Multimodal Language Models

Presented By: Mahshid Naghashzadeh, Patrick Lynch, and Joseph Islam

10/22/2024

Content:

1. VisionLLM: Large Language Model is also an Open-Ended Decoder for Vision-Centric Tasks
2. Visual Instruction Tuning
3. NExT-GPT: Any-to-Any Multimodal LLM
4. Evaluating Object Hallucination in Large Vision-Language Models

Example Scenario: Using GPT-4o for VQA

What is the cat doing in the picture?

The cat in the picture is standing on a table next to a large tipped-over cup. It appears to be curious, possibly investigating the cup or the area around it. The cup looks like it has fallen over, possibly from the cat's actions. There is also a newspaper or paper material underneath the cup, and the overall scene suggests a playful or inquisitive moment for the cat.

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

Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., ... & Dai, J. (**2024**). Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. ***Advances in Neural Information Processing Systems***, *36*.

https://arxiv.org/abs/2305.11175

Motivation - Why Vision and Language Integration?

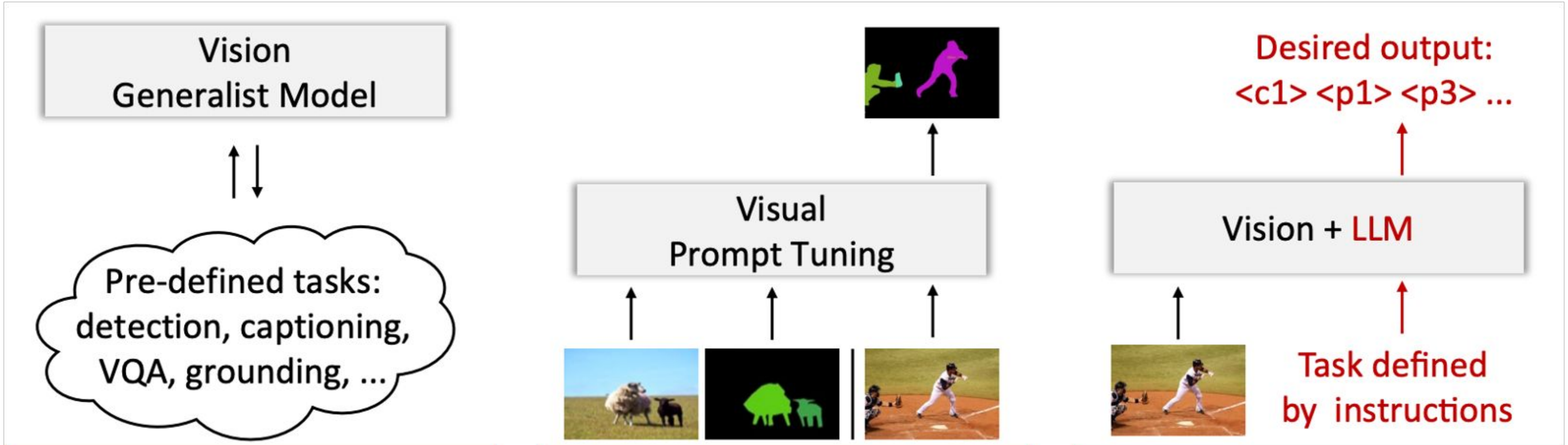
1. **LLMs**
    - Powerful zero-shot capabilities in language tasks, adapting to user prompts effortlessly

2. **Challenge in Vision Tasks**:
    - Inherent differences in data formats and task structures between vision and language

3. **Current Limitations**:
    - Extensive pre-training and fine-tuning
    - Lacking adaptability to diverse tasks and user-specific instructions.

4. **Need for VisionLLM**:
    - Combining the reasoning power of LLMs with vision-specific capabilities
    - Unified model that adapts to flexible, vision-centered prompts

Comparison of VisionLLM with popular paradigms



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

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

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

VisionLLM Overview

1. **Unified Vision-Language Model**:
   - VisionLLM is a unified, end-to-end approach that combines VFMs and LLMs
   - Unlike API-based solutions, it integrates these models within a single cohesive system for efficient vision-centric task handling.
2. **Key Contributions**:
   - **(1)** First open-ended framework for vision tasks using LLMs.
   - **(2)** Language-guided tokenizer and task decoder to align vision tasks with language instructions.
   - **(3)** Demonstrates high generalization across diverse scenarios (e.g., object detection, image captioning).
3. **Impact**:
   - Enables flexible task management, extending beyond pre-defined tasks, to bridge vision and language domains.

Visualization of VisionLLM diverse tasks

Random Object Categories

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

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

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

Random Output Format

Human: *"For each object in <image> that is a member of {'motorbike': <c0>}, output a tuple with the class label and the coordinates of a polygon with 16 points that encloses the object. The coordinates should be within the range 1024. The output format should be (c, x1, y1, x2, y2, ..., x16, y16)."*

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

(b) Object Location with *Editable Output Format*

Random Task Description

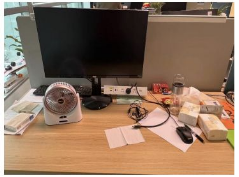
Human: *"Provide a long description of the <image>."*

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

(c) Image Description with *Controllable Text Length*

Random Task Description

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



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

(d) Visual Question Answer with *Complex Reasoning*

Background - Vision Generalist Models

1. **Generalist Models for Vision**:
    - Models like OFA, Flamingo, and GIT use a unified framework to handle a wide range of vision tasks.
    - These models often employ sequence-to-sequence (seq2seq) methods, making them capable of both image-based and non-image-based tasks.
2. **Limitations**:
    - Current models face challenges in fine-tuning and performance due to non-parallel auto-regressive processes.
    - These models are often restricted by pre-defined tasks and do not fully support flexible, open-ended tasks defined by natural language instructions.

Background - Instruction Tuning

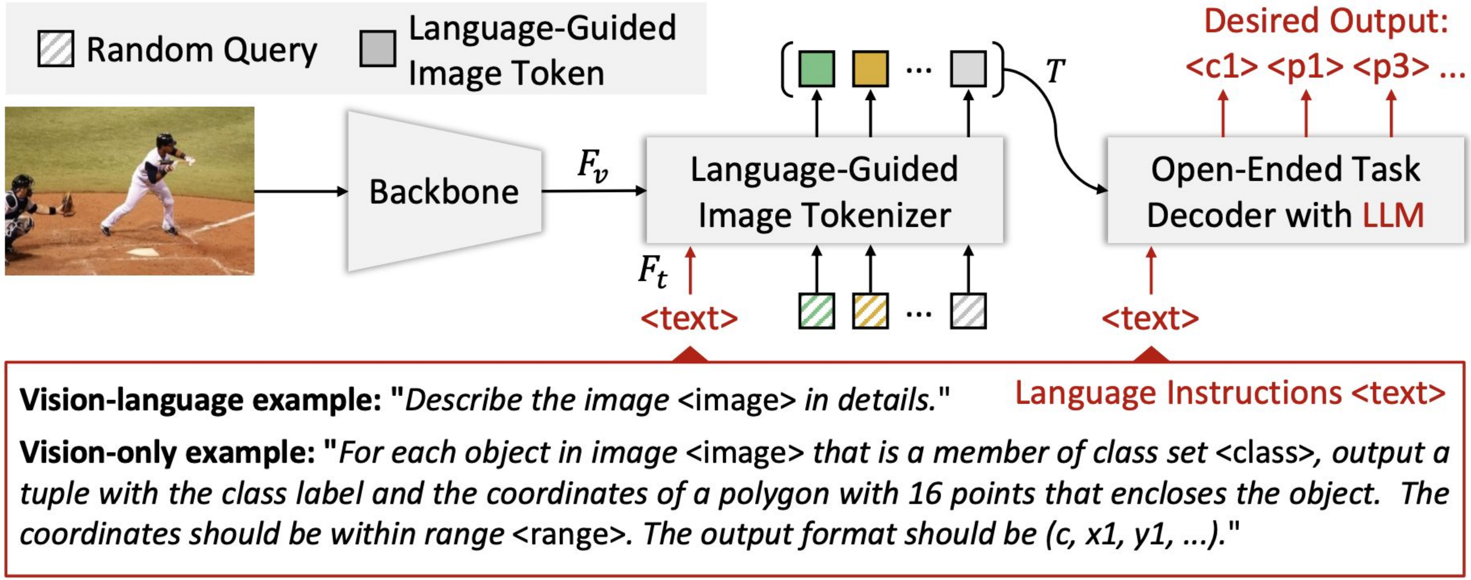
Instruction Tuning: Introduced by **GPT-3**, enhances **zero-shot** and **few-shot** learning for LLMs (e.g., **InstructGPT, FLAN**).

Vision-Language Models:

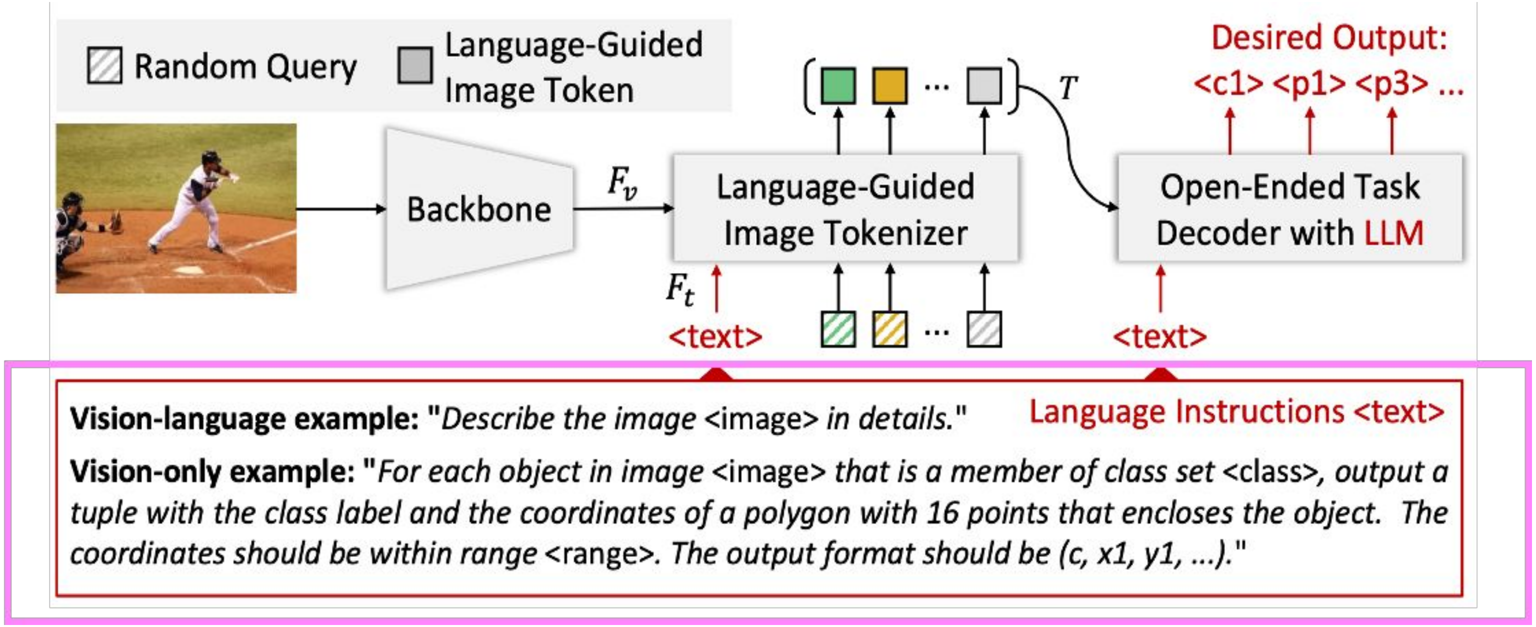
- **Flamingo**: Uses **vision + language prompts** for tasks like **image captioning** and **VQA**.
- **BLIP-2**: Connects visual encoders with LLMs via a **querying transformer**.
- **MiniGPT-4 & LLaVA**: Fine-tune BLIP-2 models for **image-to-text** tasks.

Challenges: Models struggle with **visual perception tasks** (e.g., detection, segmentation), and visual prompts don't fully leverage LLM reasoning.

VisionLLM: Model Architecture



VisionLLM: Unified Language Instruction



VisionLLM: Unified Language Instruction

Vision-Language Tasks:

- Example: “The image is `<image>`. Please generate a caption.”
- Instructions follow formats like NLP tasks for image captioning and Visual Question Answering (VQA).

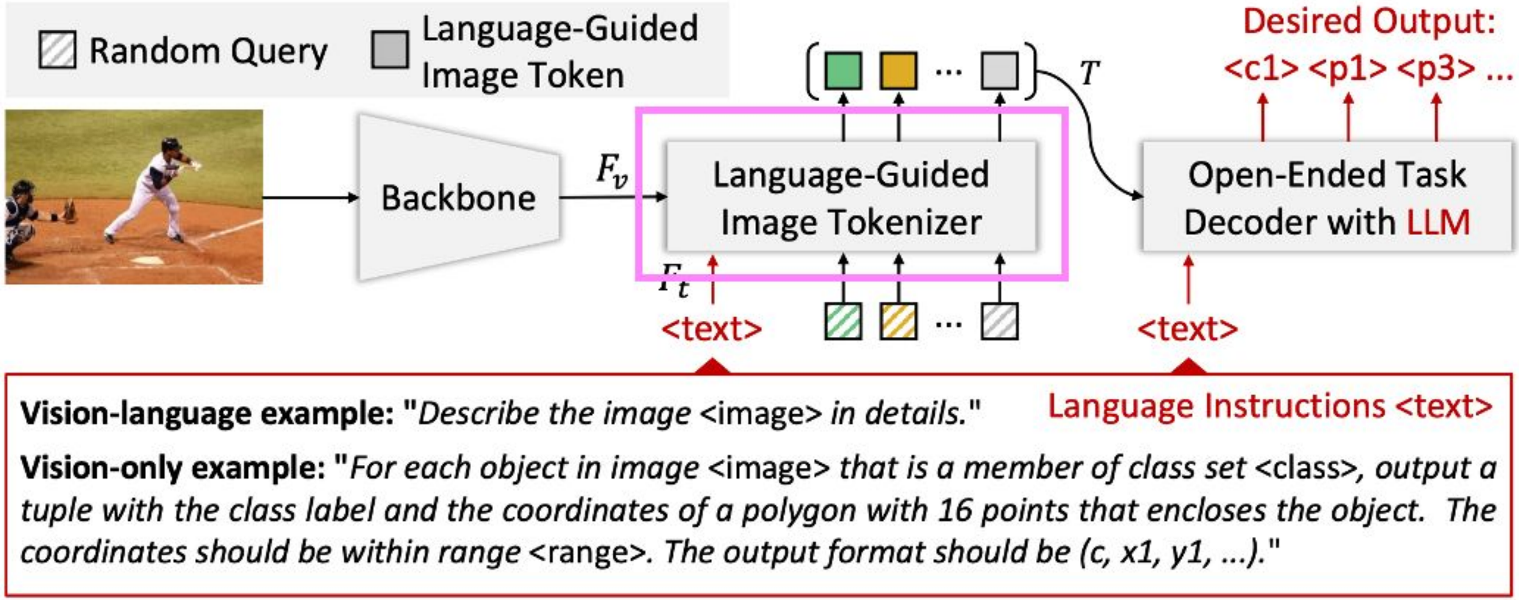
Vision-Only Tasks:

- Example: “Segment all objects of class `<class>` in the image and generate bounding coordinates.”
- Unified output format `(C, P)` with class and boundary points for object detection and segmentation.

  C: Class index in the category set `<class>`

  P: N points that locate the subject

VisionLLM: Language-Guided Image Tokenizer

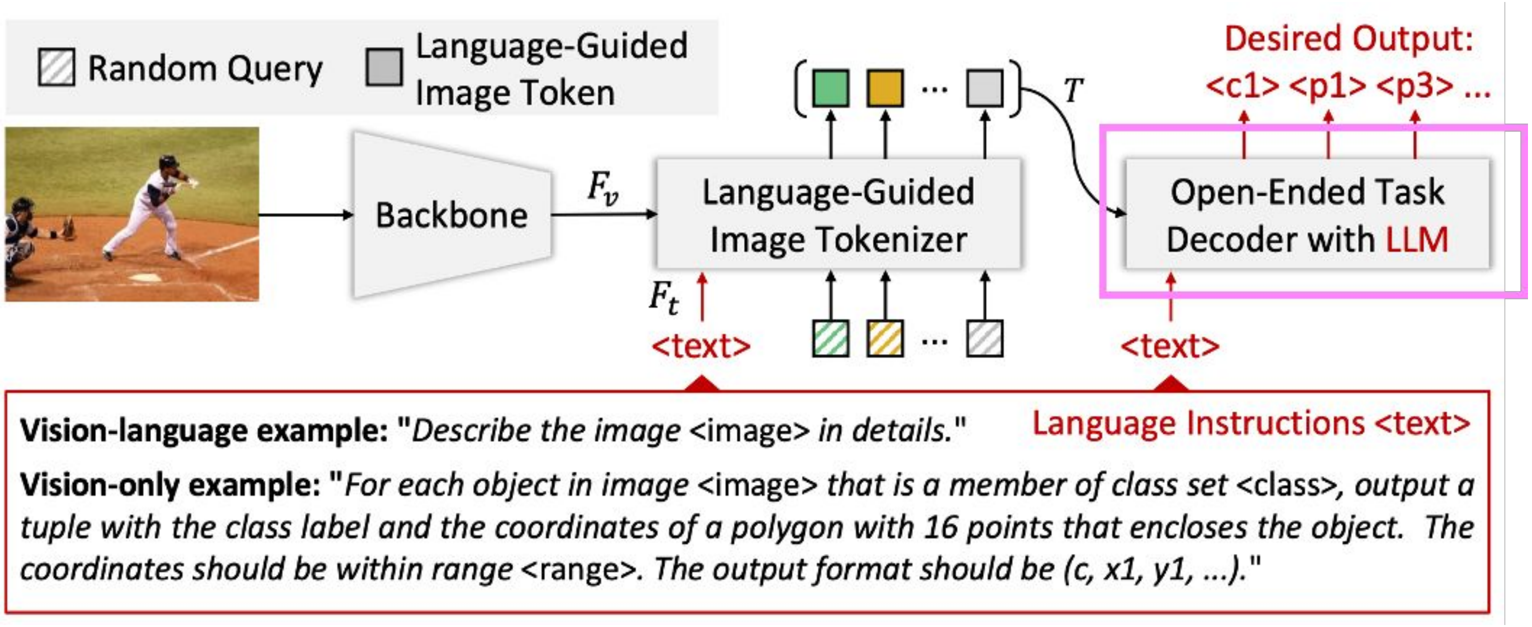


VisionLLM: Language-Guided Image Tokenizer

Process:

1. **Feature Extraction**:
   - Images $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ processed through backbones (e.g., ResNet) to extract 4-scale visual features F_v
2. **Cross-Modality Fusion**:
   - Text encoders (e.g., BERT) extract language features F_l integrated into each scale of visual features using cross-attention, resulting in language-aware visual features.
3. **Output Tokens**:
   - A transformer-based network (e.g., Deformable DETR) converts the features into token embeddings $T = \{(e_i, l_i)\}_{i=1}^{M}$, representing semantic and positional data, respectively.

LLM-based Open-Ended Task Decoder



LLM-based Open-Ended Task Decoder

Decoder is built on Alpaca (from LLaMA) to handle vision tasks using language instructions.

- **Challenges**:
  1. Limited ability to handle object locations with numerical precision.
  2. Inefficiency in category name classification using multiple tokens.
  3. Causal model inefficiencies for visual tasks.

Proposed Solution:

- **1. Location Tokens**: Introduce discretized position tokens to streamline object localization tasks.
  - Denoted as {<p-512>, ..., <p0>, ..., <p512>}
  - Each token <p i> represents the discretized offset of **i** within the range **[-512, 512]**.
  - The token's relative value to the image height or width is calculated as **i / 512**.

- **2. Category Tokens**: Replace multi-token categories with unified classification tokens.

LLM-based Open-Ended Task Decoder

3. Output-Format-as-Query: Parse structured formats from language instructions for efficient decoding.

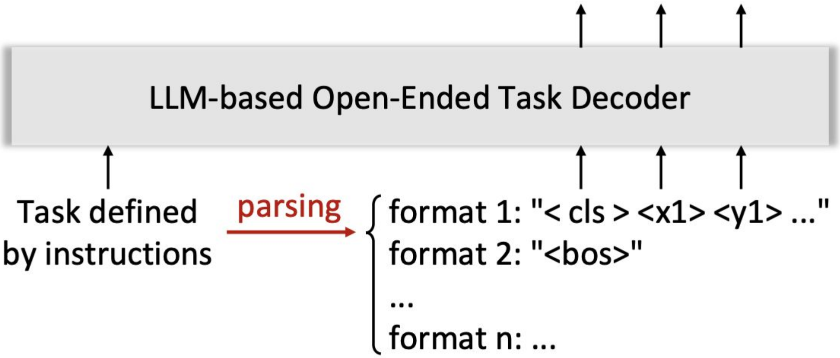


Figure 4: Illustration of the “output-format-as-query” decoding process. “<cls> <x1> <y1> ...” denote the queries of the object’s class index and boundary points, and “<bos>” denotes the beginning of string.

Experimental Settings

Datasets:

- VisionLLM is trained and evaluated on multiple vision-language tasks:
  - **COCO2017** for object detection and segmentation.
  - **RefCOCO** and **RefCOCO+** for visual grounding.
  - **COCO Caption** and **LLaVA-Instruct-150K** for image captioning and VQA.

Implementation:

- Two backbone variants: ResNet and InternImage-H.
- Text encoder: **BERT Base**.
- Vision encoder: **Deformable DETR**.
- LLM: **Alpaca-7B** (fine-tuned with **LoRA**)

Results on standard vision-centric tasks

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

AP: Average Precision

P@0.5: Precision at 0.5

BLUE-4: Bilingual Evaluation Understudy)

CIDEr: Consensus-based Image Description Evaluation

Object Customization & Ablation Studies on Image Tokenization

Object Detection:

- Outperforms Pix2Seq by 1.4 mAP on ResNet-50.
- Efficient multi-task predictions using output-format-as-query.

Visual Grounding: Achieves 86.7 P@0.5 using InternImage-H.
Instance Segmentation: Competitive AP scores with ResNet and InternImage-H.
Image Captioning: Comparable BLEU-4 and CIDEr scores, outperforming prior models.

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

(a) Object-level customization.

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

(b) Output format customization.

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

Ablation Study

- **Single Task vs. Multiple Tasks:**
  - Single-task model VisionLLM-R50_sep slightly outperforms the multi-task model.
  - Multi-tasking leads to a trade-off between generalization and accuracy.

- **Text Encoder in Language-Guided Image Tokenizer:**
  - BERT plays a crucial role in visual grounding but is less essential for object detection.
  - Freezing BERT hinders alignment between vision and language modalities.

- **Image Tokenization Method:**
  - Query-based tokenization outperforms average pooling due to its flexibility in capturing object size information.

- **Number of Localization Tokens:**
  - Increasing the number of tokens improves localization performance, peaking at 1025 tokens.

Ablation Study

Table 3: **Ablation studies on language-guided image tokenizer and hyper-parameters.**

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

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

(b) Effect of image tokenization method.

Tokenization	AP
Average Pooling	23.1
Ours	44.8

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

#Bins	AP
257	34.9
513	40.8
1025	44.8
2049	44.8

Conclusion and Key Takeaways

- VisionLLM is a novel framework that integrates large language models (LLMs) for vision-centric tasks, promoting open-ended and customizable tasks like object detection, instance segmentation, and image captioning.

- The unified language instruction approach allows flexible task definitions using natural language.

- VisionLLM achieves competitive performance across tasks by leveraging a language-guided image tokenizer and an LLM-based task decoder.

Future Directions

- Explore more advanced model scaling for larger datasets.
- Incorporate more diverse vision-language tasks.
- Investigate potential real-world applications and challenges in multi-modal systems.

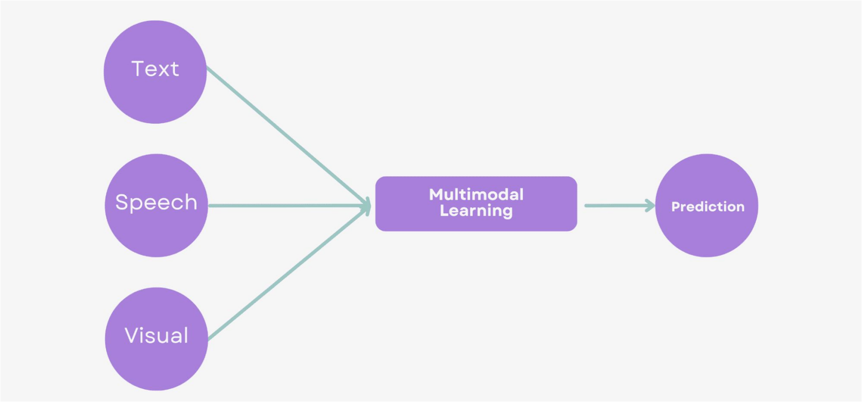
Visual Instruction Tuning

Liu et al. (2023)

https://arxiv.org/abs/2304.08485

Motivation

- Onset of multimodal models
- Instruction tuning for LLMs
- Goal, make vision language models:
    - more user friendly
    - more general purpose



Overview

- Instruction-following vision-language data pipeline
- LLaVa
    - Multimodal vision-language model
- LLaVa-Bench
    - 2 quantitative benchmarks (COCO and In-the-Wild)

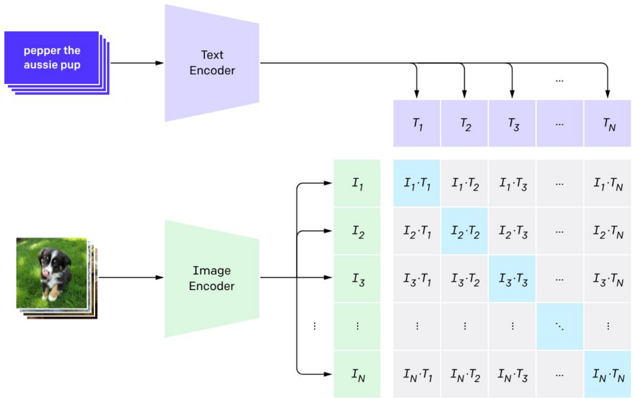
Source: https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg

User: What is unusual about this image?

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

Background

- CLIP
  - Radford et al. (2021). Learning Transferable Visual Models From Natural Language Supervision
  - Contrastive objective, positive samples pushed closer, negative samples pushed further away
  - Robust image encoder
- Vicuna
  - Chiang et al. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality
  - LLM



GPT-assisted Visual Instruction Generation

- Supply text GPT-4 with captions and object bboxes
- Few human generated annotations for in-context learning
- 3 types:
  - Conversation
  - Detailed Description
  - Complex Reasoning
- 158K samples
- For Multimodal Chatbot fine-tuning

Context type 1: Captions

A group of people standing outside of a black vehicle with various luggage.
Luggage surrounds a vehicle in an underground parking area
People try to fit all of their luggage in an SUV.
The sport utility vehicle is parked in the public garage, being packed for a trip
Some people with luggage near a van that is transporting it.

Context type 2: Boxes

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

Response type 1: conversation

Question: What type of vehicle is featured in the image?
Answer: The image features a black sport utility vehicle (SUV) ...<omitted>

Response type 2: detailed description

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

Response type 3: complex reasoning

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

System

- CLIP image encoder
- Embedding projection layer
- Vicuna LLM

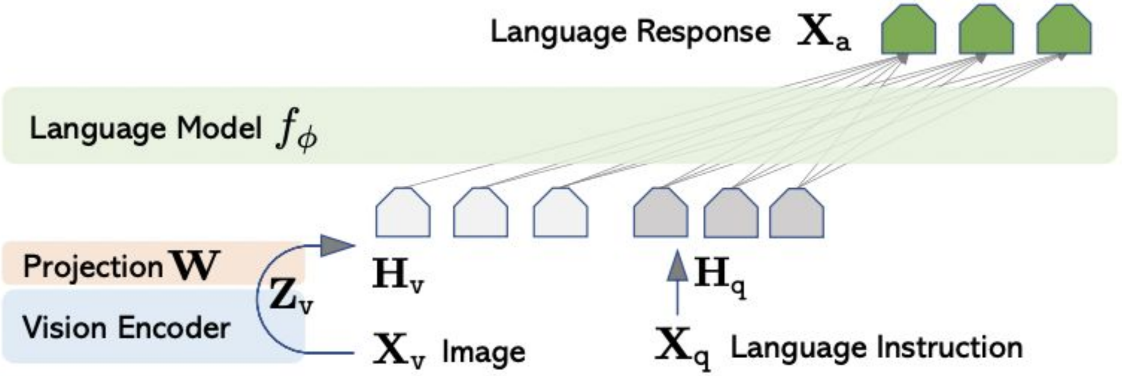


Figure 1: LLaVA network architecture.

2-Stage Training

- Stage 1: Pre-training for Feature Alignment
  - Visual encoder ❄️, LLM ❄️, projection matrix 🔥
  - 595K image-text pairs from CC3M
  - Converted to instruction-following data, GT is captions

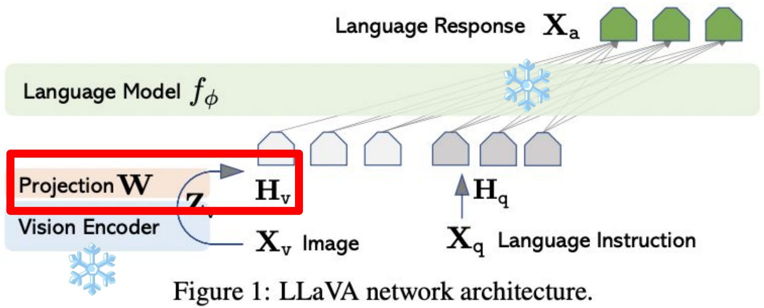


Figure 1: LLaVA network architecture.

2-Stage Training

- Stage 1: Pre-training for Feature Alignment
- Stage 2: Fine-tuning End-to-End
  - Visual encoder ❄️, LLM 🔥, projection matrix 🔥
  - Use Multimodal Chatbot and ScienceQA datasets

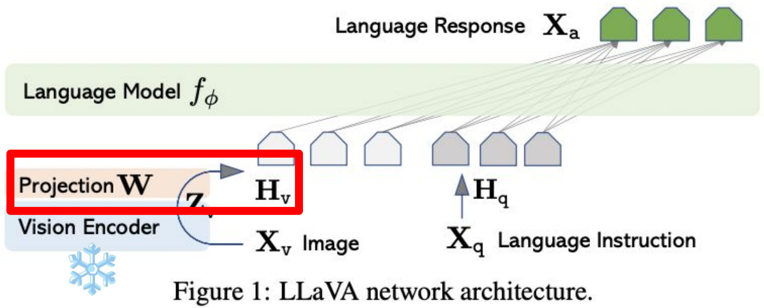


Figure 1: LLaVA network architecture.

Multimodal Chatbot Evaluation

- Fine-tuned with 158K instruction-following dataset
- Generate response from LLaVa
- Generate response from text GPT-4
- Use separate text GPT-4 as judge
    - Given bboxes, captions, LLaVa output, GPT-4 output

Multimodal Chatbot Evaluation

- 1) LLaVa-Bench (COCO)
- 30 images from COCO-Val-2014
  - 3 variations each: conversation, detailed description, complex reasoning
  - 90 questions in total

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

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

Multimodal Chatbot Evaluation

- 2) LLaVa-Bench (In-the-Wild)
- 24 hand-collected “challenging” images
  - indoor/outdoor, sketches, paintings, memes
  - 60 questions in total

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

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

Challenges

- Requiring high-res images
  - Fine-grained details
- Broad knowledge coverage
  - Ramen side dishes
- “bag of patches”
  - Strawberries + yogurt = strawberry yogurt

Filled fridge [source]

An open refrigerator filled with a variety of food items. In the left part of the compartment, towards the front, there is a plastic box of strawberries with a small bag of baby carrots on top. Towards the back, there is a stack of sauce containers. In the middle part of the compartment, towards the front, there is a green plastic box, and there is an unidentified plastic bag placed on it. Towards the back, there is a carton of milk. In the right part of the compartment, towards the front, there is a box of blueberries with three yogurts stacked on top. The large bottle of yogurt is Fage non-fat yogurt, and one of the smaller cups is Fage blueberry yogurt. The brand and flavor of the other smaller cup are unknown. Towards the back, there is a container with an unknown content.

What is the brand of the blueberry-flavored yogurt?

Is there strawberry-flavored yogurt in the fridge?

ScienceQA Evaluation

- Fine-tune using 21K diverse multiple choice questions
- Model-ensembling with few-shot GPT-4
    - Complement - when GPT-4 fails to answer, use LLaVa
    - Judge - when GPT-4 and LLaVa generate different answers, ask GPT-4 again using those 2 answers

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

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

Conclusion + Future Directions

- Visual instruction tuning is effective
- SOTA on ScienceQA
- Excellent for visual-chat comprehension questions
- In the future,
  - More fine-tuning
  - Fusing other modalities

NExT-GPT: Any-to-Any Multimodal LLM

Wu et al.

Joseph Islam (2024)

https://arxiv.org/pdf/2309.05519

Research Context

Multimodality inherently limits all text only llms, despite their linguistic formidability.

- Modalities include audio, images, videos, and text.

Multimodality extends traditional llms using modern “adapter” approaches.

- multimodal **input** llms: BLIP-2, Flamingo, MiniGPT-4, Video-LLaMa, LLaVa, PandaGPT, and SpeechGPT

Human-AI task alignment requires more effort towards multimodal output models

- Multimodal **output** llms: Emu, DreamLLM, GILL, SEED. **fewer!**

Destination

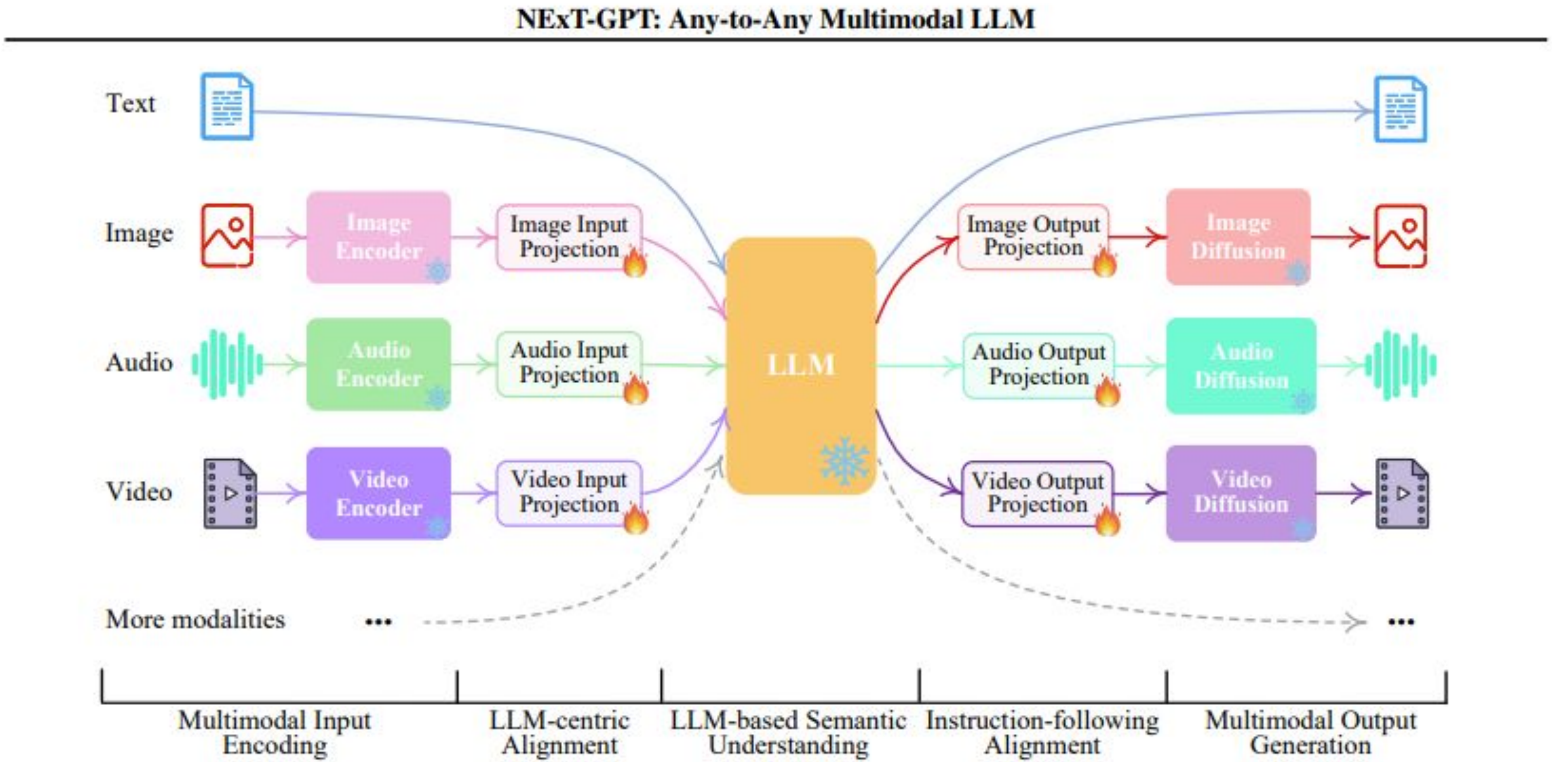
Challenges in Any-to-Any Multimodal LLMs.

- Sub-SOTA reasoning (CoDi 2023)

- Clunky processing pipelines pose misalignment and limited performance as task complexity increases. (Visual-ChatGPT, Hugging GPT 2023)

- End to End Pipelines like NExT-GPT with Tuned Models Address resolve this

NExT-GPT: Any-to-Any Multimodal LLM



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

- Encoding:
  - Leverages ImageBind to accept text, image, video, audio.
  - Synthesizes input with Concept Tokens (discussed later)

- LLM
  - Vicuna (7B-v0) with LoRA tuning for modern reasoning.
  - Produces mode signal token with output to designate diffusion mode

- Decoding
  - Stable Diffusion (SD-v1.5) for image generation.
  - Zeroscope (v2-576w) for video generation.
  - AudioLDM (l-full) for audio synthesis.

- Dataset: MosIT. Covered more later. Basically, designed for this model and task.

- Task: Train Projection layers to tie pipeline together *in an intuition aligned way.*

	Encoder		Input Projection	
	Name	Param	Name	Param
Text	—	—	—	—
Image Audio Video	ImageBind	1.2B	Grouping	28M

LLM		Output Projection	
Name	Param	Name	Param
		—	—
Vicuna (LoRA	7B 33M)	Transformer	31M
		Transformer	31M
		Transformer	32M

Diffusion	
Name	Param
—	—
SD	1.3B
AudioLDM	975M
Zeroscope	1.8B

Next-GPT: Lightweight Training Overview

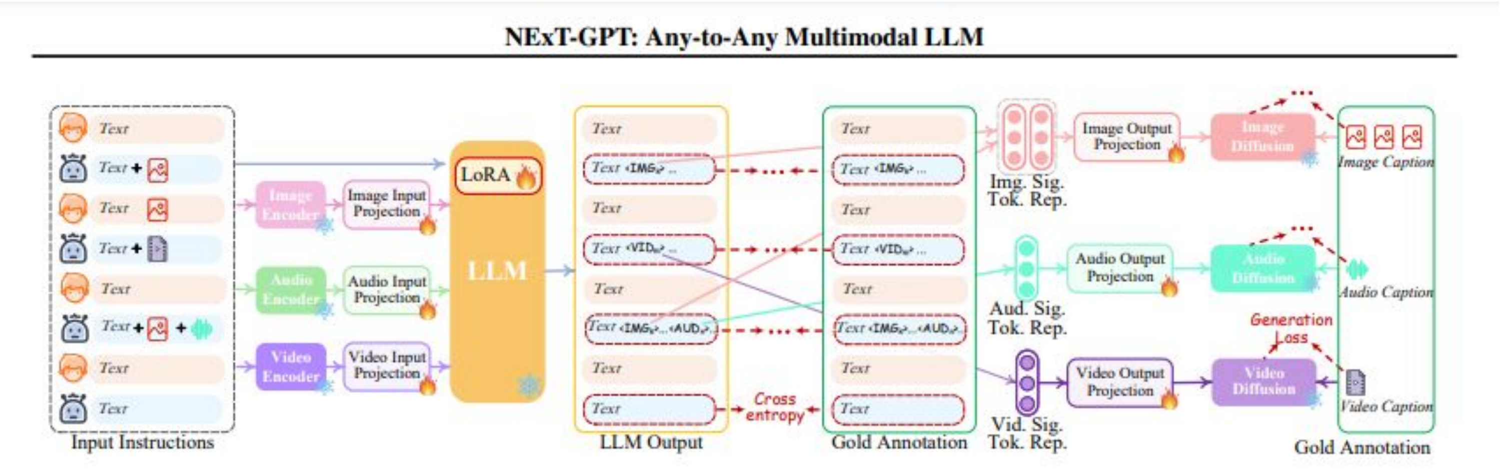


Figure 3. Illustration of modality-switching instruction tuning.

The existing models, ImageBind, Vicuna, and the diffusers, are frozen for training. Notice the Snowflake!
The lightweight projection layers are trained. Notice the Fire!

Steps:

- Train away deformed reasoning from fusing fragmented encodings using cross entropy
- LoRA Tune Vicuna for diffusion mode indicating signal tokens and diffuser aligned prompts.
- Ensure Intuitive diffusion alignment to resolve clunky, aforementioned weird output painpoint by end using MosIT.

Next-GPT: Lightweight Training Overview

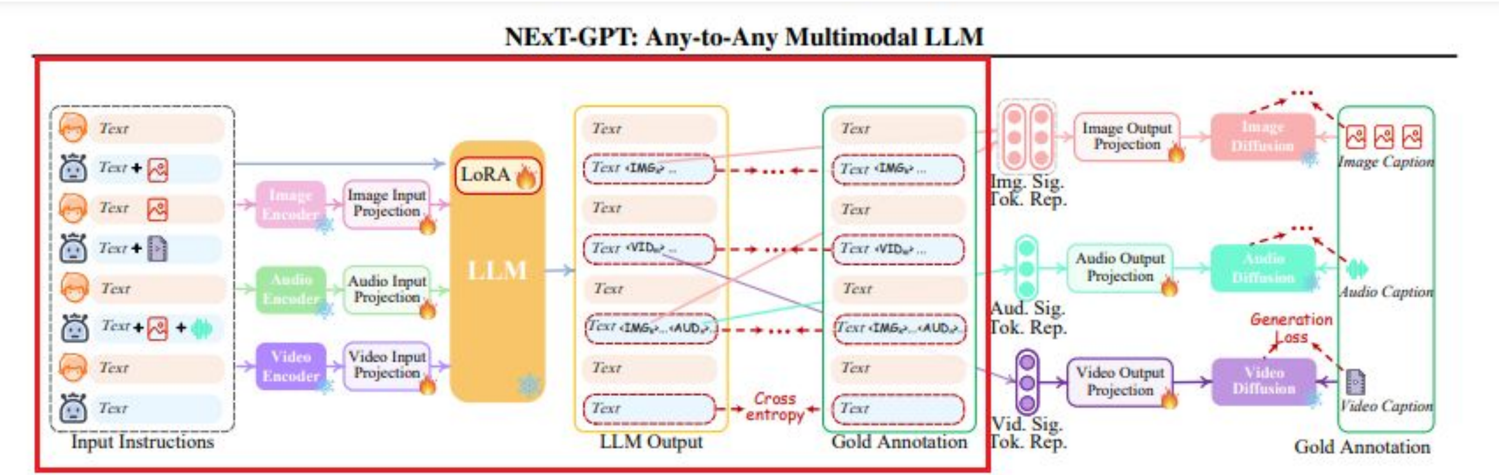


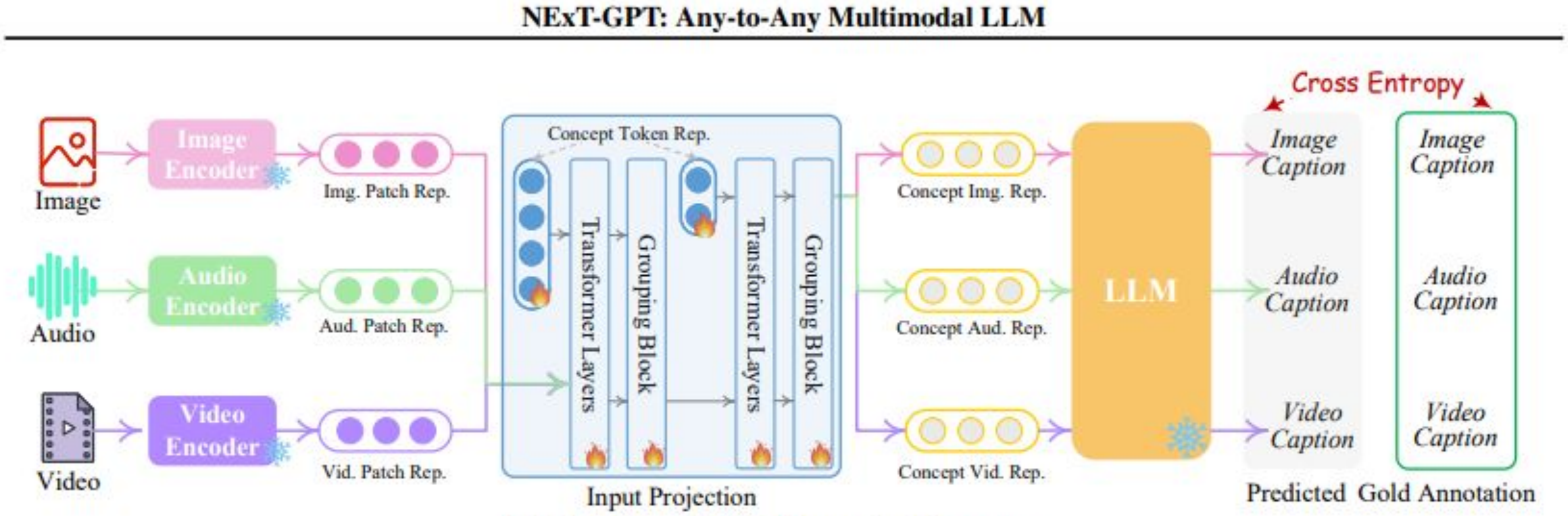
Figure 3. Illustration of modality-switching instruction tuning.

The existing models, ImageBind, Vicuna, and the diffusers, are frozen for training. Notice the Snowflake!
The lightweight projection layers are trained. Notice the Fire!

Steps:

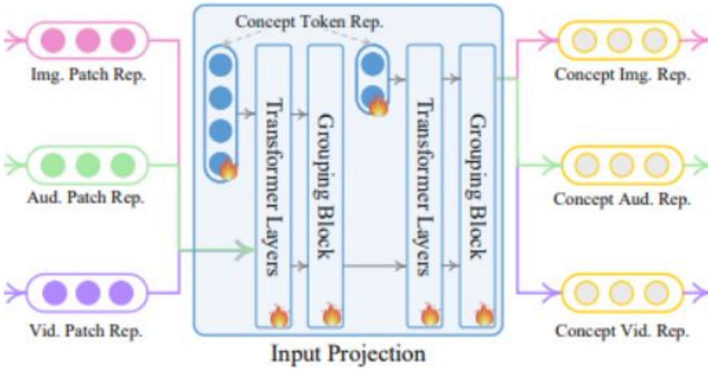
- ⚠️ Train away deformed reasoning from fusing fragmented encodings using cross entropy
- LoRA Tune Vicuna for diffusion mode indicating signal tokens and diffuser aligned prompts.
- Ensure Intuitive diffusion alignment to resolve clunky, aforementioned weird output painpoint by end using MosIT.

Next-GPT: The Encoding Step



(a) Encoding-side LLM-centric Alignment

Next-GPT: Concept Tokens



Patch level features reduces reasoning performance.

Example:

The image encoder gives isolated visual fragments, like “red pants” or “belt buckle” or “arm.” LLM gets confused. What is that? (Zhong et. al. 2022)

Say now given a video. Beyond what’s there, what’s happening, as a concept?

Concept Token Rep Breakdown:

- Transformer and unifies the idea of the objects in the scene.
    - (person leg + spine = book) + (table leg + tabletop = table) = book on table. Object Resolution.
- Grouping block adds coherency to business / across time.
    - Person looking at book for a 5 seconds may be a person reading! Busy moment resolution.
- Transformer now aggregates multiple concepts in case of busy scene.
    - Person reading + person reading + person reading = reading group. Busy scene resolution.
- Grouping Block
    - Reading Group in session. Concept Resolution.

Train the Fire!

Next-GPT: Lightweight Training Overview

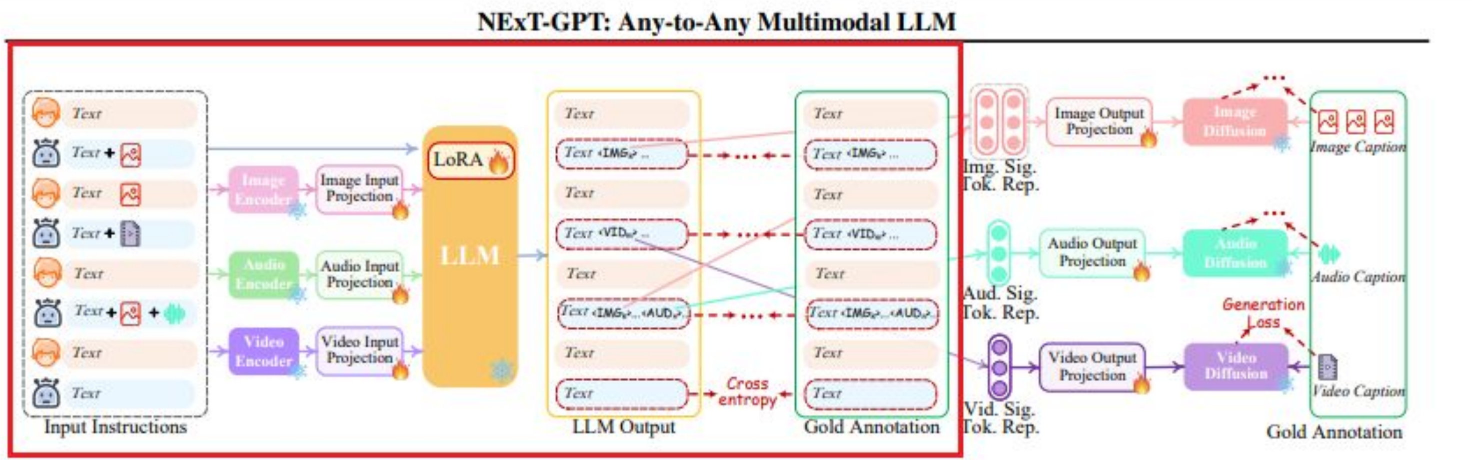


Figure 3. Illustration of modality-switching instruction tuning.

The existing models, ImageBind, Vicuna, and the diffusers, are frozen for training. Notice the Snowflake!
The lightweight projection layers are trained. Notice the Fire!

Steps:

- ✅ Train away deformed reasoning from fusing fragmented encodings using cross entropy
- LoRA Tune Vicuna for diffusion mode indicating signal tokens and diffuser aligned prompts.
- Ensure Intuitive diffusion alignment to resolve clunky, aforementioned weird output painpoint by end using MosIT.

Next-GPT: Lightweight Training Overview

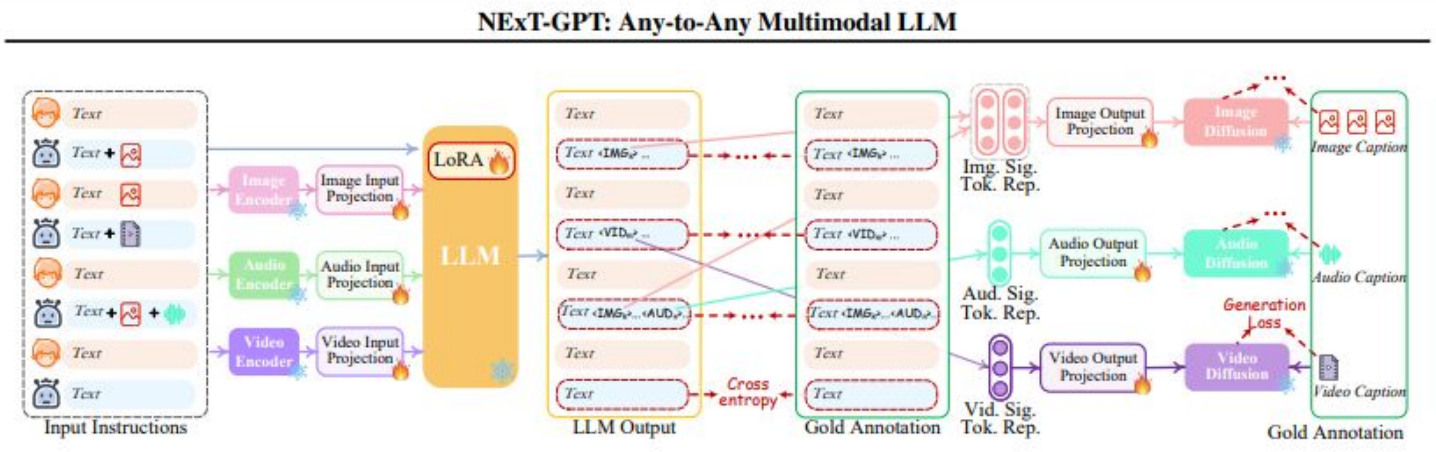


Figure 3. Illustration of modality-switching instruction tuning.

The existing models, ImageBind, Vicuna, and the diffusers, are frozen for training. Notice the Snowflake!
The lightweight projection layers are trained. Notice the Fire!

Steps:

✅ Train away deformed reasoning from fusing fragmented encodings using cross entropy

- LoRA Tune Vicuna for diffusion mode indicating signal tokens and diffuser aligned prompts.
- Ensure Intuitive diffusion alignment to resolve clunky, aforementioned weird output painpoint by end using MosIT.

Next-GPT: Lightweight Training Overview

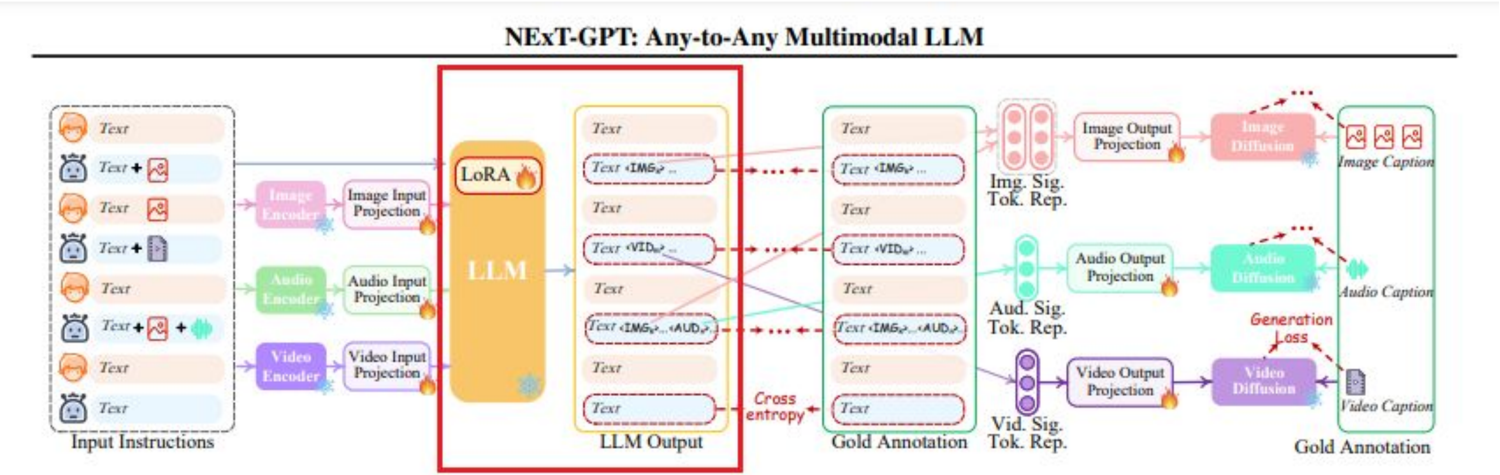


Figure 3. Illustration of modality-switching instruction tuning.

The existing models, ImageBind, Vicuna, and the diffusers, are frozen for training. Notice the Snowflake!
The lightweight projection layers are trained. Notice the Fire!

Steps:

- ✅ Train away deformed reasoning from fusing fragmented encodings using cross entropy
- ⚠️ LoRA tune Vicuna for diffusion mode indicating signal tokens and diffuser aligned prompts.
- Ensure Intuitive diffusion alignment to resolve clunky, aforementioned weird output painpoint by end using MosIT.

NExT-GPT: Vicuna Tuning

No LoRA, then these occur.

- Painpoint: Model unreliably diffuses in correct mode.
- Painpoint: Bad reasoning from base Vicuna's misalignment multimodality
- Painpoint: Compounding errors through system from vicuna misalignment.

Applying LoRA

- Challenge: limited data exists to align diffuser selection and pipeline intuition.
- Resolution: MosIT gold annotations LoRA Vicuna for signal tokens and quality

Reduces immediate or compounding comprehension and reasoning errors.

Changes about 1% of parameters.

Next-GPT: Lightweight Training Overview

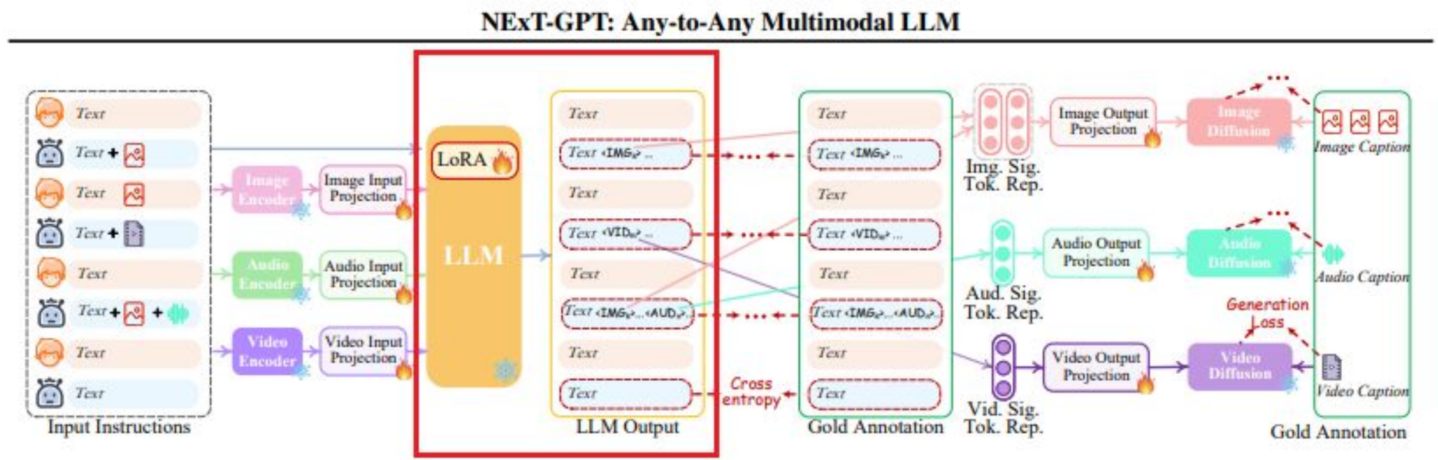


Figure 3. Illustration of modality-switching instruction tuning.

The existing models, ImageBind, Vicuna, and the diffusers, are frozen for training. Notice the Snowflake!
The lightweight projection layers are trained. Notice the Fire!

Steps:

✅ Train away deformed reasoning from fusing fragmented encodings using cross entropy
✅ LoRA tune Vicuna for diffusion mode indicating signal tokens and diffuser aligned prompts.
- Ensure Intuitive diffusion alignment to resolve clunky, aforementioned weird output painpoint by end using MosIT.

Next-GPT: Lightweight Training Overview

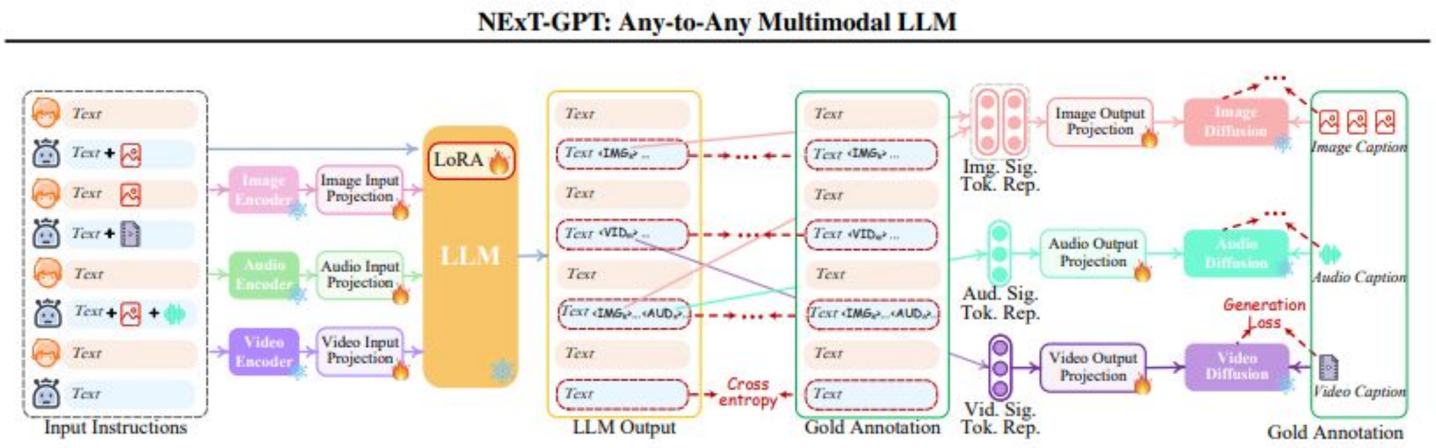


Figure 3. Illustration of modality-switching instruction tuning.

The existing models, ImageBind, Vicuna, and the diffusers, are frozen for training. Notice the Snowflake!
The lightweight projection layers are trained. Notice the Fire!

Steps:

- ✅ Train away deformed reasoning from fusing fragmented encodings using cross entropy
- ✅ Tune Vicuna for diffusion mode indicating signal tokens and diffuser aligned prompts.
- ⚠️ Ensure Intuitive diffusion alignment to resolve clunky, aforementioned weird output painpoint by end using MosIT.

Next-GPT: Lightweight Training Overview

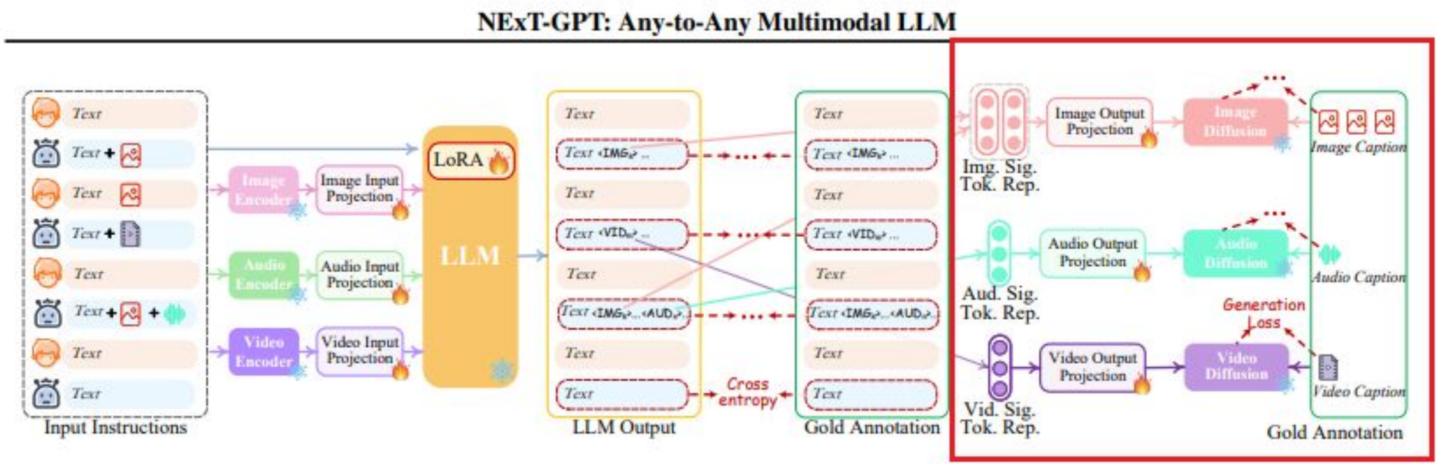


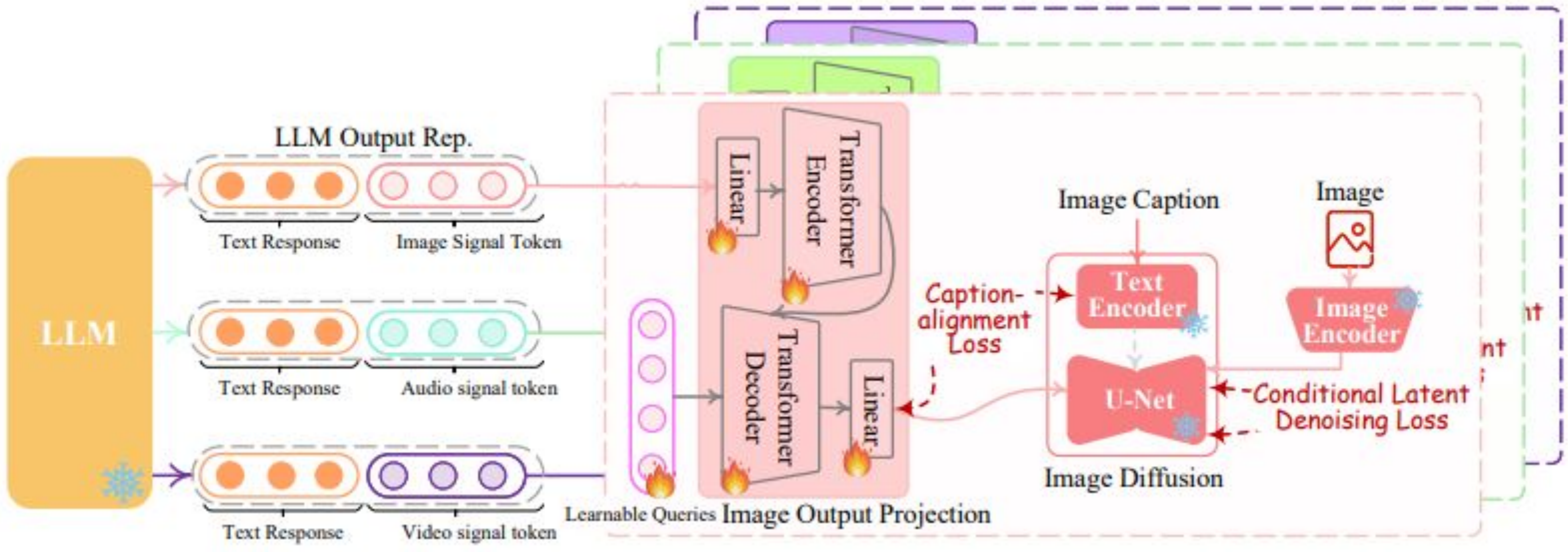
Figure 3. Illustration of modality-switching instruction tuning.

The existing models, ImageBind, Vicuna, and the diffusers, are frozen for training. Notice the Snowflake!
The lightweight projection layers are trained. Notice the Fire!

Steps:

✅ Train away deformed reasoning from fusing fragmented encodings using cross entropy
✅ Tune Vicuna for diffusion mode indicating signal tokens and diffuser aligned prompts.
⚠️ Ensure Intuitive diffusion alignment to resolve clunky, aforementioned weird output painpoint by end using MosIT.

Next-GPT Decoding Side



(b) Decoding-side Instruction-following Alignment

Output projection: Takes Vicuna Output, Signal Token, and a trainable attention vector. Linear Fits input to transformer. Encoder emphasizes with attn. Decoder emphasizes using attention vector. Linear finish

Caption Alignment Loss: Ensures that diffuser reflects learned prompt from Image Output projection. This adds semantic consistency to the model allows output to better reflect input intuition.

Conditional Latent Denoising Loss: Ensures high diffusion quality. This helps the model ensure very clean, high quality images, so that the output is saitsfactory to the end user.

Next-GPT: Decoding Side Training

Challenge: Ensure diffuser intent matches intuition with CE over vicuna output.

Beats LoRA’ing Vicuna for each diffuser.

Challenge: Ensure quality of output. Diffusers can output subpar results. We resolve this with Conditional Latent Denoising Loss.

Challenge: Diffuse in correct mode. Trivially, use signal token.

This minimizes training cost while ensuring overall alignment and quality.

Next-GPT: Lightweight Training Overview

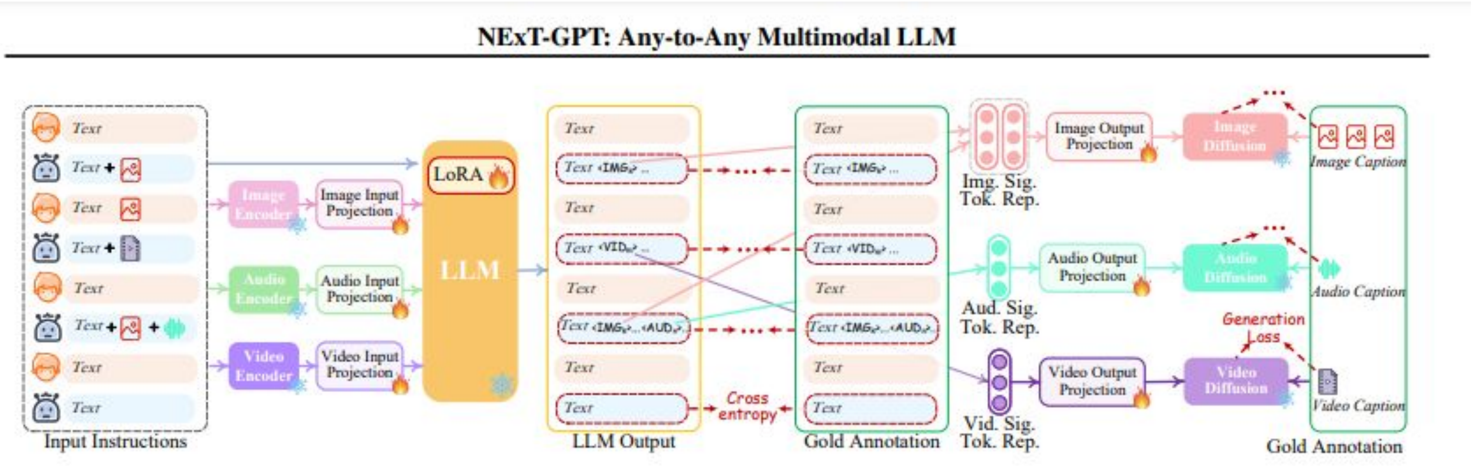


Figure 3. Illustration of modality-switching instruction tuning.

The existing models, ImageBind, Vicuna, and the diffusers, are frozen for training. Notice the Snowflake!
The lightweight projection layers are trained. Notice the Fire!

Steps:
- ✅ Train away deformed reasoning from fusing fragmented encodings using cross entropy
- ✅ Tune Vicuna for diffusion mode indicating signal tokens and diffuser aligned prompts.
- ✅ Ensure Intuitive diffusion alignment to resolve clunky, aforementioned weird output painpoint by end using MosIT.

Next-GPT: Pipeline Alignment Challenges

Challenges:

1. Gap in Understanding User Instructions:
   Example: A user requests "Show me a video of a sunset with calming music," but the model only generates the text description, failing to output both the video and music as requested.
2. Fine-tuning the LLM and Multimodal Outputs:
   Example: Without fine-tuning, when a user inputs "Generate an image of a cat and describe it," the LLM might only describe the cat but fail to output the actual image.
3. Multimodal Signal Token Alignment:
   Example: If the signal token for video [VID] isn't aligned properly, the system may output an image of the sunset instead of a video, even though the user requested a video.

We introduce MosIT

Next-GPT: Pipeline Alignment Resolutions:

MosIT dataset:

- 5k samples designed for **Mo**dality **S**witching **I**nstruction **T**uning that cover many 3-7 turn and many topic human machine interactions
- Turns involve complex reasoning over multimodal samples.
- Designed to mimic real world scenarios involving cross modal information and generate accurate multimodal output
- Template conversations generated using gpt-4.

We now train our entire pipeline over the MosIT dataset to resolve aforementioned intent, follow through, and modality challenges.

Next-GPT: Training over MosIT

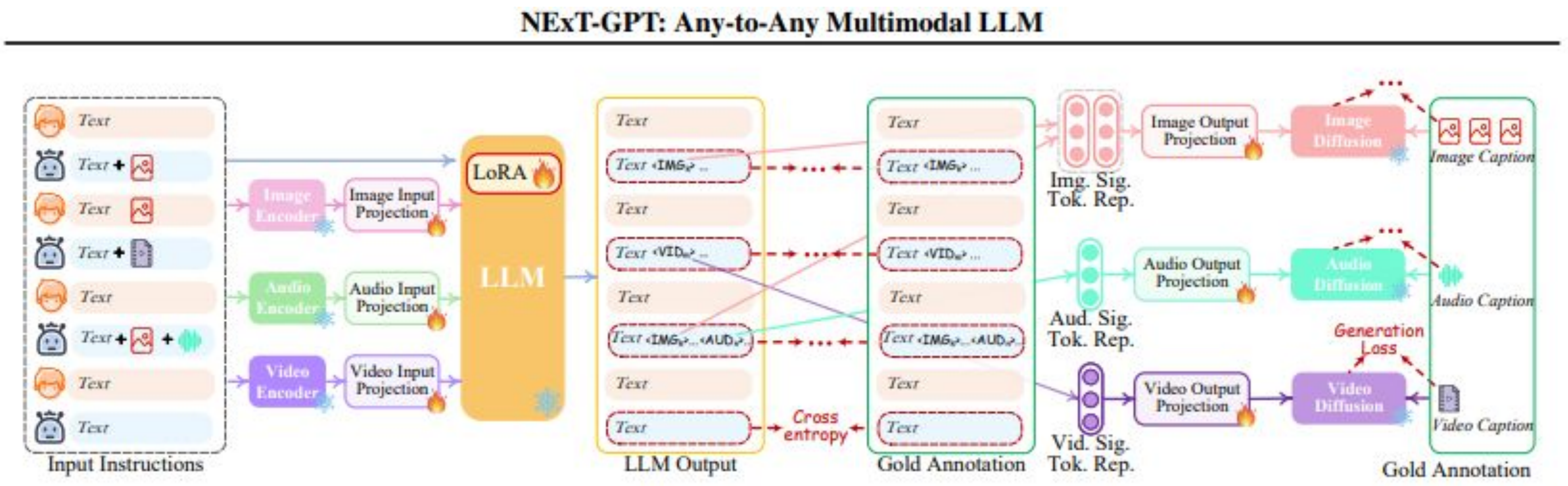


Figure 3. Illustration of modality-switching instruction tuning.

MosIT gold encoding annotations are text for encoding and modal for diffusion.

Each sample provides consistent multimodal input and expected output, which aptly aligns the pipeline.

Next-GPT: Interesting Results

NExT-GPT demonstrates strong performance across image, video, and audio generation tasks, achieving comparable or better results than state-of-the-art models on various benchmarks (Table 2 and 3).

Baseline remains intact.

Table 2. Zero-shot evaluation of image captioning with CIDEr (↑) score on NoCaps (Agrawal et al., 2019), Flickr 30K (Young et al., 2014) and COCO (Karpathy & Fei-Fei, 2017), and image question answering on VQA^{v2} (Goyal et al., 2017), VizWiz (Gurari et al., 2018) and OKVQA (Marino et al., 2019), and two evaluation-only benchmarks, MMB (Liu et al., 2023c) and SEED (Li et al., 2023a). The best results are marked in bold, and the second ones are underlined.

Model	Version	Image Captioning			Image Question Answering			Comprehensive	
		NoCaps	Flickr 30K	COCO	VQA^{v2}	VizWiz	OKVQA	MMB	SEED
InstructBLIP (Dai et al., 2023)	Vicuna-7B	123.1	82.4	102.2	-	33.4	33.9	36.0	-
LLaVA (Liu et al., 2023b)	LLaMA-2-7B-Chat	120.7	82.7	-	-	-	-	36.2	-
mPLUG-Owl (Ye et al., 2023b)	LLaMA-7B	117.0	80.3	119.3	-	39.0	-	46.6	34.0
Emu (Sun et al., 2023)	LLaMA-7B	-	-	117.7	40.0	35.4	34.7	-	-
DREAMLLM (Dong et al., 2023)	Vicuna-7B	-	-	115.4	56.6	45.8	44.3	49.9	-
Video-LLaVA (Lin et al., 2023)	Vicuna-7B	-	-	-	**74.7**	48.1	-	**60.9**	-
NExT-GPT	Vicuna-7B	**123.7**	**84.5**	**124.9**	66.7	**48.4**	**52.1**	58.0	**57.5**

Table 3. Comparison of video reasoning tasks on MSRVTT (Xu et al., 2016), MSVD-QA and MSRVTT-QA (Xu et al., 2017) and NExTQA (Xiao et al., 2021), and the audio captioning task on AudioCaps (Kim et al., 2019). Scores with ∗ means being fine-tuned on the training dataset.

Model	Version	Video Captioning	Video Question Answering			Audio Captioning
		MSR-VTT	MSVD-QA	MSRVTT-QA	NExTQA	AudioCaps
Codi (Tang et al., 2023)	-	74.4*	-	-	-	78.9*
UIO-2XXL (Lu et al., 2023)	6.8B	48.8*	41.5	52.1	-	48.9*
Video-LLaMA (Zhang et al., 2023c)	LLaMA-7B	-	51.6	-	29.6	-
Video-LLaVA (Lin et al., 2023)	Vicuna-7B	-	**70.7**	59.2	-	-
Emu (Sun et al., 2023)	LLaMA-7B	-	32.4	14.0	6.8	-
NExT-GPT	Vicuna-7B	**76.2***	64.5	**61.4**	**50.7**	**81.3***

Next-GPT: Interesting Results

NExT-GPT excels in text-to-image, video, and audio generation, outperforming models like GILL, Emu, and UIO-2XXL.

New SOTA for multimodal output generation.

Special note: Table 4 data collected in zero shot context!

Table 4. Results on text-to-image/audio/video generation (MS COCO (Lin et al., 2014), AudioCaps (Kim et al., 2019), and MSRVTT (Xu et al., 2016)). †: zero-shot results.

Model	Image FID (↓)	Audio FAD (↓)	Video CLIPSIM (↑)
SD-1.5 (Wang et al., 2022c)	11.21	-	-
Codi (Huang et al., 2023a)	11.26	1.80	28.90
AudioLDM-L (Liu et al., 2023a)	-	1.96	-
GILL-8B† (Koh et al., 2023)	12.20	-	-
Emu-13B† (Sun et al., 2023)	11.66	-	-
UIO-2XXL (Lu et al., 2023)	13.39	2.64	-
NExT-GPT	**10.07**	**1.68**	**31.97**
NExT-GPT†	11.18	1.74	30.96

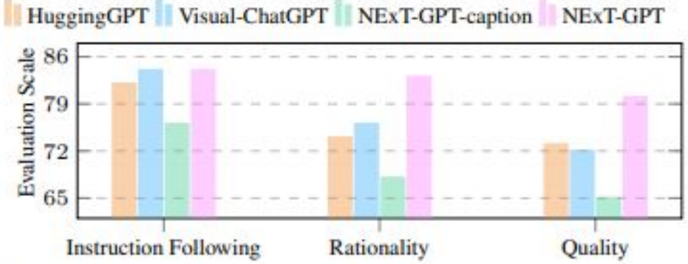
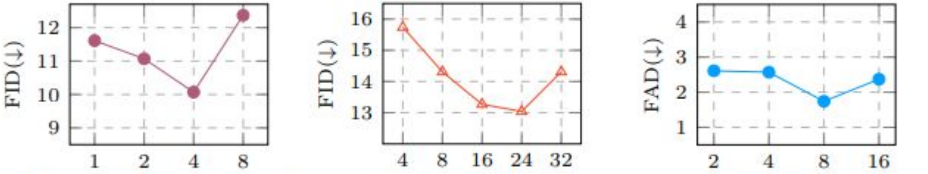


Figure 5. Human Evaluation (1-100 scale, results are on average) of NExT-GPT in comparison with pipeline baselines.

Result Worth Mentioning

NExT-GPT: Any-to-Any Multimodal LLM



(a) Text-to-image Generation (b) Text-to-video Generation (c) Text-to-audio Generation

Figure 6. The generation quality under different numbers of modality signal tokens.

Table 5. The perception performance of NExT-GPT by varying input projection mechanisms.

Model	Image Question Answering		Video Question Answering		Audio Captioning
	VQA^{v2}	VizWiz	MSVD-QA	MSRVTT-QA	AudioCaps
NExT-GPT	66.7	48.4	64.5	61.4	81.3
w Linear Layer	63.8	45.4	60.8	57.1	77.4
w Q-former + Linear Layer	65.1	46.9	63.4	58.1	79.7

Where lower FID is better, signal token count impacts performance.

The starting design is better than slight modifications to input projection.

Next-GPT: Motivating Results

Impressive alignment with user intention encouraged the authors to motivate their results. They think

- The concept tokens enhance perceptual capabilities by aggregating fragmented but related information into a coherence structure before passing it to the model, unlike HuggingGPT and other engineering based architectures
- The end to end training helps align visual and textual data, leading to higher reasoning capacities over multimodality.

Next Steps

- Expand dataset for multimodality. Add complex samples towards robustness
- Optimize model efficiency. Reduce Computation load towards scaling.
- Enhance instruction tuning. Better align instruction following.

Summary

- NextGPT makes a new any-to-any end-to-end fully trained multimodal model.
- It stands out because existing work either covers one side or another or multimodality, or is too engineered for performance.
- It is lightly trained, and highly attuned to user intuition
- It sets a new SOTA in multimodal io, while maintaining existing benchmarks.
- It can be improved with additional data and simplifying architectures, towards better enduser alignment.

Evaluating Object Hallucination in Large Vision-Language Models

Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W. X., & Wen, J. R. (2023). Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

https://arxiv.org/abs/2305.10355

Introduction

Problem Statement:

- Large Vision-language models (LVLMs) are prone to **object hallucination**, i.e., generating descriptions of objects not present in the target image.
- This issue negatively impacts the accuracy of tasks like image captioning and visual question answering (VQA).

In this paper:

- Evaluating object hallucination for several popular LVLMs
- Investigating the effect of visual instructions on object hallucination
- Introducing a new method to evaluate object hallucination

Object Hallucination

What is Object Hallucination?

Models generate descriptions with objects that are **inconsistent** or **absent** from the target image.

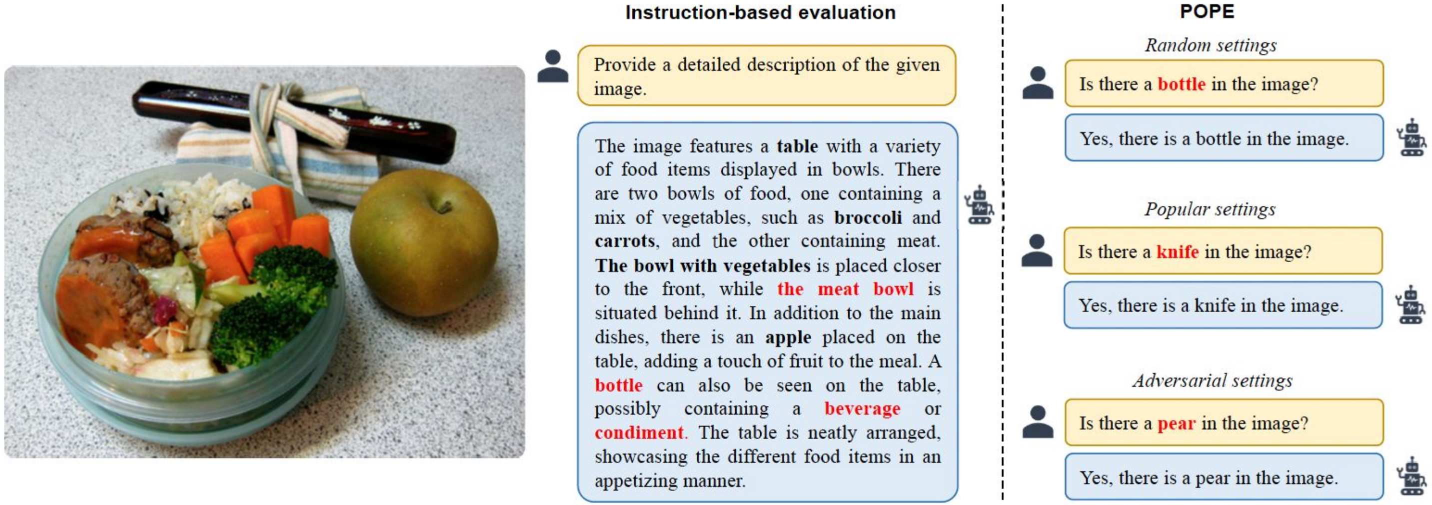


Figure 1: Cases of object hallucination in LVLMs. **Bold** objects are ground-truth objects in the annotations and **red** objects are hallucinated objects by LVLMs. The left case is from the traditional instruction-based evaluation method, and the right cases are from three variants of POPE.

Object Hallucination in LVLMs: Evaluation Settings

CHAIR (Caption Hallucination Assessment with Image Relevance):

- Metric for evaluating object hallucination in **image captioning** tasks.

Two Variants of CHAIR:

1. object instance level.
2. Sentence level.

$$\mathrm{CHAIR}_I = \frac{|\{\text{hallucinated objects}\}|}{|\{\text{all mentioned objects}\}|},$$

$$\mathrm{CHAIR}_S = \frac{|\{\text{captions with hallucinated objects}\}|}{|\{\text{all captions}\}|}.$$

Prompts:

1. Generate a short caption of the image.
2. Provide a brief description of the given image.

Dataset: MSCOCO.

https://arxiv.org/pdf/1405.0312v3

Object Hallucination in LVLMs: Evaluation Results

Different Visual instruction

- InstructBLIP: Trained on public available datasets, shorter

- LLaVA: Visual instruction generated by LLM, longer

I	Model	$CHAIR_I$	$CHAIR_S$	Len
-	$OSCAR_{Base}$	7.1	13.0	-
	$VinVL_{Large}$	5.5	10.5	-
	OFA_{Large}	**4.7**	8.9	-
	$BLIP_{Large}$	**4.7**	**8.8**	-
I_1	mPLUG-Owl	14.8	25.4	35.8
	LLaVA	10.5	32.7	64.3
	MultiModal-GPT	11.1	15.0	11.6
	MiniGPT-4	6.7	9.5	24.7
	InstructBLIP	**2.6**	**3.7**	8.5
I_2	mPLUG-Owl	30.2	76.8	98.5
	LLaVA	18.8	62.7	90.7
	MultiModal-GPT	18.2	36.2	45.7
	MiniGPT-4	9.2	31.5	116.2
	InstructBLIP	**2.5**	**3.4**	7.5

Table 1: Results of CHAIR on VLPMs and LVLMs. I_1 denotes "*Generate a short caption of the image*" and I_2 denotes "*Provide a brief description of the given image*". Len refers to the average length of generated captions. The results of VLPMs (OSCAR, VinVL, BLIP, and OFA) are collected from Dai et al. (2023b). The best results in each block are denoted in bold.

Disadvantages of CHAIR

- Unstable when different instructions are employed.
- Relies on complex human-crafted parsing rules for exact matching

I	Model	CHAIR_I	CHAIR_S	Len
-	OSCAR_{Base}	7.1	13.0	-
	VinVL_{Large}	5.5	10.5	-
	OFA_{Large}	**4.7**	8.9	-
	BLIP_{Large}	**4.7**	**8.8**	-
I_1	mPLUG-Owl	14.8	25.4	35.8
	LLaVA	10.5	32.7	64.3
	MultiModal-GPT	11.1	15.0	11.6
	MiniGPT-4	6.7	9.5	24.7
	InstructBLIP	**2.6**	**3.7**	8.5
I_2	mPLUG-Owl	30.2	76.8	98.5
	LLaVA	18.8	62.7	90.7
	MultiModal-GPT	18.2	36.2	45.7
	MiniGPT-4	9.2	31.5	116.2
	InstructBLIP	**2.5**	**3.4**	7.5

Table 1: Results of CHAIR on VLPMs and LVLMs. I_1 denotes "*Generate a short caption of the image*" and I_2 denotes "*Provide a brief description of the given image*". Len refers to the average length of generated captions. The results of VLPMs (OSCAR, VinVL, BLIP, and OFA) are collected from Dai et al. (2023b). The best results in each block are denoted in bold.

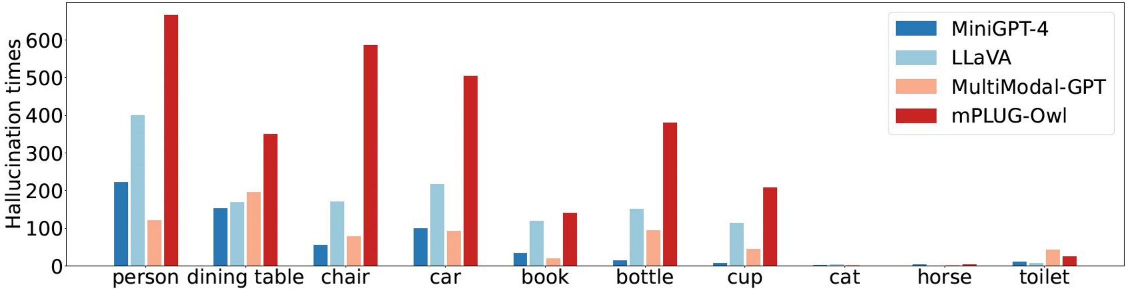
Influence of Instruction Data on Object Hallucination

- **Issue**: Larger LVLMs tend to hallucinate more than smaller VLPMs
- **Possible Cause**: The **visual instruction-tuning process** in LVLMs might exacerbate hallucination.

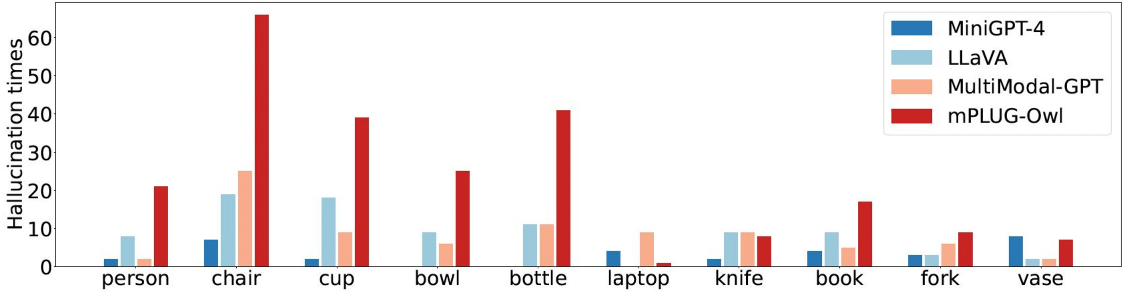
Hypotheses

1. **Hypothesis 1**:
   - LVLMs are prone to hallucinate **frequently appearing objects** in the instruction dataset (e.g., MSCOCO).
2. **Hypothesis 2**:
   - LVLMs hallucinate **co-occurring objects** frequently seen together with the ground-truth objects in the image.

Qualitative Analysis of Object Hallucination



(a) Hallucination times of top ten frequently appearing objects, whose frequencies decrease from right to left.



(b) Hallucination times of top ten objects co-occurring with "dining table", whose frequencies decrease from right to left.

Figure 2: Hallucination times of frequently appearing/co-occurring objects in MSCOCO.

Quantitative Analysis - top-*k* hit ratio

Number of top-k frequently appearing objects in Hallucinated(i)

$$\mathrm{HR}_A@k = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathrm{Hit}@k(i)}{\mathrm{Hallucinated}(i)}$$

Total Number of Images

Number of Hallucinated Objects in i[th] Example

Number of top-k frequently co-occurring objects with the probing object o in Hallucinated(i)

$$\mathrm{HR}_C@k(o) = \frac{1}{m}\sum_{i=1}^{m}\frac{\mathrm{Hit}@k(i,o)}{\mathrm{Hallucinated}(i)}$$

Total Number of Images

Number of Hallucinated Objects in i[th] Example

k: specific number of the **most frequent objects** in the dataset

HR@k : proportion of top-k frequently appearing or co-occurring objects in all hallucinated objects. [0,1]

Quantitative Analysis

Model	HR_A			HR_C(`dining table`)		
	@10	@20	@30	@10	@20	@30
mPLUG-Owl	0.5455	0.6591	0.7533	0.6608	0.7926	0.8253
LLaVA	0.4620	0.5911	0.6796	0.5628	0.7329	0.8595
MultiModal-GPT	0.4152	0.5399	0.6743	0.5742	0.7849	0.8961
MiniGPT-4	0.4610	0.5758	0.7207	0.5600	0.6980	0.9145

Table 2: Results on MSCOCO that quantify the correlations between the appearing/co-occurring frequency of objects and the hallucination times of LVLMs.

Quantitative Analysis

≈ 0.5 ≈ 0.6

Model	HR_A			HR_C(dining table)		
	@10	@20	@30	@10	@20	@30
mPLUG-Owl	0.5455	0.6591	0.7533	0.6608	0.7926	0.8253
LLaVA	0.4620	0.5911	0.6796	0.5628	0.7329	0.8595
MultiModal-GPT	0.4152	0.5399	0.6743	0.5742	0.7849	0.8961
MiniGPT-4	0.4610	0.5758	0.7207	0.5600	0.6980	0.9145

Table 2: Results on MSCOCO that quantify the correlations between the appearing/co-occurring frequency of objects and the hallucination times of LVLMs.

Quantitative Analysis

Model	HR_A		≈ **0.7**	HR_C(`dining table`)		≈ **0.9**
	@10	@20	@30	@10	@20	@30
mPLUG-Owl	0.5455	0.6591	0.7533	0.6608	0.7926	0.8253
LLaVA	0.4620	0.5911	0.6796	0.5628	0.7329	0.8595
MultiModal-GPT	0.4152	0.5399	0.6743	0.5742	0.7849	0.8961
MiniGPT-4	0.4610	0.5758	0.7207	0.5600	0.6980	0.9145

Table 2: Results on MSCOCO that quantify the correlations between the appearing/co-occurring frequency of objects and the hallucination times of LVLMs.

Quantitative Analysis

≈ 0.7 ≈ 0.9

Model	HR_A			HR_C(dining table)		
	@10	@20	@30	@10	@20	@30
mPLUG-Owl	0.5455	0.6591	0.7533	0.6608	0.7926	0.8253
LLaVA	0.4620	0.5911	0.6796	0.5628	0.7329	0.8595
MultiModal-GPT	0.4152	0.5399	0.6743	0.5742	0.7849	0.8961
MiniGPT-4	0.4610	0.5758	0.7207	0.5600	0.6980	0.9145

Table 2: Results on MSCOCO that quantify the correlations between the appearing/co-occurring frequency of objects and the hallucination times of LVLMs.

LVLMs mostly hallucinate common objects in the visual instruction data.

POPE: Polling-based Object Probing Evaluation

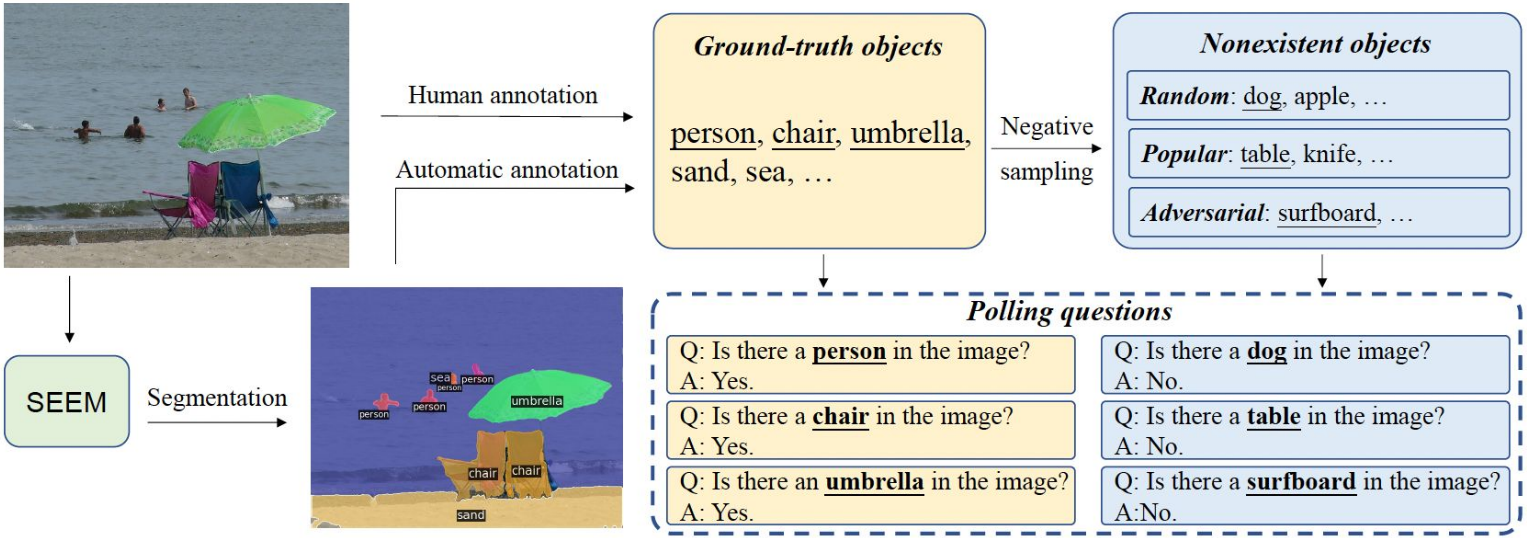


Figure 3: Overview of the POPE pipeline. Given an input image, POPE first extracts ground-truth objects in the image either from human annotations or with the help of automatic segmentation tools like SEEM. Then, POPE conducts negative sampling for nonexistent objects in the image under *Random/Popular/Adversarial* settings. Finally, the ground-truth objects and nonexistent objects are formulated into question templates to poll LVLMs.

POPE: Evaluation on MSCOCO

Dataset	POPE	Model	Accuracy	Precision	Recall	F1 Score	Yes (%)
MSCOCO	*Random*	mPLUG-Owl	53.30	51.71	99.53	68.06	96.23
		LLaVA	54.43	52.32	99.80	68.65	95.37
		MultiModal-GPT	50.03	50.02	**100.00**	66.68	99.97
		MiniGPT-4	77.83	75.38	82.67	78.86	54.83
		InstructBLIP	**88.73**	**85.08**	93.93	**89.29**	55.20
	Popular	mPLUG-Owl	50.63	50.32	99.27	66.79	98.63
		LLaVA	52.43	51.25	99.80	67.72	97.37
		MultiModal-GPT	50.00	50.00	**100.00**	66.67	100.00
		MiniGPT-4	68.30	64.27	82.40	72.21	64.10
		InstructBLIP	**81.37**	**75.07**	93.93	**83.45**	62.57
	Adversarial	mPLUG-Owl	50.67	50.34	99.33	66.82	98.67
		LLaVA	50.77	50.39	99.87	66.98	99.10
		MultiModal-GPT	50.00	50.00	**100.00**	66.67	100.00
		MiniGPT-4	66.60	62.45	83.27	71.37	66.67
		InstructBLIP	**74.37**	**67.67**	93.33	**78.45**	68.97

Table 3: Results of LVLMs under three evaluation settings of POPE on the validation set of MSCOCO. Yes denotes the proportion of answering "Yes" to the given question. The best results in each block are denoted in bold.

POPE: Evaluation on MSCOCO

Dataset	POPE	Model	Accuracy	Precision	Recall	F1 Score	Yes (%)
MSCOCO	*Random*	mPLUG-Owl	53.30	51.71	99.53	68.06	96.23
		LLaVA	54.43	52.32	99.80	68.65	95.37
		MultiModal-GPT	50.03	50.02	**100.00**	66.68	99.97
		MiniGPT-4	77.83	75.38	82.67	78.86	54.83
		InstructBLIP	**88.73**	**85.08**	93.93	**89.29**	55.20
	Popular	mPLUG-Owl	50.63	50.32	99.27	66.79	98.63
		LLaVA	52.43	51.25	99.80	67.72	97.37
		MultiModal-GPT	50.00	50.00	**100.00**	66.67	100.00
		MiniGPT-4	68.30	64.27	82.40	72.21	64.10
		InstructBLIP	**81.37**	**75.07**	93.93	**83.45**	62.57
	Adversarial	mPLUG-Owl	50.67	50.34	99.33	66.82	98.67
		LLaVA	50.77	50.39	99.87	66.98	99.10
		MultiModal-GPT	50.00	50.00	**100.00**	66.67	100.00
		MiniGPT-4	66.60	62.45	83.27	71.37	66.67
		InstructBLIP	**74.37**	**67.67**	93.33	**78.45**	68.97

Table 3: Results of LVLMs under three evaluation settings of POPE on the validation set of MSCOCO. Yes denotes the proportion of answering "Yes" to the given question. The best results in each block are denoted in bold.

POPE: Evaluation on MSCOCO

Dataset	POPE	Model	Accuracy	Precision	Recall	F1 Score	Yes (%)
MSCOCO	*Random*	mPLUG-Owl	53.30	51.71	99.53	68.06	96.23
		LLaVA	54.43	52.32	99.80	68.65	95.37
		MultiModal-GPT	50.03	50.02	**100.00**	66.68	99.97
		MiniGPT-4	77.83	75.38	82.67	78.86	54.83
		InstructBLIP	**88.73**	**85.08**	93.93	**89.29**	55.20
	Popular	mPLUG-Owl	50.63	50.32	99.27	66.79	98.63
		LLaVA	52.43	51.25	99.80	67.72	97.37
		MultiModal-GPT	50.00	50.00	**100.00**	66.67	100.00
		MiniGPT-4	68.30	64.27	82.40	72.21	64.10
		InstructBLIP	**81.37**	**75.07**	93.93	**83.45**	62.57
	Adversarial	mPLUG-Owl	50.67	50.34	99.33	66.82	98.67
		LLaVA	50.77	50.39	99.87	66.98	99.10
		MultiModal-GPT	50.00	50.00	**100.00**	66.67	100.00
		MiniGPT-4	66.60	62.45	83.27	71.37	66.67
		InstructBLIP	**74.37**	**67.67**	93.33	**78.45**	68.97

Table 3: Results of LVLMs under three evaluation settings of POPE on the validation set of MSCOCO. Yes denotes the proportion of answering "Yes" to the given question. The best results in each block are denoted in bold.

POPE: Evaluation on MSCOCO

Dataset	POPE	Model	Accuracy	Precision	Recall	F1 Score	Yes (%)
MSCOCO	*Random*	mPLUG-Owl	53.30	51.71	99.53	68.06	96.23
		LLaVA	54.43	52.32	99.80	68.65	95.37
		MultiModal-GPT	50.03	50.02	**100.00**	66.68	99.97
		MiniGPT-4	77.83	75.38	82.67	78.86	54.83
		InstructBLIP	**88.73**	**85.08**	93.93	**89.29**	55.20
	Popular	mPLUG-Owl	50.63	50.32	99.27	66.79	98.63
		LLaVA	52.43	51.25	99.80	67.72	97.37
		MultiModal-GPT	50.00	50.00	**100.00**	66.67	100.00
		MiniGPT-4	68.30	64.27	82.40	72.21	64.10
		InstructBLIP	**81.37**	**75.07**	93.93	**83.45**	62.57
	Adversarial	mPLUG-Owl	50.67	50.34	99.33	66.82	98.67
		LLaVA	50.77	50.39	99.87	66.98	99.10
		MultiModal-GPT	50.00	50.00	**100.00**	66.67	100.00
		MiniGPT-4	66.60	62.45	83.27	71.37	66.67
		InstructBLIP	**74.37**	**67.67**	93.33	**78.45**	68.97

Table 3: Results of LVLMs under three evaluation settings of POPE on the validation set of MSCOCO. Yes denotes the proportion of answering "Yes" to the given question. The best results in each block are denoted in bold.

POPE: Evaluation on MSCOCO

Dataset	POPE	Model	Accuracy	Precision	Recall	F1 Score	Yes (%)
		mPLUG-Owl	53.30	51.71	99.53	68.06	96.23
		LLaVA	54.43	52.32	99.80	68.65	95.37
	Random	MultiModal-GPT	50.03	50.02	**100.00**	66.68	99.97
		MiniGPT-4	77.83	75.38	82.67	78.86	54.83
		InstructBLIP	**88.73**	**85.08**	93.93	**89.29**	55.20
		mPLUG-Owl	50.67	50.34	99.33	66.82	98.67
		LLaVA	50.77	50.39	99.87	66.98	99.10
	Adversarial	MultiModal-GPT	50.00	50.00	**100.00**	66.67	100.00
		MiniGPT-4	66.60	62.45	83.27	71.37	66.67
		InstructBLIP	**74.37**	**67.67**	93.33	**78.45**	68.97

LVLMs are prone to hallucinate the frequently appearing and co-occurring objects.

Table 3: Results of LVLMs under three evaluation settings of POPE on the validation set of MSCOCO. Yes denotes the proportion of answering "Yes" to the given question. The best results in each block are denoted in bold.

Comparing POPE and CHAIR on Different Prompts

POPE		CHAIR	
Prompt	F1 Score	**Prompt**	CHAIR_I
Is there a <object> in the image?	68.65	Generate a short caption of the image.	10.50
Does the image contain a <object>?	66.83	Provide a brief description of the image.	18.80
Have you noticed a <object> in the image?	66.67	Generate a concise description for the image.	14.60
Can you see a <object> in the image?	67.58	Create a short textual summary for the image.	11.60
Avg±Std.	67.43±0.78		13.88±3.22

Table 4: Evaluation results of LLaVA on POPE and CHAIR with different prompt templates.

Comparing POPE and CHAIR on Different Prompts

POPE		CHAIR	
Prompt	F1 Score	**Prompt**	CHAIR_I
Is there a <object> in the image?	68.65	Generate a short caption of the image.	10.50
Does the image contain a <object>?	66.83	Provide a brief description of the image.	18.80
Have you noticed a <object> in the image?	66.67	Generate a concise description for the image.	14.60
Can you see a <object> in the image?	67.58	Create a short textual summary for the image.	11.60
Avg±Std.	67.43±0.78		13.88±3.22

Table 4: Evaluation results of LLaVA on POPE and CHAIR with different prompt templates.

Stability of POPE compared to CHAIR

POPE Results for Automatic Segmentation

Dataset	POPE	Model	Accuracy	Precision	Recall	F1 Score	F1 Score (Truth)	Yes (%)
MSCOCO	*Random*	LLaVA	50.47	50.24	**99.67**	66.80	68.65	99.20
		MiniGPT-4	73.77	79.25	64.40	71.06	78.86	40.63
		InstructBLIP	**86.60**	**80.74**	96.13	**89.29**	89.27	59.53
	Popular	LLaVA	50.00	50.00	**99.27**	66.50	67.72	99.27
		MiniGPT-4	67.80	**68.80**	65.13	66.92	72.21	47.33
		InstructBLIP	**71.27**	64.20	96.13	**76.99**	83.45	74.87
	Adversarial	LLaVA	49.77	49.88	**99.20**	66.38	66.98	99.43
		MiniGPT-4	61.93	**61.46**	64.00	62.70	71.37	52.07
		InstructBLIP	**62.53**	57.50	96.13	**71.96**	78.45	83.60

Table 5: SEEM-based POPE results of LVLM on MSCOCO. F1 Score (Truth) are the results of POPE using ground-truth annotations, which are copied from Table 3. The best results in each block are denoted in bold.

Scalability of POPE

Consistency of POPE

Data Collected from: InstructBLIP and MiniGPT-4:

- **"No" Responses**:
  - InstructBLIP: Out of **1303** objects with **"No"** responses, **0** were referenced in captions.
  - MiniGPT-4: Out of **1445** objects with **"No"** responses, only **5** were mentioned in captions.
- **"Yes" Responses**:
  - InstructBLIP: All **664** objects mentioned in captions received "**Yes**" verdicts.
  - MiniGPT-4: Out of **1034** objects mentioned in captions, **961** received **"Yes"** responses.

Consistency of POPE

POPE vs VQA Performance

Dataset	Model	POPE↑	VQA↑
A-OKVQA	InstructBLIP	**87.20**	**59.68**
	MiniGPT-4	72.47	38.69
	LLaVA	66.64	50.51
GQA	InstructBLIP	**85.32**	**62.12**
	MiniGPT-4	67.13	42.24
	LLaVA	66.56	47.60

Table 6: Evaluation results of LVLMs on POPE and VQA. For VQA tasks, we report the VQA score on A-OKVQA and Accuracy on GQA. For POPE, we copy the result under the random setting from Table 11.

POPE vs VQA Performance

Dataset	Model	POPE↑	VQA↑
A-OKVQA	InstructBLIP	**87.20**	**59.68**
	MiniGPT-4	72.47	38.69
	LLaVA	66.64	50.51
GQA	InstructBLIP	**85.32**	**62.12**
	MiniGPT-4	67.13	42.24
	LLaVA	66.56	47.60

Table 6: Evaluation results of LVLMs on POPE and VQA. For VQA tasks, we report the VQA score on A-OKVQA and Accuracy on GQA. For POPE, we copy the result under the random setting from Table 11.

POPE vs VQA Performance

Dataset	Model	POPE↑	VQA↑
A-OKVQA	InstructBLIP	**87.20**	**59.68**
	MiniGPT-4	72.47	38.69
	LLaVA	66.64	50.51
GQA	InstructBLIP	**85.32**	**62.12**
	MiniGPT-4	67.13	42.24
	LLaVA	66.56	47.60

Table 6: Evaluation results of LVLMs on POPE and VQA. For VQA tasks, we report the VQA score on A-OKVQA and Accuracy on GQA. For POPE, we copy the result under the random setting from Table 11.

MiniGPT-4 : instruction dataset of only derives from image caption data,
LLaVA: uses158K visual instructions data involving complex visual questions.

Limitations of the Study

1. **Narrow Scope**: Focuses only on object hallucination, not overall LVLM performance.
2. **Limited Dataset**: Evaluated on a small portion of the validation set, potentially skewing results.
3. **Answer Matching**: Relies on "Yes" or "No" answers, which models may not always provide explicitly.
4. **Annotation Inconsistencies**: Automatic segmentation tool labels may differ from human annotations.
5. **Few Models Evaluated**: Only a small number of LVLMs were tested, excluding newer models.

Conclusion and Key Takeaways

- **Evaluation of LVLMs**: We evaluated multiple LVLMs and identified their susceptibility to **object hallucination**.
- **Impact of Visual Instructions**: The object distributions in the visual instructions significantly influence hallucination behavior in LVLMs.
- **Limitations of Existing Methods**: Current evaluation methods can be unreliable, as they are affected by the input instructions and generated text.
- **Proposed POPE Method**: We introduced **POPE**, a polling-based query method, to provide more accurate evaluation of object hallucination.
- **Experimental Validation**: Results show that POPE offers better insights into object hallucination issues compared to traditional methods.

Future Direction

- Extending the analysis from coarse-grained to fine-grained object hallucinations such as the number, attributes, and positions of the object

- Evaluating object hallucination for more LVLMs.

- **Cross-Domain Generalization**: Test LVLMs on datasets from various domains to assess hallucination across different tasks.

Future Directions of MLLMs: Modularity vs. Pre-training

Modular Structures:

- Research focuses on replacing **black-box pretraining** with more **modular** models to enhance control, understanding, and faithfulness.

- **Causality and Counterfactual Reasoning**: Exploration into models like **Cm3** that incorporate causal and counterfactual reasoning in multimodal tasks.

Ghosh, A., Acharya, A., Saha, S., Jain, V., & Chadha, A. (2024). Exploring the frontier of vision-language models: A survey of current methodologies and future directions. *arXiv preprint arXiv:2404.07214*.
https://arxiv.org/pdf/2404.07214

Future Directions: Efficient and Domain-Specific Models

- **Training Efficiency**:
  - Efforts are underway to develop more **efficient multimodal models** like **BLIP-2**, which surpasses Flamingo-80B with fewer trainable parameters.

- **Domain-Specific VLMs**:
  - Specialized models such as **MedFlamingo** and **SkinGPT** are emerging in fields like **healthcare**, with more progress expected in sectors like **education** and **agriculture**.

Ghosh, A., Acharya, A., Saha, S., Jain, V., & Chadha, A. (2024). Exploring the frontier of vision-language models: A survey of current methodologies and future directions. *arXiv preprint arXiv:2404.07214*.
https://arxiv.org/pdf/2404.07214

Future Directions: Continuous Learning and Fine-Grained Evaluation

Continual Learning:

- Research on models that can **learn continuously** without retraining from scratch, inspired by LLM approaches.

Fine-Grained Evaluation:

- New evaluation metrics for bias and fairness

Ghosh, A., Acharya, A., Saha, S., Jain, V., & Chadha, A. (2024). Exploring the frontier of vision-language models: A survey of current methodologies and future directions. *arXiv preprint arXiv:2404.07214*.
https://arxiv.org/pdf/2404.07214