to **imprecise visual interpretations**. Also due to the lack of visual grounding, these MLLMs will potentially **produce hallucinations**.

- Visual Grounding
- Visual Segmentation
- Visual Editing
- Visual Inpainting



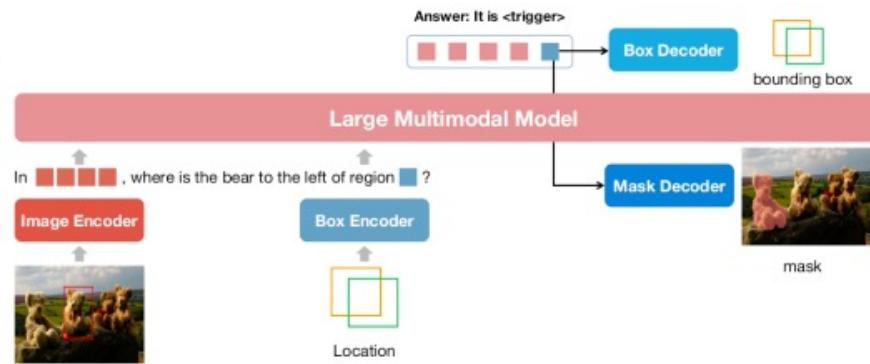
MLLM Functionality: Fine-grained Capabilities

	Modality (w/ Language)			
	Image	Video	Audio	3D
Input-side Perceiving	Flamingo, Kosmos-1, Blip2, mPLUG-Owl, Mini-GPT4, LLaVA, InstructBLIP, VPGTrans, CogVLM, Monkey, Chameleon, Otter, Qwen-VL, GPT-4v, SPHINX, Yi-VL, Fuyu, ... [Pixel-wise] GPT4RoI, LION, MiniGPT-v2, NExT-Chat, Kosmos-2, GLaMM, LISA, DetGPT, Osprey, PixelLM, ... Video-LLaVA, Chat-UniVi, LLaMA-VID	VideoChat, Video-ChatGPT, Video-LLaMA, PandaGPT, MovieChat, Video-LLaVA, LLaMA-VID, Momentor, ... [Pixel-wise] PG-Video-LLaVA, Merlin, MotionEpic, ...	AudioGPT, SpeechGPT, VIOLA, AudioPaLM, SALMONN, MU-LLaMA, ... -	3D-LLM, 3D-GPT, LL3DA, SpatialVLM, PointLLM, PointBind, ... -
Perceiving + Generating	GILL, EMU, MiniGPT-5, DreamLLM, LLaVA-Plus, InternLM-XComposer2, SEED-LLaMA, LaVIT, Mini-Gemini, ... [Pixel-wise] Vitron NExT-GPT, Unified-IO 2, AnyGPT, CoDi-2, Modaverse, ViT-Lens, ...	GPT4Video, Video-LaVIT, VideoPoet, ... -	AudioGPT, SpeechGPT, VIOLA, AudioPaLM, ... -	-

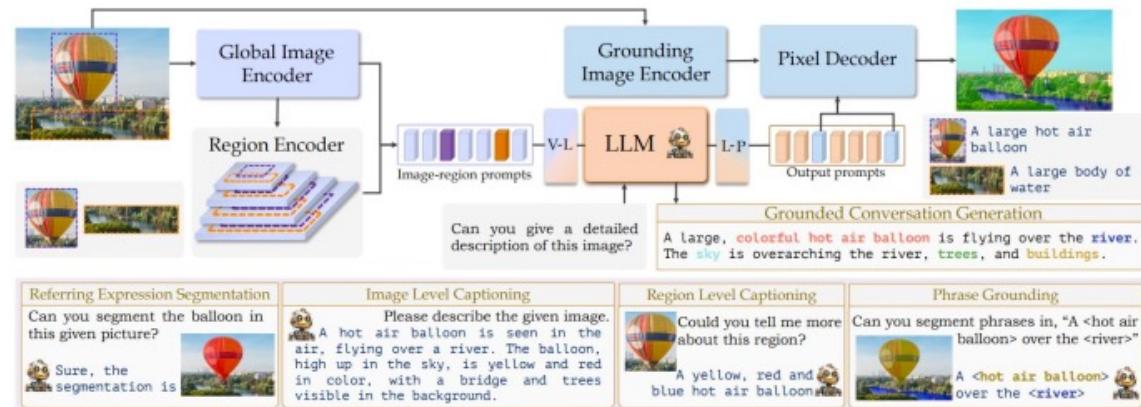
MLLM Functionality: Image Fine-grained Capabilities

- Image-oriented Pixel-wise

+ NExT-Chat



+ GLaMM



[1] NExT-Chat: An LMM for Chat, Detection and Segmentation. 2023

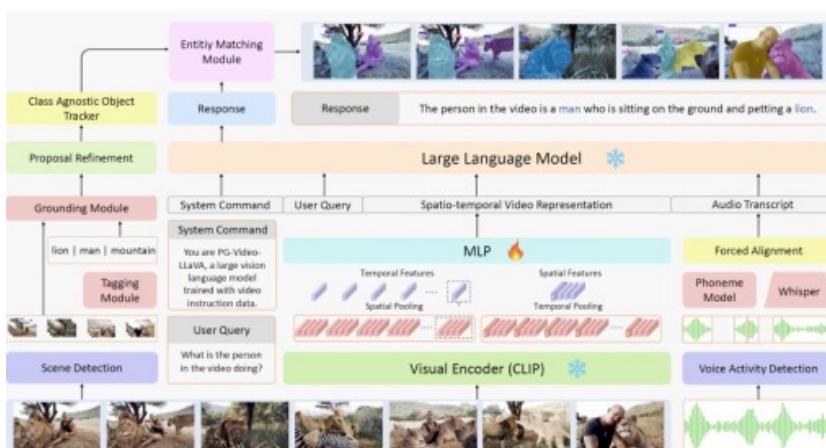
[2] GLaMM: Pixel Grounding Large Multimodal Model. 2023

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

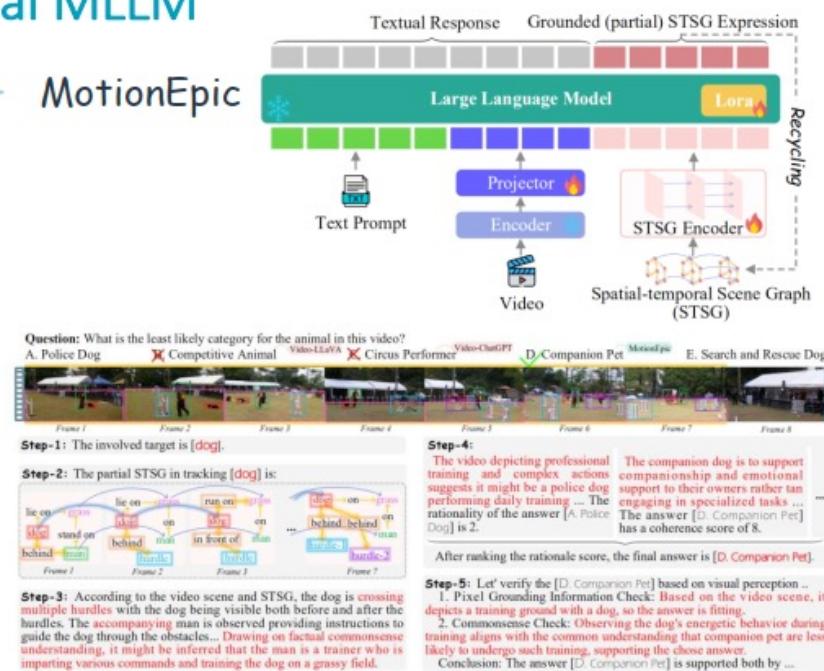
MLLM Functionality: Video Fine-grained Capabilities

- Video-oriented Pixel-wise Regional MLLM

+ PG-Video-LLaVA



+ MotionEpic



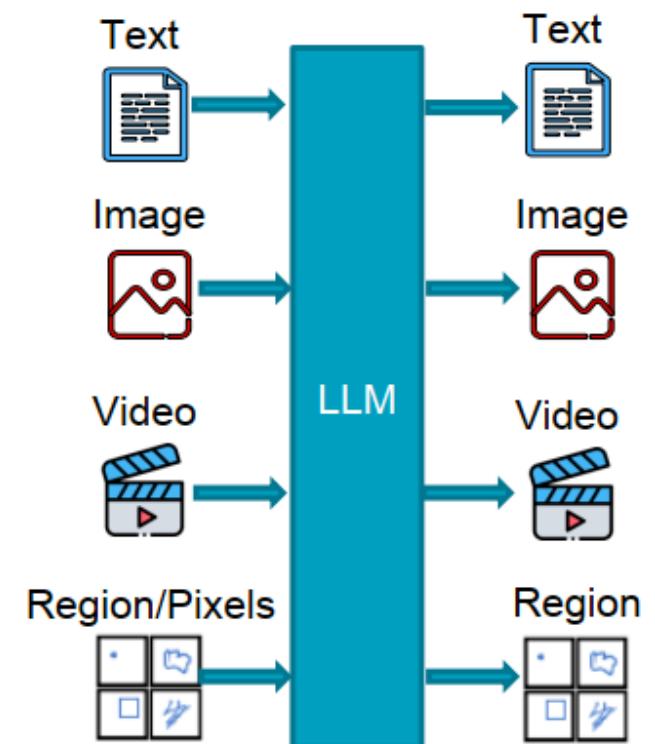
[1] PG-Video-LLaVA: Pixel Grounding in Large Multimodal Video Models. 2023

[2] Video-of-Thought: Step-by-Step Video Reasoning from Perception to Cognition. 2024

MLLM Functionality: Unified Fine-grained Capabilities

[Vitron]

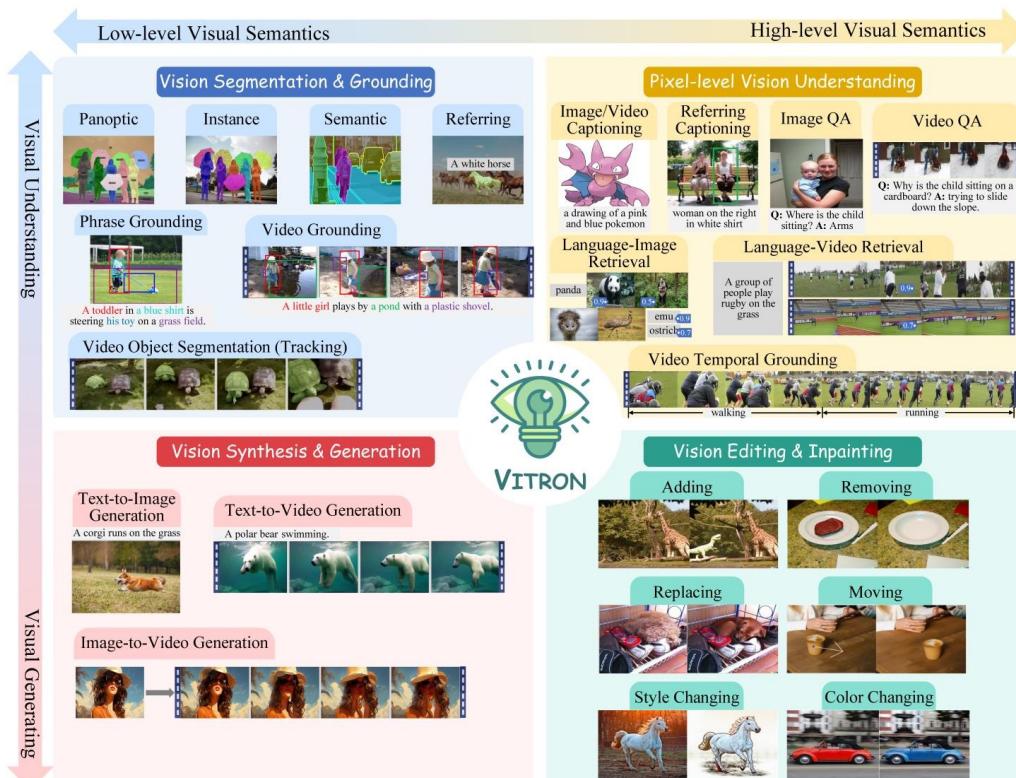
Users input either an image or video (potentially specifying a region), and the LLM outputs content based on its understanding, generating, grounding or tracking the content to specific pixel-level regions of the image, video.



[1] VITRON: A Unified Pixel-level Vision LLM for Understanding, Generating, Segmenting, Editing. 2024

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

MLLM Functionality: Unified Fine-grained Capabilities



Model	Vision Supporting		Pixel/Regional Understanding	Segmenting/Grounding	Generating	Editing
	Image	Video				
Flamingo [1]	✓	✗	✗	✗	✗	✗
BLIP-2 [45]	✓	✗	✗	✗	✗	✗
MiniGPT-4 [126]	✓	✗	✗	✗	✗	✗
LLaVA [57]	✓	✗	✗	✗	✗	✗
GILL [39]	✓	✗	✗	✗	✗	✗
Emu [90]	✓	✗	✗	✗	✗	✗
MiniGPT-5 [125]	✓	✗	✗	✗	✗	✗
DreamLLM [23]	✓	✗	✗	✗	✗	✗
GPT4Rof [122]	✓	✗	✓	✓	✗	✗
NExT-Chat [118]	✓	✗	✓	✓	✗	✗
MiniGPT-v2 [13]	✓	✗	✓	✓	✗	✗
Shikra [14]	✓	✗	✓	✓	✗	✗
Kosmos-2 [72]	✓	✗	✓	✓	✗	✗
GLaMM [78]	✓	✗	✓	✓	✗	✗
Osprey [117]	✓	✗	✓	✓	✗	✗
PixelLM [79]	✓	✗	✓	✓	✗	✗
LLaVA-Plus [58]	✓	✗	✗	✓	✓	✓
VideoChat [46]	✗	✓	✗	✗	✗	✗
Video-LLaMA [120]	✗	✓	✗	✗	✗	✗
Video-LLaVA [52]	✓	✓	✗	✗	✗	✗
Video-ChatGPT [61]	✗	✓	✗	✗	✗	✗
GPT4Video [99]	✗	✓	✗	✗	✓	✗
PG-Video-LLaVA [67]	✗	✓	✓	✓	✗	✗
NExT-GPT [104]	✓	✓	✗	✓	✓	✗
VITRON (Ours)	✓	✓	✓	✓	✓	✓

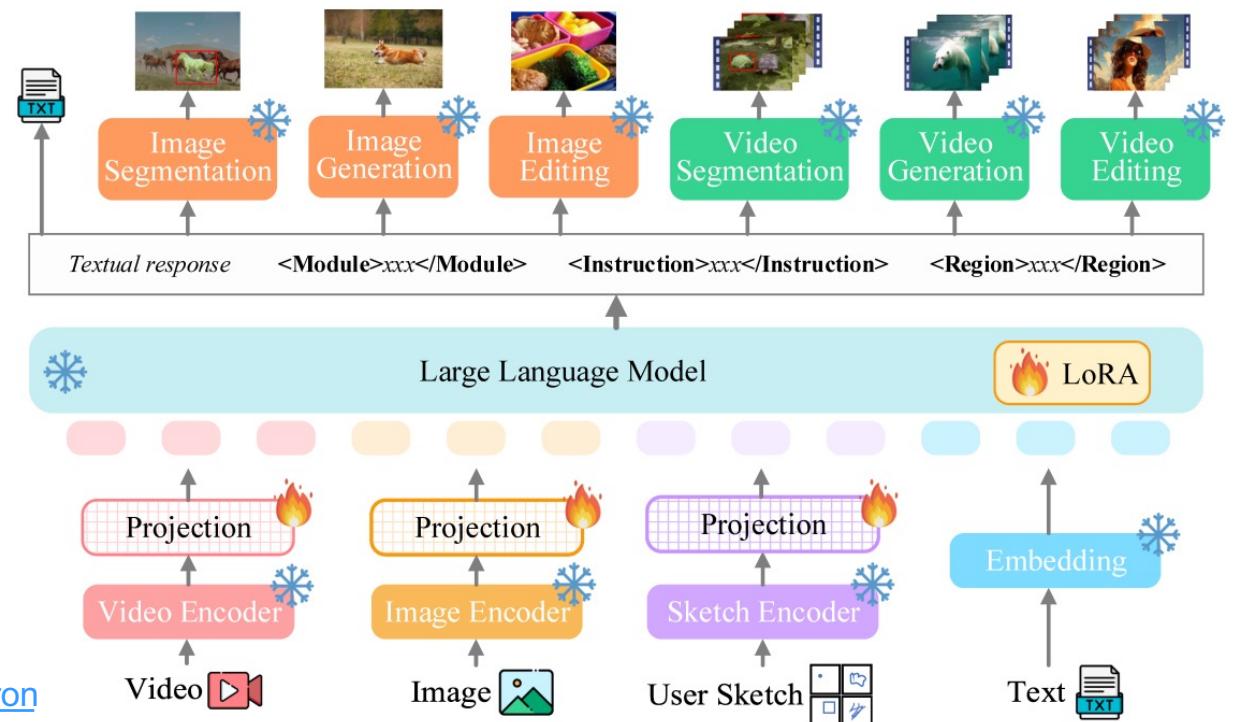
[1] VITRON: A Unified Pixel-level Vision LLM for Understanding, Generating, Segmenting, Editing. 2024

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

MLLM Functionality: Unified Fine-grained Capabilities



Project: <https://vitron-llm.github.io/>
 Paper: <https://is.gd/aGu0VV>
 Code&Demo: <https://github.com/SkyworkAI/Vitron>



[1] VITRON: A Unified Pixel-level Vision LLM for Understanding, Generating, Segmenting, Editing. 2024

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

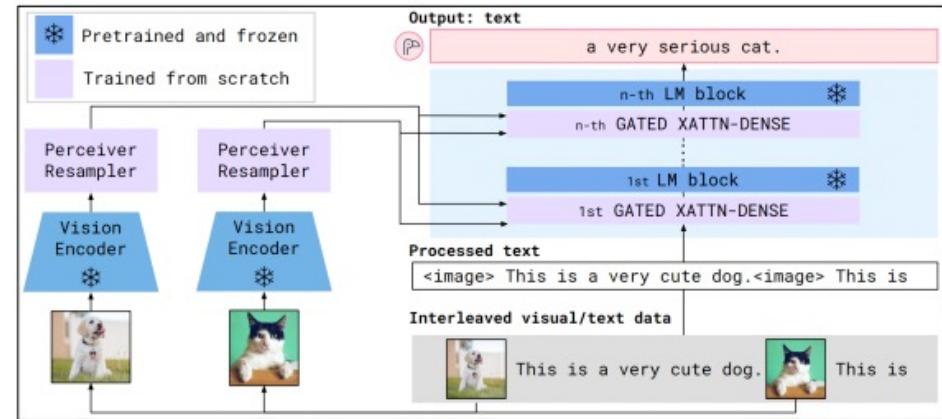
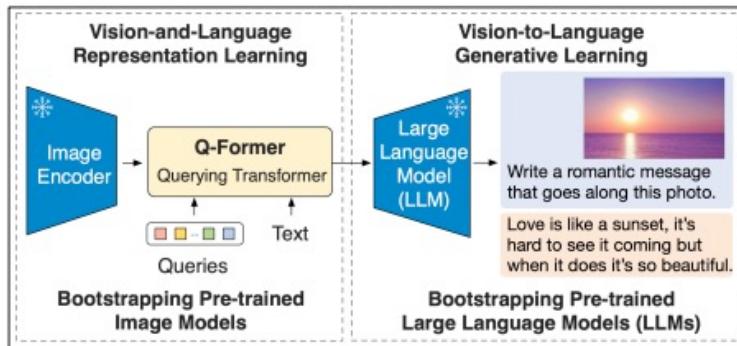
Plan for this lecture

1. Intro to Multimodal Large Language Models (MLLM)
2. Architectures
3. Encodings
4. Functionality
 - Contrastive Learning Models
 - Generative Models
5. Multimodal Instruction Tuning
6. What's Next?

Multimodal Instruction Tuning

Why Multimodal Instruction Tuning?

Multimodal Instruction Tuning



- Pretrained models aligns multiple modalities, can understand basic information from different modalities, and sometimes perform simple question-answering.
- Cannot follow complex instructions, and often require **task-specific** fine-tuning for it to perform well on downstream tasks.

[Wang et al. 2022] GIT: A Generative Image-to-text Transformer for Vision and Language

[Li et al. 2023] Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models

[Alayrac et al. 2022] Flamingo: a visual language model for few-shot learning

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

Why Multimodal Instruction Tuning?

From Single-Purpose to General-Purpose

- Traditional vision models are **task-specific**, which requires training and using multiple models for different tasks and **restrict the potential synergies from diverse tasks**;
- These vision models typically have a pre-defined and fixed interface, leading to **limited interactivity and adaptability in following users' task instructions**.
- Multimodal Instruction Tuning allows multimodal models to **generalize to unseen tasks by following new instructions**, thus boosting **zero-shot performance**.



Instruction Tuning is NOT multitask learning

Multitask learning (with task tokens)

Training

INPUT: <image><tok_task_1=short_cap>
OUTPUT: <generated short descriptions>

INPUT: <image><tok_task_2=yes_no>
OUTPUT: yes/no

Testing

Only with <tok_task_1>, <tok_task_2>...

Does not work with <new_task=long_cap>

Instruction tuning (with natural language task instructions)

Training

INPUT: <image>Describe this image
briefly.
OUTPUT: <generated short descriptions>

INPUT: <image>Is this xxx?
OUTPUT: yes/no

Testing

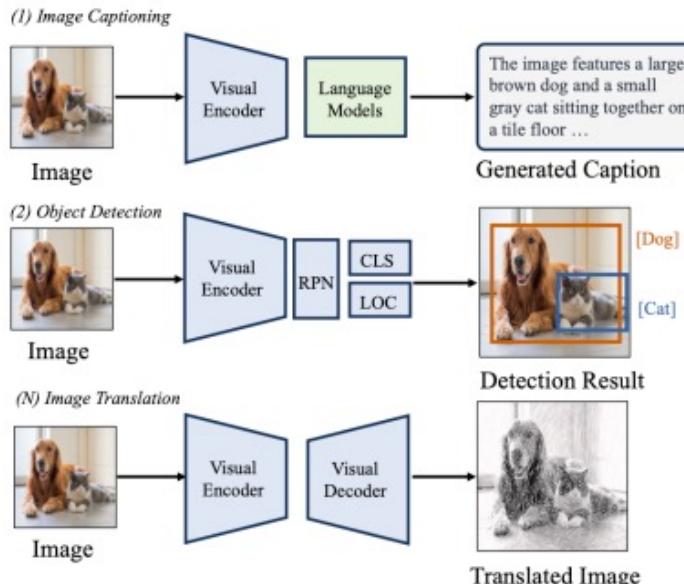
INPUT: <image>Describe this image in detail.
OUTPUT: <long descriptions>

Generalizes to new instructions zero-shot.

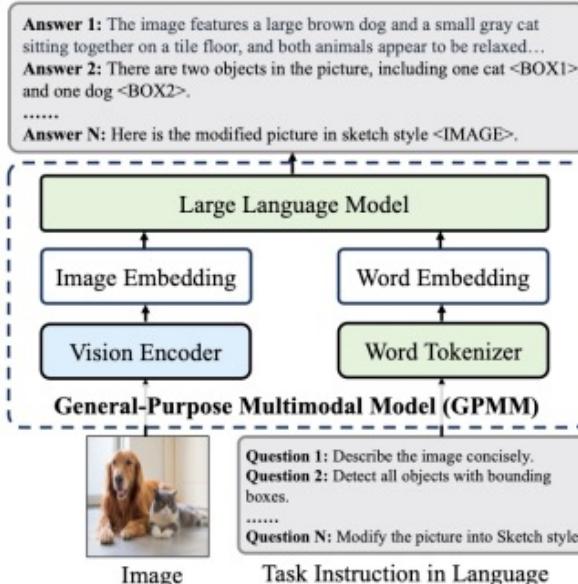
Why Multimodal Instruction Tuning?

From Single-Purpose to General-Purpose

(a). Traditional Task Paradigm for Computer Vision



(b). Instruction-based Task Paradigm for Computer Vision



[1] Visual Instruction Tuning towards General-Purpose Multimodal Model: A Survey. 2023

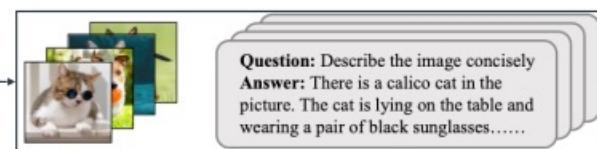
[2] A Survey on Multimodal Large Language Models. 2024

MLLM Instruction Tuning Framework

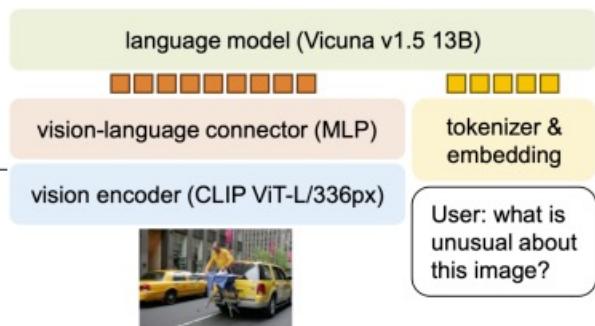
Data Construction



Visual Instruction-following data



Visual Instruction Tuning Framework Example: LLaVA-1.5

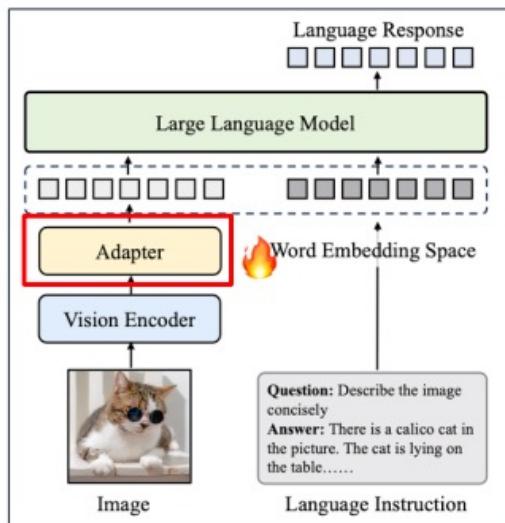


Popular MLLMs: LLaVA, MiniGPT4, LLaVA-NeXT, ViP-LLaVA, LLaVA-UHD, MiniCPM, Qwen-VL, CogAgent, InternVL, mPLUG-OWL, Monkey, MiniGemini, LLaVA-HR, SPHINX, DeepSeek-VL, MoAI

Training Paradigms

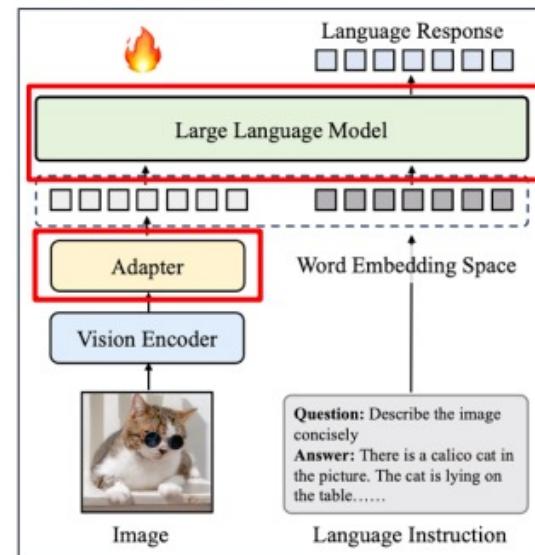
Stage1: Pretraining Stage

- + Align different modalities, provide world knowledge



Stage2: Instruction Tuning Stage

- + Teach models to better understand the instructions from users and fulfill the demanded tasks.



[1] MMC: Advancing Multimodal Chart Understanding with Large-scale Instruction Tuning. NAACL 2024.

[2] Visual Instruction Tuning. NeurIPS 2023.

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

Multimodal learning as a translation problem

Instruction

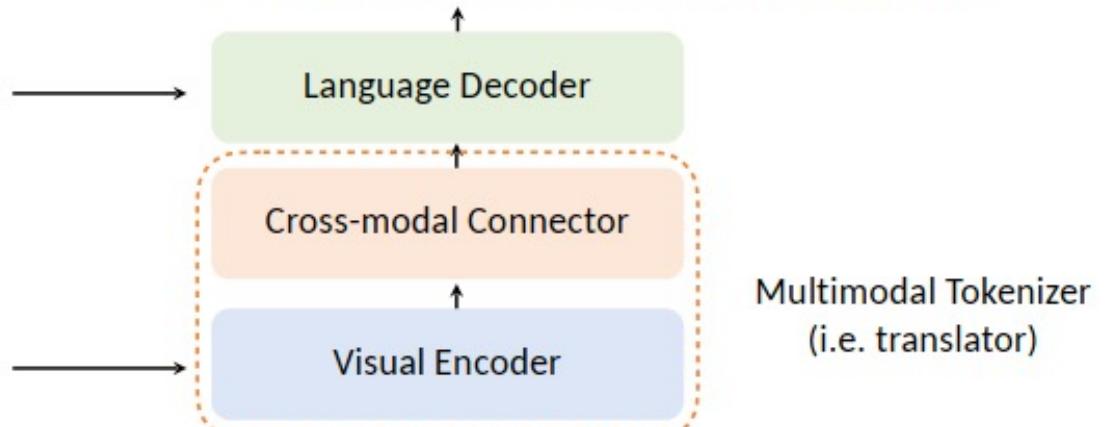
What's the color of the shirt that the man is wearing?

Image



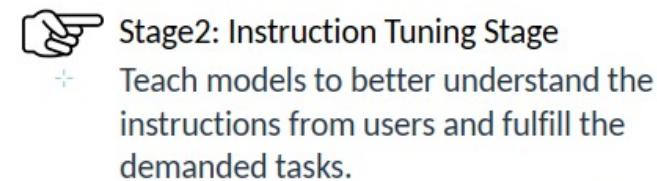
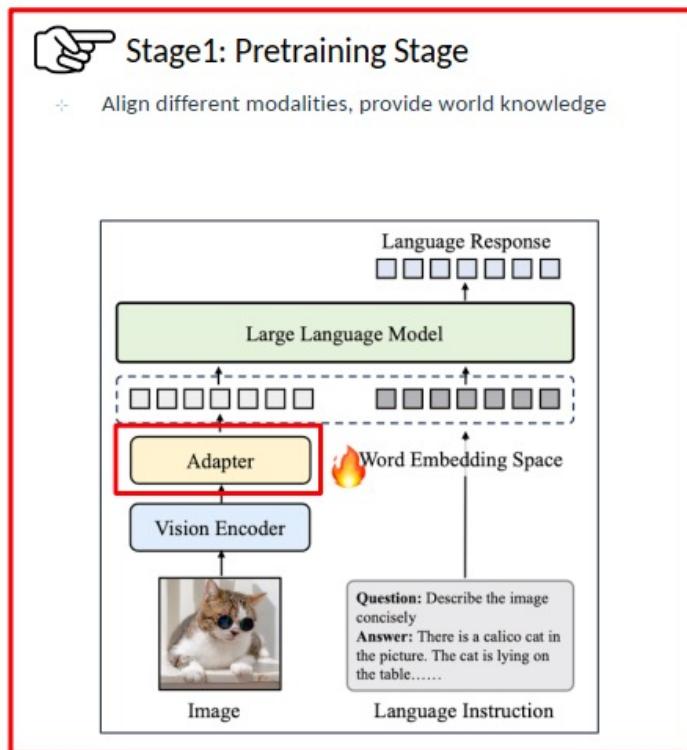
Output

The color of the shirt the man's wearing is yellow.



LLM “learns” a visual foreign language efficiently.

Multimodal Instruction Tuning Data Generation - Pretraining Data



[1] MMC: Advancing Multimodal Chart Understanding with Large-scale Instruction Tuning. NAACL 2024.

[2] Visual Instruction Tuning. NeurIPS 2023.

Multimodal Instruction Tuning Data Generation - Pretraining Data

[Coarse-gained Image-text]

Data volume is **large**, the captions are **shorts** and **noisy**.

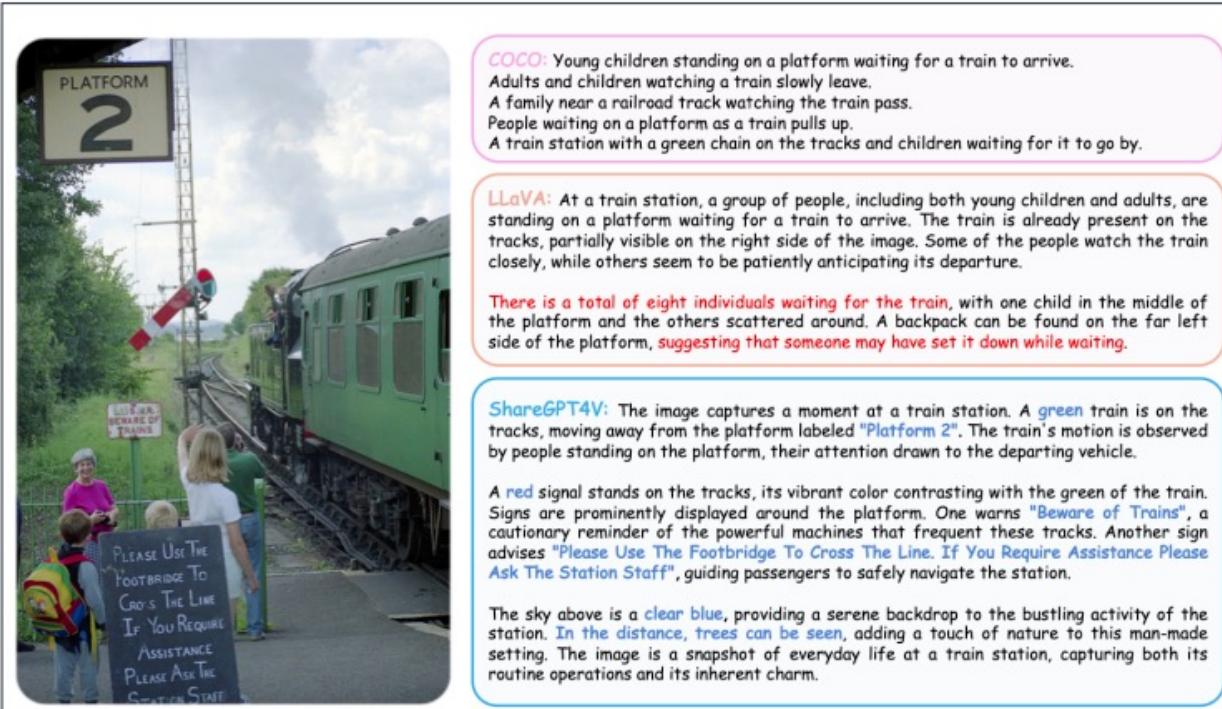
[Fine-gained Image-Text]

High quality, **longer** and **more accurate descriptions**, fine-gained alignment between different modalities.

Dataset	Samples	Taxonomies
CC-3M	3.3M	Coarse-grained Image-Text
CC-12M	12.4M	Coarse-grained Image-Text
LAION-5B	5.9B	Coarse-grained Image-Text
SBU-Captions	1M	Coarse-grained Image-Text
ShareGPT4V-PT	1.2M	Fine-grained Image-Text
LVIS-Instruct4V	111k	Fine-grained Image-Text
<u>ALLaVA</u>	709k	Fine-grained Image-Text
MSR-VTT	200k	Video-Text
<u>WavCaps</u>	24k	Audio-Text
MMC-Instruction	600k	Chart-Text

[1] A Survey on Multimodal Large Language Models. 2024

Multimodal Instruction Tuning Data Generation - Pretraining Data



Coarse-gained Image-text

Coarse-gained Image-text

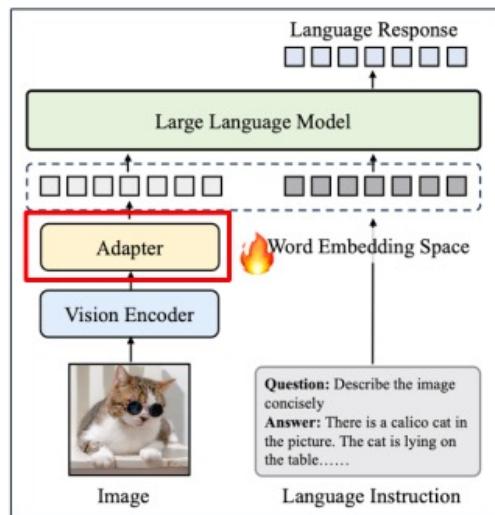
Fine-gained Image-text

[1] ShareGPT4V: Improving Large Multi-Modal Models with Better Captions. 2023.

Multimodal Instruction Tuning Data Generation - Instruction Data Generation

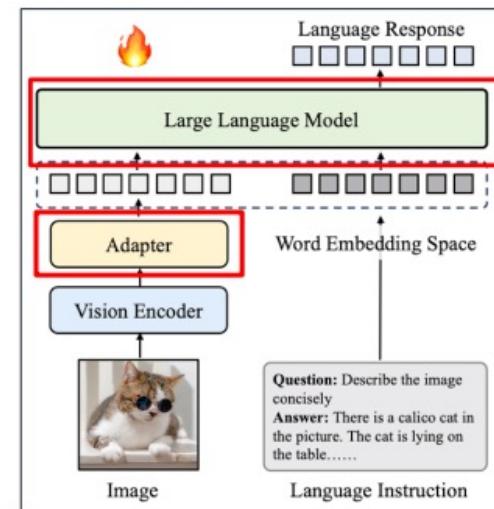
Stage1: Pretraining Stage

+ Align different modalities, provide world knowledge



Stage2: Instruction Tuning Stage

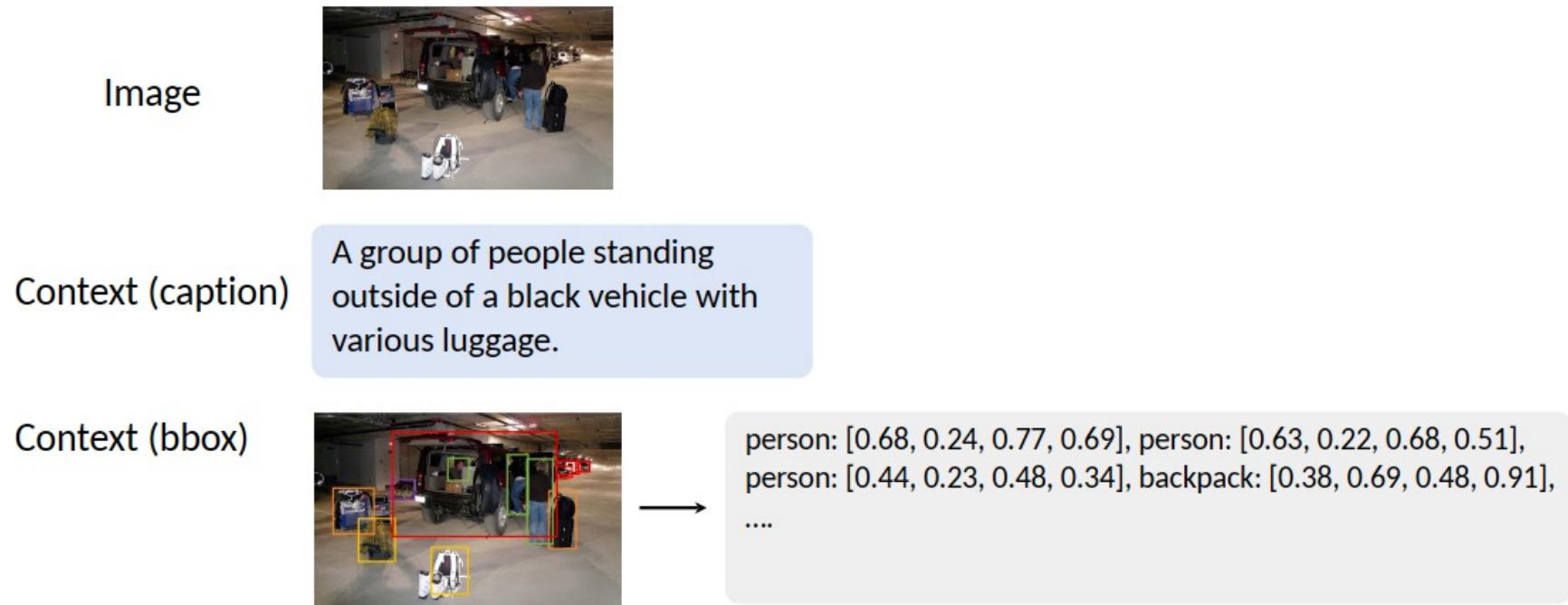
+ Teach models to better understand the instructions from users and fulfill the demanded tasks.



[1] MMC: Advancing Multimodal Chart Understanding with Large-scale Instruction Tuning. NAACL 2024.

[2] Visual Instruction Tuning. NeurIPS 2023.

Multimodal Instruction Tuning Data Generation - Instruction Data Generation

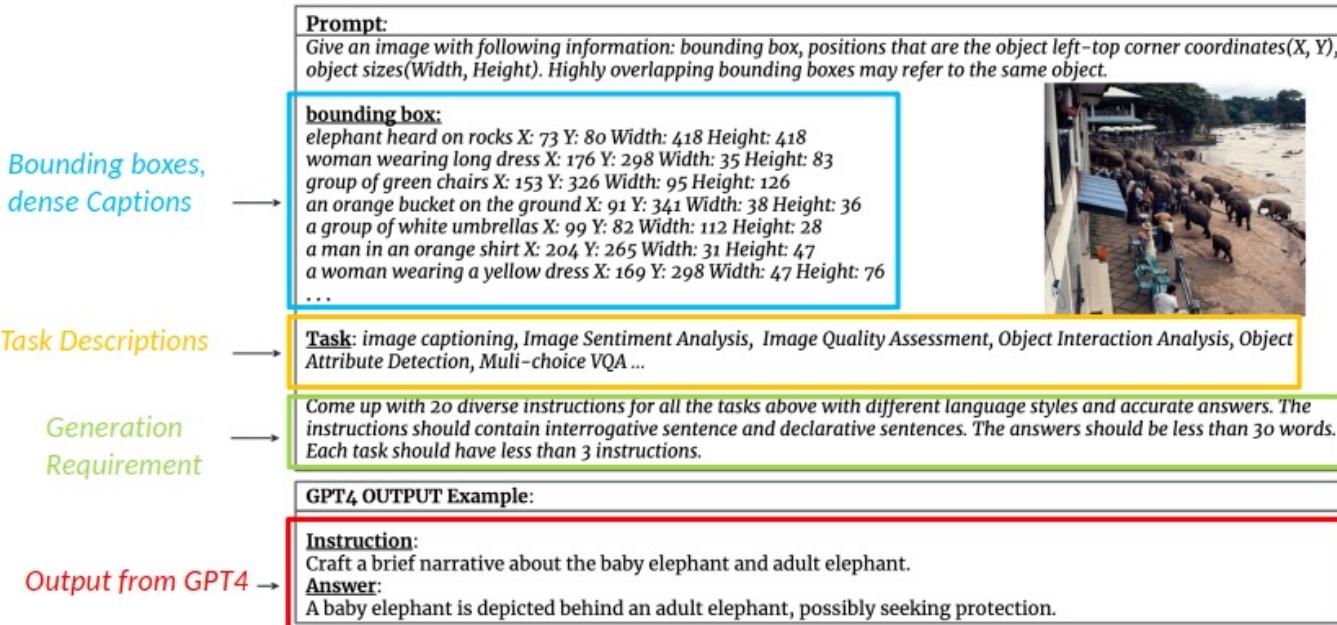


[1] Visual Instruction Tuning. NeurIPS 2023.

Multimodal Instruction Tuning Data Generation - Instruction Data Generation

Self Instruction

First, Translate images into **dense captions and bounding boxes**. Second, prompt **text-only GPT-4**.



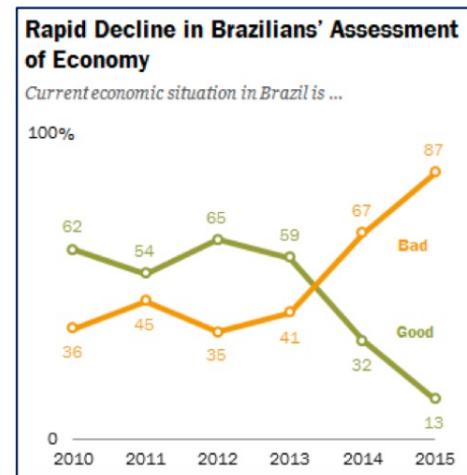
[1] Aligning large multi-modal model with robust instruction tuning. ICLR 2024.

Multimodal Large Language Models (MLLM) – Tutorial - <https://mllm2024.github.io/CVPR2024/>

Multimodal Instruction Tuning Data Generation - Instruction Data Generation

❖ Existing Data

The answers of existing VQA and caption datasets are usually **concise**, directly using these datasets for instruction tuning **may limit the output length of MLLMs**.



Question:

Which year has the most divergent opinions about Brazil's economy?

Answer:

2015

[1] ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. ACL 2022 Findings.

Multimodal Instruction Tuning Data Generation - Instruction Data Generation

Visual input example, Different Format Prompts:



Normal prompt

What is the color of the shirt that the man is wearing?

Response

The man is wearing a yellow shirt.

Instruction Tuning Dataset

Dataset	Size	Modalities	Constructions
LLaVA-Instruct-158k	158k	Image, Text	ChatGPT-generated
LRV-Instruction	400k	Image, Text	GPT4-generated
MMC-Instruction	600k	Chart, Text	GPT4-generated/adapted
Clotho-Detail	3.9k	Text, Audio	GPT4-generated
MACAW-LLM	119k	Image, Video, Text	GPT-3.5-turbo-generated
MIMIC-IT	2.8M	Image, Video, Text	ChatGPT-generated
StableLLaVA	126k	Image, Text	StableDiffusion & ChatGPT-generated
LAMM	196k	Image, PointCloud, Text	GPT4-generated
VIGC-LLaVA	1.8M	Image, Text	Model-generated
X-LLM	10k	Image, Video, Text	ChatGPT-generated

Summary

[How we teach multimodal models]

- **Pretraining:**
A dictionary to teach LLM to understand (vocabularies from) a new modality
- **Instruction tuning (short answer VQA):**
Small puzzles to effectively/efficiently injects new domain knowledge
- **Instruction tuning (natural conversation VQA):**
Real-world applications to practice the skills



Plan for this lecture

1. Intro to Multimodal Large Language Models (MLLM)
2. Architectures
3. Encodings
4. Functionality
 - Contrastive Learning Models
 - Generative Models
5. Multimodal Instruction Tuning
6. What's Next?

Tutorials

- Introduction to Vision-Language Models in Python [[link](#)]
- Vision-Language Models [[link](#)]



Visual Programming (Gupta & Kembhavi, CVPR 2023)

Compositional Visual Question Answering

IMAGE: 

Question: Are there both ties and glasses in the picture?

Program:

```
BOX0=Loc(image=IMAGE, object='ties')
ANSWER0=Count(box=BOX0)
BOX1=Loc(image=IMAGE, object='glasses')
ANSWER1=Count(box=BOX1)
ANSWER2=Eval("yes" if {ANSWER0} > 0 and {ANSWER1} > 0 else "no")
RESULT=ANSWER2
```

Prediction: no

Natural Language Visual Reasoning

LEFT: 

RIGHT: 

Statement: The left and right image contains a total of six people and two boats.

Program:

```
ANSWER0=Vqa(image=LEFT, question='How many people are in the image?')
ANSWER1=Vqa(image=RIGHT, question='How many people are in the image?')
ANSWER2=Vqa(image=LEFT, question='How many boats are in the image?')
ANSWER3=Vqa(image=RIGHT, question='How many boats are in the image?')
ANSWER4=Eval("{ANSWER0} + {ANSWER1} == 6 and {ANSWER2} + {ANSWER3} == 2")
RESULT=ANSWER4
```

Prediction: False

Factual Knowledge Object Tagging

IMAGE: 

Prediction: 

Instruction: Tag the 7 main characters on the TV show Big Bang Theory

Program:

```
OBJ0=FaceDet(image=IMAGE)
LIST0=List(query='main characters on the TV show Big Bang Theory', max=7)
OBJ1=Classify(image=IMAGE, object=OBJ0, categories=LIST0)
IMAGE0=Tag(image=IMAGE, object=OBJ1)
RESULT=IMAGE0
```

Natural Language Image Editing

IMAGE: 

Prediction: IMAGE1

IMAGE: 

Instruction: Hide Daniel Craig with 8) and Sean Connery with ;

Program:

```
OBJ0=FaceDet(image=IMAGE)
OBJ1=Select(image=OBJ0, object=OBJ1, query='Daniel Craig', category=None)
IMAGE0=Emoji(image=IMAGE, object=OBJ1, emoji='smiling_face_with_sunglasses')
OBJ2=Select(image=IMAGE, object=OBJ0, query='Sean Connery', category=None)
IMAGE1=Emoji(image=IMAGE0, object=OBJ2, emoji='winking_face')
RESULT=IMAGE1
```

IMAGE: 

Prediction: IMAGE0

IMAGE: 

Instruction: Replace desert with lush green grass

Program:

```
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=OBJ0, object=OBJ1, query='desert', category=None)
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt='lush green grass')
RESULT=IMAGE0
```

IMAGE: 

Prediction: IMAGE0

IMAGE: 

Instruction: Create a color pop of Barack Obama (person)

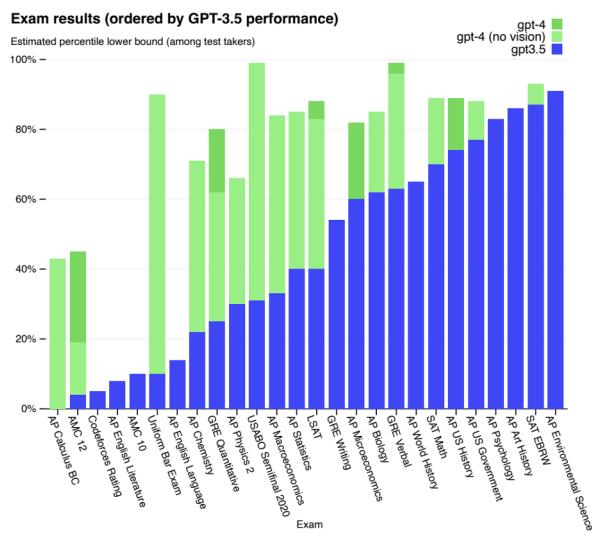
Program:

```
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=OBJ0, object=OBJ1, query='Barack Obama', category='person')
IMAGE0=ColorPop(image=IMAGE, object=OBJ1)
RESULT=IMAGE0
```

Visual Programming (Gupta & Kembhavi, CVPR 2023)

Image Understanding	Loc OWL-ViT	FaceDet DSFD (pypi)	Seg MaskFormer	Select CLIP-ViT	Classify CLIP-ViT	Vqa ViLT
	Replace Stable Diffusion	ColorPop PIL.convert() cv2.grabCut()	BgBlur PIL.GaussianBlur() cv2.grabCut()	Tag PIL.rectangle() PIL.text()	Emoji AugLy (pypi)	
Image Manipulation	Crop PIL.crop()	CropLeft PIL.crop()	CropRight PIL.crop()	CropAbove PIL.crop()	CropBelow PIL.crop()	
	List GPT3	Arithmetic & Logical		Eval eval()	Count len()	Result dict()

Do As I Can, Not As I Say: Grounding Language in Robotic Affordances (SayCan) (Ahn et al., CoRL 2023)



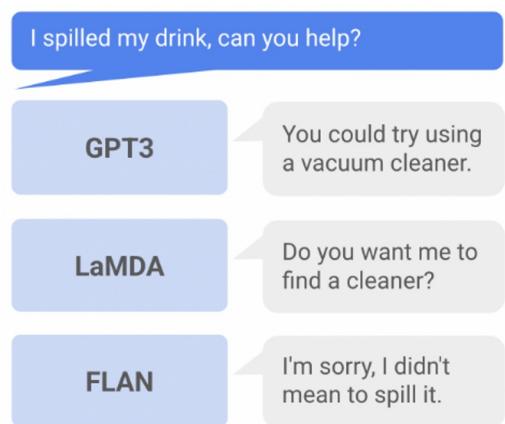
LLMs contain large amounts of commonsense knowledge



Can this be harnessed by an embodied agent?

OpenAI. "GPT-4 Technical Report." *ArXiv* (2023).
Huang, Wenlong et al. "Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents." *ICML* (2022)

LLMs are not grounded in the real world



1. Doesn't know which actions are doable for an physical agent
2. Doesn't know about physical state of environment
3. Or Physical State of Agent

SayCan Method

1. Score likelihood: a skill will make progress towards goal or high level instruction
2. Affordance function: likelihood of successfully completing a *skill* from current state
 - a. Uses reinforcement learning (RL) to learn language-conditioned value functions that simulate affordance modeling



SayCan: Language x Affordance

c_π = completion

ℓ_π = language description of skill

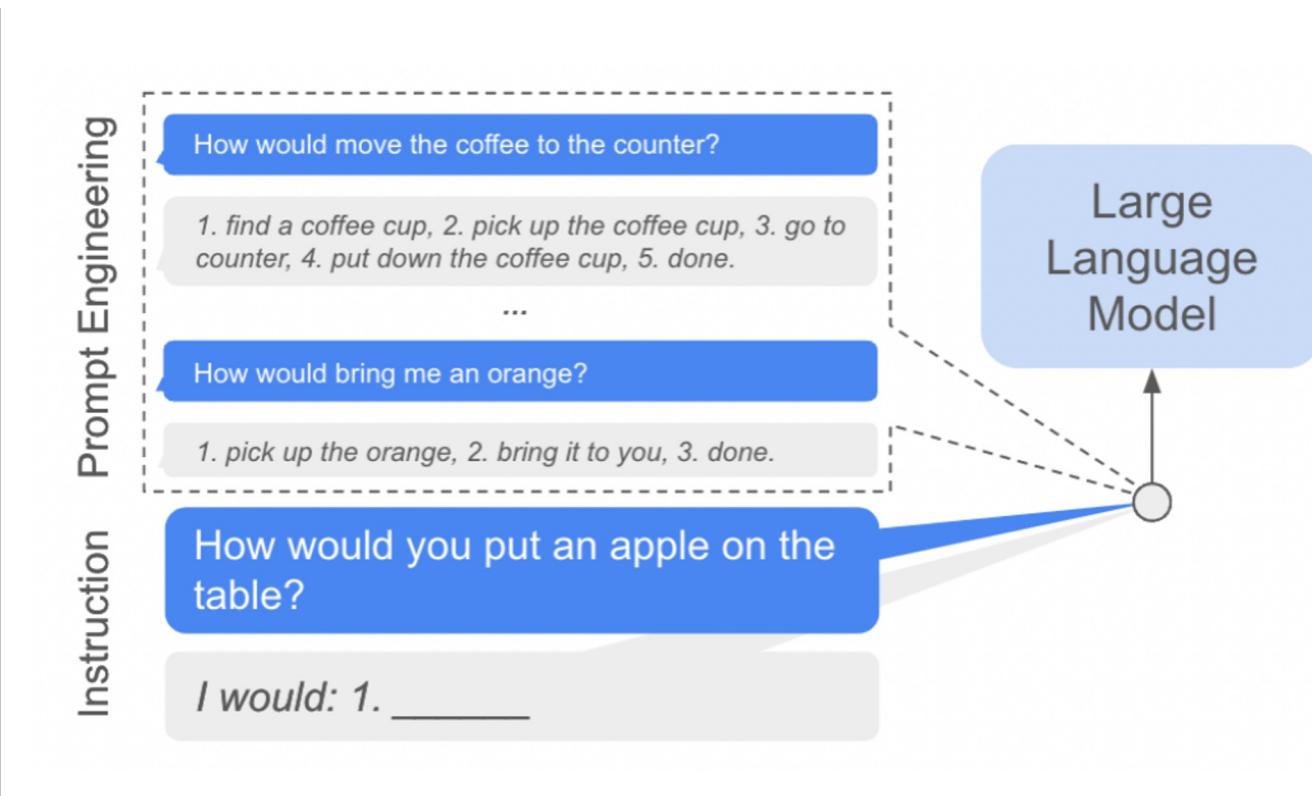
$| i$ = high-level instruction
("How can I clean up
this mess")

skill π and $p(c_\pi|s, \ell_\pi)$ Probability of completing skill given state and language
description of skill (affordance fn; Q fn)

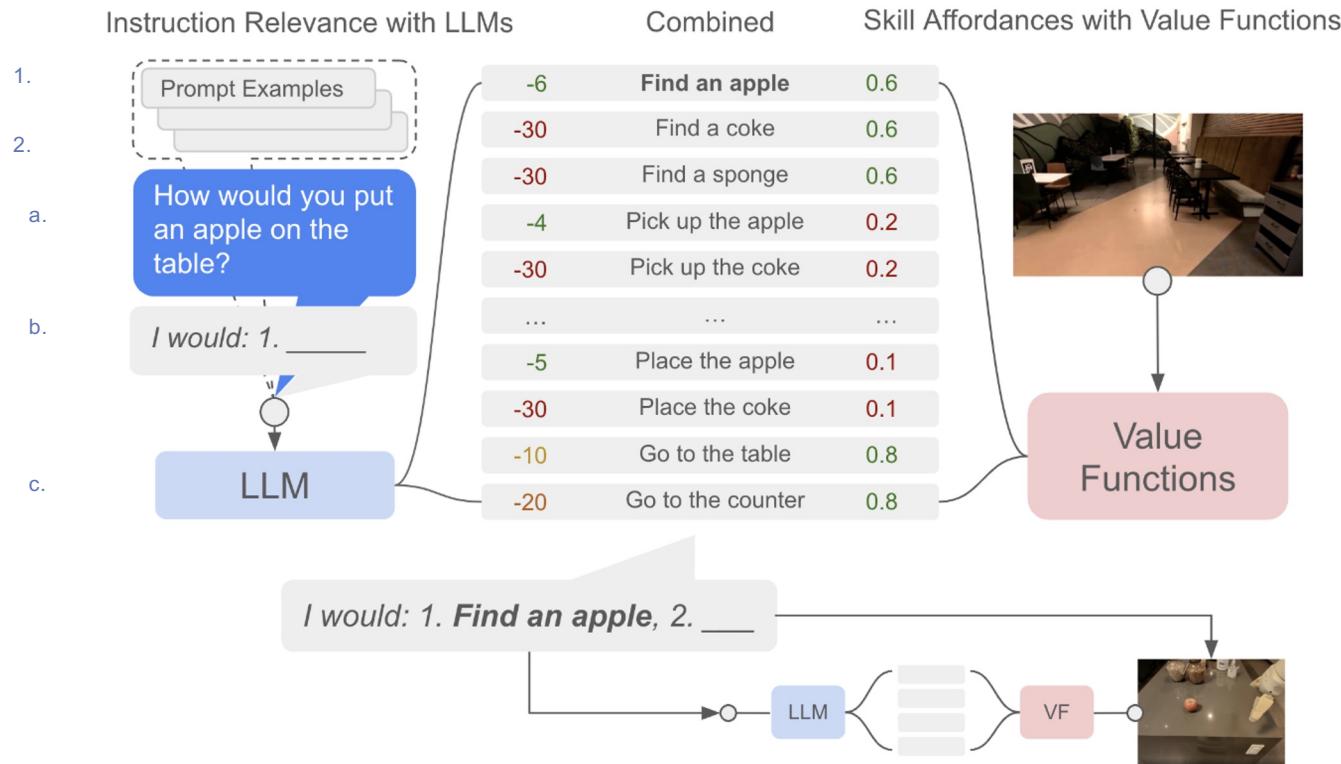
LLM provides us with $p(\ell_\pi|i)$ Probability of skill as a valid next step for
a given instruction and current sequence

$$p(c_i|i, s, \ell_\pi) \propto p(c_\pi|s, \ell_\pi)p(\ell_\pi|i)$$

Constraining output to sequences of primitive skills



Constraining output to sequences of primitive skills



SayCan Algorithm

Algorithm 1 SayCan

Given: A high level instruction i , state s_0 , and a set of skills Π and their language descriptions ℓ_Π

```

1:  $n = 0, \pi = \emptyset$ 
2: while  $\ell_{\pi_{n-1}} \neq \text{"done"}$  do
3:    $\mathcal{C} = \emptyset$ 
4:   for  $\pi \in \Pi$  and  $\ell_\pi \in \ell_\Pi$  do
5:      $p_\pi^{\text{LLM}} = p(\ell_\pi | i, \ell_{\pi_{n-1}}, \dots, \ell_{\pi_0})$                                  $\triangleright$  Evaluate scoring of LLM
6:      $p_\pi^{\text{affordance}} = p(c_\pi | s_n, \ell_\pi)$                                       $\triangleright$  Evaluate affordance function
7:      $p_\pi^{\text{combined}} = p_\pi^{\text{affordance}} p_\pi^{\text{LLM}}$ 
8:      $\mathcal{C} = \mathcal{C} \cup p_\pi^{\text{combined}}$ 
9:   end for
10:   $\pi_n = \arg \max_{\pi \in \Pi} \mathcal{C}$ 
11:  Execute  $\pi_n(s_n)$  in the environment, updating state  $s_{n+1}$ 
12:   $n = n + 1$ 
13: end while

```

SanCay Example Output

Human: I spilled my coke, can you bring me something to clean it up?

Robot: I would
 1. Find a sponge
 2. Pick up the sponge
 3. Bring it to you
 4. Done

Language x Affordance Combined Score

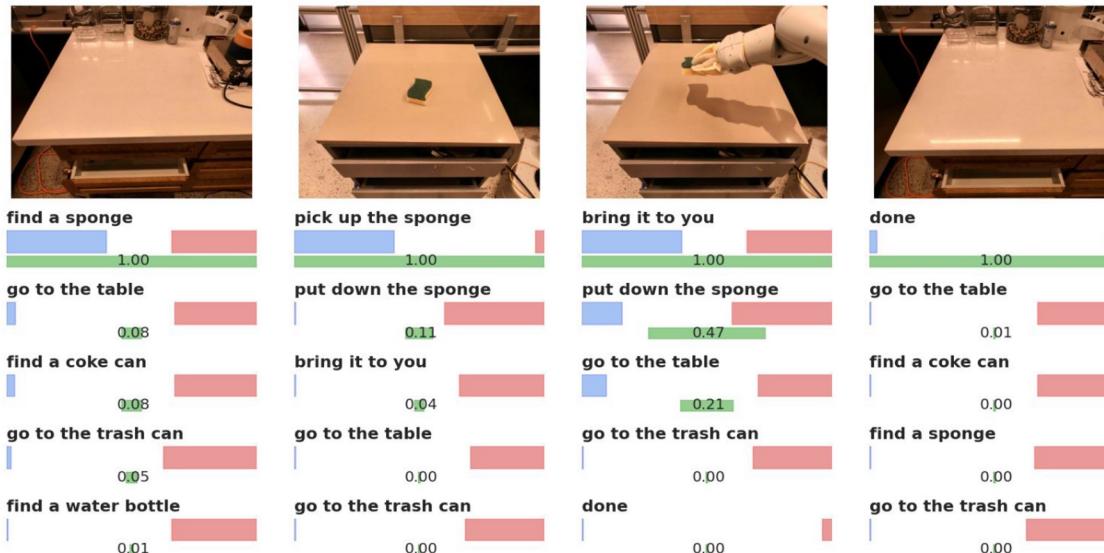


Figure 6: Visualization of PaLM-SayCan's decision making, where the top combined score chooses the correct skill.