

Multi-modal LLMs

Boyu Yang, Liangyu Li, Yunchang Xie

Learning Transferable Visual Models From Natural Language Supervision

Contrastive Language-Image Pre-training (CLIP)

Authors: Alec Radford et al., 2021
Institute: OpenAI

Radford, Alec, et al. "Learning transferable visual models from natural language supervision." *International conference on machine learning*. PmLR, 2021.

Background

Two main directions of Visual Representation Learning before 2021

- **Weakly supervised models**
 - Use massive image datasets but fixed label spaces (e.g., ImageNet-1k).
 - Strong accuracy but poor flexibility — can't handle unseen categories.
 - Rely on static softmax classifiers → limited zero-shot capability.
- **Language-supervised models** (e.g., VirTex, ConVIRT)
 - Learn from image-caption pairs instead of discrete labels.
 - Capture richer semantics and stronger vision-language alignment.
 - However, trained on small datasets (hundreds of thousands of pairs) → lower performance.

Core limitation of both paradigms

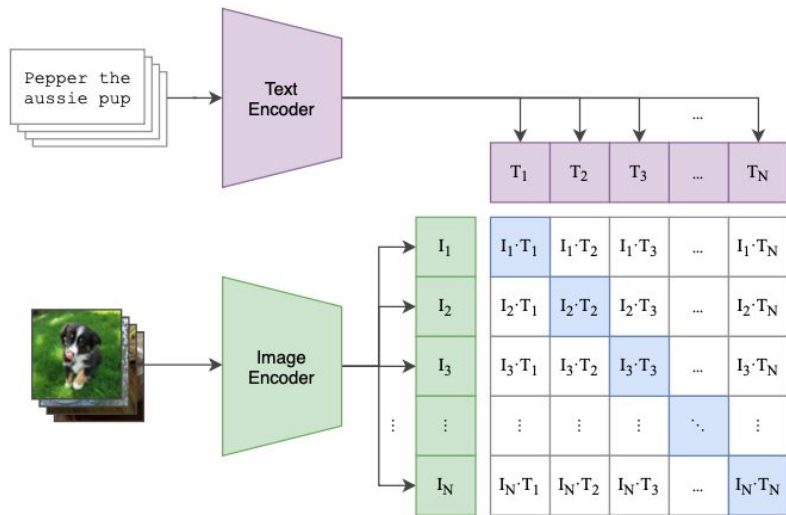
- Weakly supervised → *large-scale but narrow semantics*.
- Language-supervised → *rich semantics but small-scale*.

Motivation for CLIP

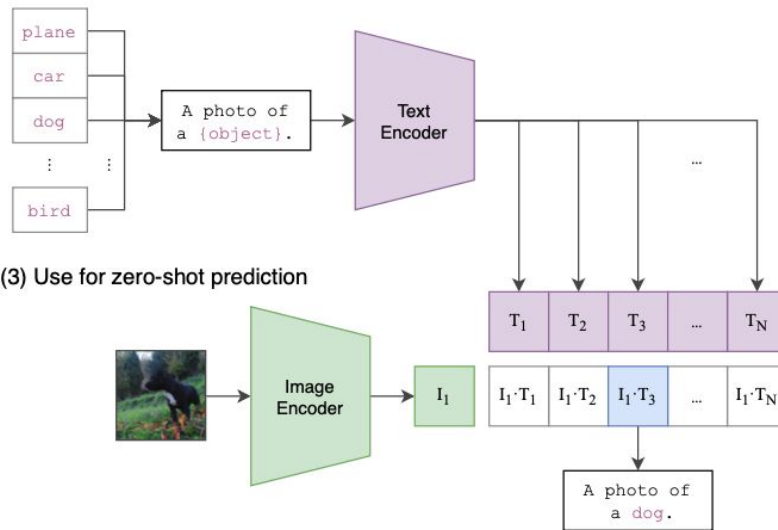
- Bridge the gap between *scale* and *semantic richness*.
- Scale natural-language supervision to **hundreds of millions** of image-text pairs.
- Introduce a **contrastive pre-training objective** → enables **zero-shot transfer** across diverse visual tasks.

CLIP: Summarized Approach

(1) Contrastive pre-training



(2) Create dataset classifier from label text



(3) Use for zero-shot prediction

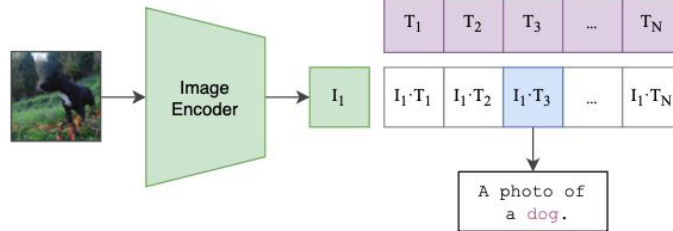


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

Data and Encoder

Data Collection

Data Volume: **400 million (image, text) pairs.**

Source: Collected from the internet (web pages, social media, etc.).

Text Type: Typically image titles, alt-text, and descriptive text.

Characteristics: Diverse data, high noise levels, but broad coverage.

CLIP does not manually clean tags; instead, it directly uses natural language descriptions as "supervisory signals."

Encoder Selection

Image Encoder:

- ResNet 50, RN101, EfficientNet-style-model (RN50x4, RN50x16, RN50x64)

(Global average -> attention pooling for better adapt to multiple semantic regions.)

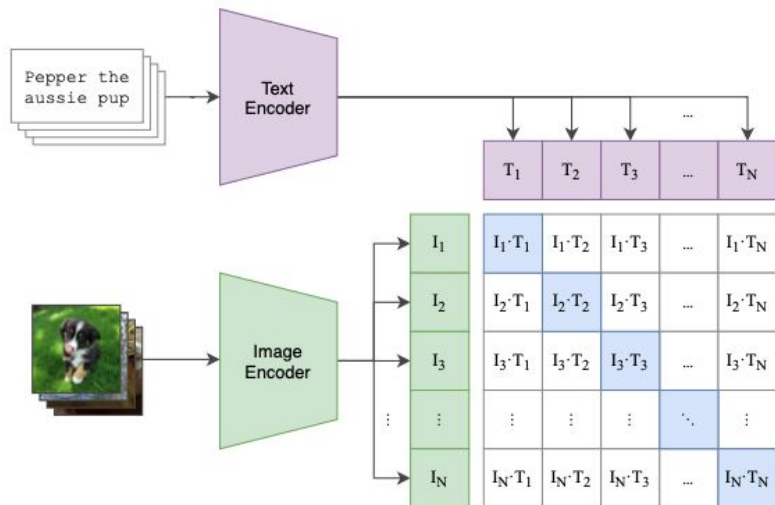
- ViT-B/32, ViT-B/16, ViT-L/14

Text Encoder:

- standard **Transformer**

Pre-training Methods

(1) Contrastive pre-training



```
# image_encoder - ResNet or Vision Transformer
# text_encoder - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l] - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T) #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

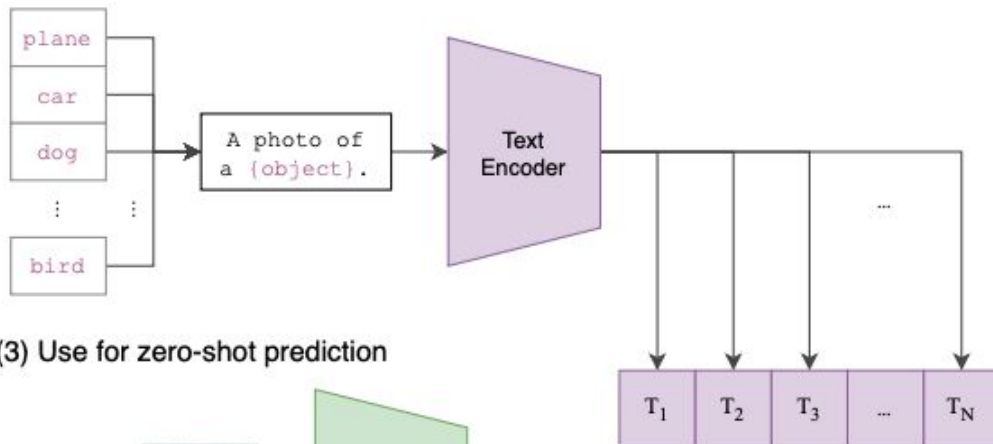
# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss = (loss_i + loss_t)/2
```

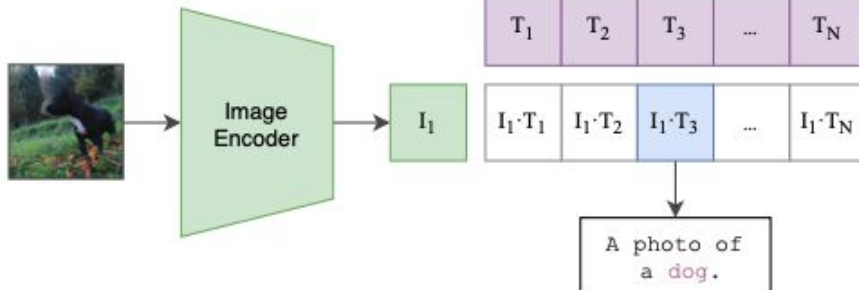
- The idea is simple: The positive pairs are just images with the associated textual descriptions: (I, T) ; the unpaired images and text are just negative pairs.
- Given a batch of B (image-text) pairs, we maximize the similarity between B correct pairs of image-text associations while minimizing the similarity for the other $B(B-1)$ incorrect pairs via a symmetric (along rows and columns) cross-entropy loss.

Inference

(2) Create dataset classifier from label text



(3) Use for zero-shot prediction



Example:

Input:

- choices (label): ["a", "b", "c"]
- Image

Model Output:

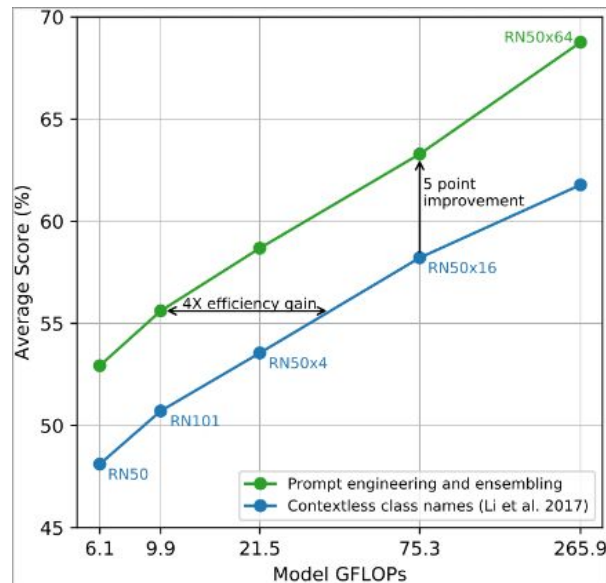
- The probability of this image that belongs to each label

Essentially, model is like solving **multiple choice problems** instead of prediction or generation problem

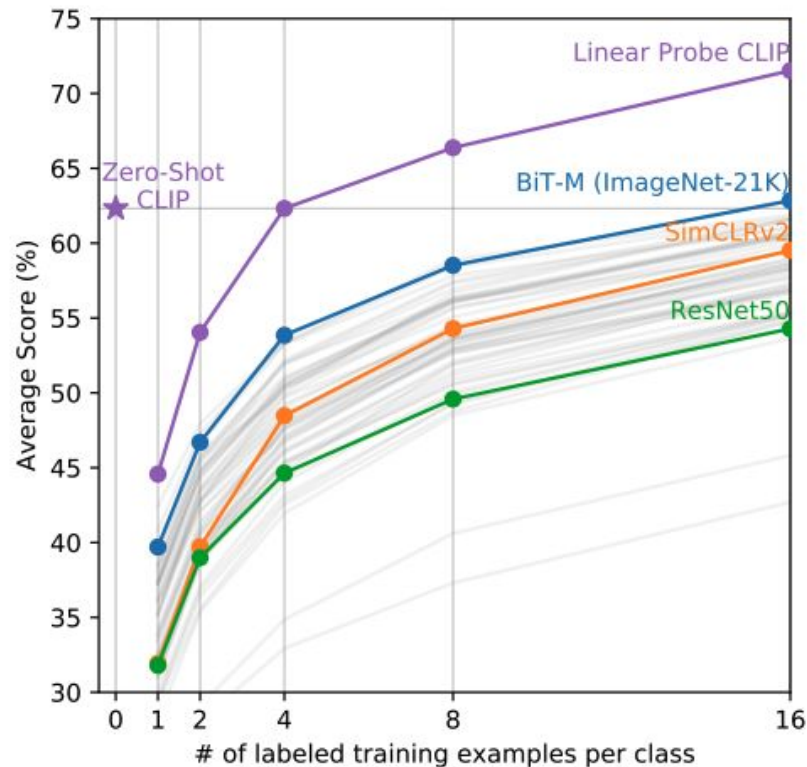
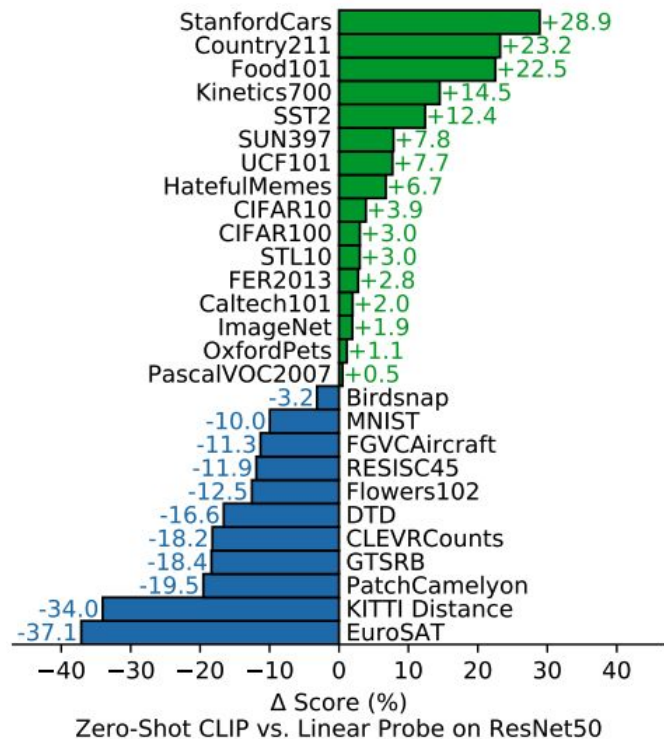
Prompting Engineering and Ensembling

- After CLIP is pre-trained, it performs **zero-shot classification** by comparing an image's embedding to text embeddings of possible class names
- **Problems:**
 1. Polysemy : e.g. ImageNet construction crane v.s. crane (animal).
 2. Distribution Gap
- **Solution:**
 - (a) create prompt template: "A photo of a {label}"
 - (b) Further customized prompt to specific task : "A photo of a {label}, a type pf pet"
 - (c) Prompt Ensembling:
Encode each one with CLIP's text encoder → obtain embeddings
→ average them to form a single, more stable class representation

```
imagenet_templates = [  
    'a bad photo of a {}. ',  
    'a photo of many {}. ',  
    'a sculpture of a {}. ',  
    'a photo of the hard to see {}. ',  
    'a low resolution photo of the {}. ',  
    'a rendering of a {}. ',  
    'graffiti of a {}. ',  
]
```

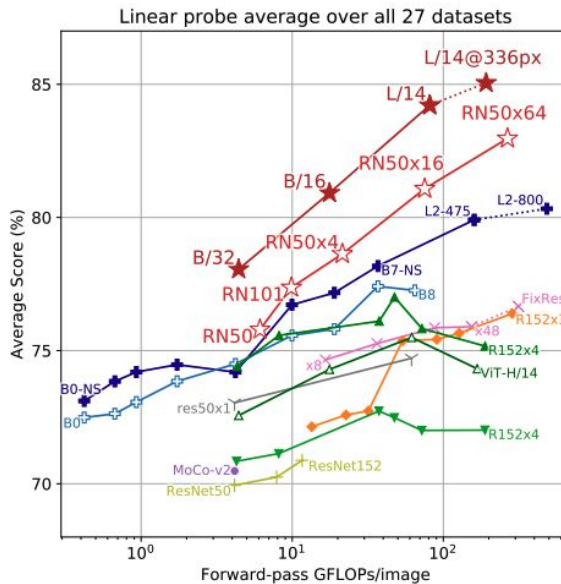
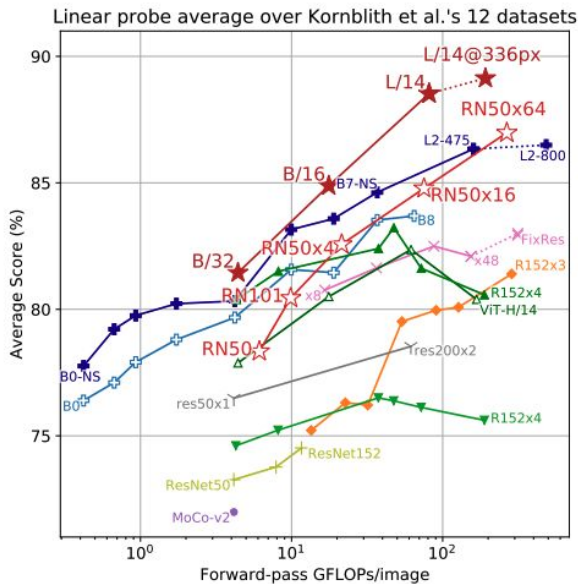


CLIP: Zero-shot and Few-shot Generalization



CLIP: Linear-Probe Transfer Learning

- Linear Probe Transfer: Linear Classifier on features extracted by CLIP.



CLIP: Robustness to Distribution Shift

	Dataset Examples	ImageNet ResNet101	Zero-Shot CLIP	Δ Score
ImageNet		76.2	76.2	0%
ImageNetV2		64.3	70.1	+5.8%
ImageNet-R		37.7	88.9	+51.2%
ObjectNet		32.6	72.3	+39.7%
ImageNet Sketch		25.2	60.2	+35.0%
ImageNet-A		2.7	77.1	+74.4%

Limitations

- Poor performance on fine-grained classification
 - differentiating between models of cars, species of flowers, and variants of aircraft.
- Poor Generalization on data that is truly out-of-distribution for it
 - worse classification on MINST even compared to logistic regression
- **CLIP performs zero-shot classification but can only choose from predefined concepts, limiting its flexibility compared to models that generate free-form descriptions.**
 - Multiple choice → QA
- Poor Data efficiency
- Not choosing State-of-art baseline models (e.g. Huge ViT..)

Visual Instruction Tuning

Summary of “LLaVA: Large Language and Vision Assistant”

Authors: Haotian Liu, Chunyuan Li, Qingyang Wu, Yong Jae Lee (2023)

Liu, Haotian, et al. "Visual instruction tuning." *Advances in neural information processing systems* 36 (2023): 34892-34916.

Motivation & Background

Text-only LLMs (ChatGPT, GPT-4) can follow instructions but cannot see images.



Vision models (like CLIP) can recognize objects but cannot understand language instructions.



build a general-purpose multimodal assistant that can follow vision + language commands

Overview of LLaVA

Vision Encoder: CLIP (ViT-L/14) extracts visual features Z_v .

Projection Layer W : Maps Z_v into Vicuna's embedding space.

Language Model f_ϕ : Vicuna 13B generates responses based on both image and text.

Inputs: Image (X_v) and Instruction (X_q);

Output: Language response (X_a).

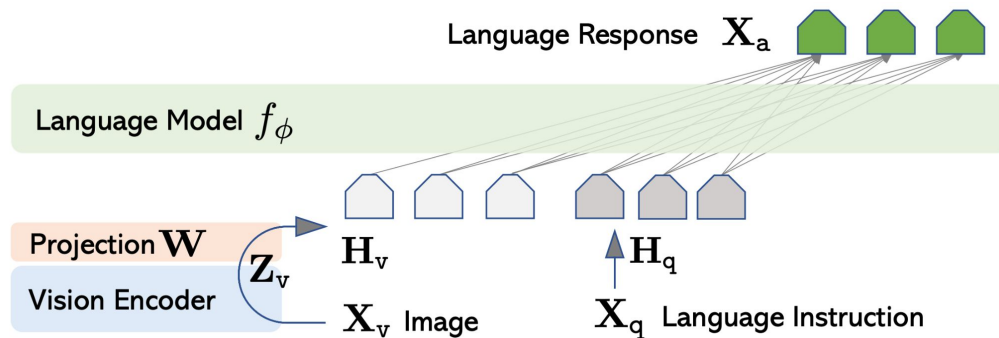


Figure 1: LLaVA network architecture.

Generating Multimodal Instruction Data

Major problem → no large dataset for image + instruction + answer.

Solution → use **GPT-4 as a teacher** to generate synthetic Q&A data.

Start from image–caption pairs, feed caption text to GPT-4, and ask it to create natural Q&A.

Three data types:

- 1. Conversation questions
- 2. Detailed descriptions
- 3. Complex reasoning

Total ≈ 158 K instruction samples

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>

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

Two-Stage Visual Instruction Tuning

Stage 1 – Feature Alignment:

- 595 K image–caption pairs (CC3M).
- Train projection W only to align visual features to text space.

Stage 2 – End-to-End Fine-Tuning:

- Freeze CLIP; train W + Vicuna on 158 K GPT-4-generated samples.
- Objective: next-token prediction given image and instruction.

$X_{\text{system-message}}$ **<STOP>**
Human : X_{instruct}^1 **<STOP>** Assistant: X_a^1 **<STOP>**
Human : X_{instruct}^2 **<STOP>** Assistant: X_a^2 **<STOP>** ...

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, we follow Vicuna-v0 [9] to set the system message $X_{\text{system-message}}$ and we set **<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.

LLaVA as a Visual Chatbot

Image shows a man ironing clothes on top of a taxi.

Question: “What is unusual about this image?”

LLaVA → correctly reasons that it’s unsafe and unusual.

BLIP-2 → just describes the scene.

OpenFlamingo → incorrect answer.

Demonstrates true instruction-following beyond captioning.



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

User	What is unusual about this image?
LLaVA	The unusual aspect of this image is a man ironing clothes on the back of a minivan or van. This is not a typical place to perform this activity, as one would usually iron clothes in a more stationary and safe location, such as a home, using a regular ironing board. The scene depicted in the image is peculiar as it involves a makeshift ironing setup on a vehicle, which can be both unsafe and unconventional. Additionally, it is not clear how the man is able to maintain balance and stability while ironing clothes in such an unstable environment. [Start a new conversation, and clear the history]
User	What’s happening in the scene?
LLaVA	The image depicts a man standing on top of a yellow SUV in a busy city street. He is holding a portable ladder, seemingly ironing clothes while standing on the vehicle. Around the scene, there are other cars, a traffic light, and a fire hydrant. Additionally, there are two other people in the vicinity, one close to the man on the car and another further away. The man on the ladder appears to be attracting attention due to his unconventional choice of ironing his clothes on top of a moving car . The city street around him is bustling with activity, adding to the unique nature of the scene.
User	What is unusual about this image?
GPT-4 [36]	The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.
User	What is unusual about this image?
BLIP-2	a man is sitting on the back of a yellow cab
User	What is unusual about this image?
OpenFlamingo	The man is drying his clothes on the hood of his car.

Table 3: Example prompt from GPT-4 paper [36] to compare visual reasoning and chat capabilities Compared to BLIP-2 [28] and OpenFlamingo [5], LLaVA accurately follows the user’s instructions, instead of simply describing the scene. LLaVA offers a more comprehensive response than GPT-4 Even when merely asked to describe the image, LLaVA identifies atypical aspects of the image.

Benchmarking on LLaVA-Bench

Two benchmarks:

- COCO subset (standard images)
- In-the-Wild (real diverse images)

Metrics: helpfulness, accuracy, and detail (scored by GPT-4).

Performance:

- LLaVA $\approx$ 85 % of GPT-4 on COCO
- +29 % vs BLIP-2, +48 % vs OpenFlamingo on In-the-Wild

	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.

	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.

Where LLaVA Still Struggles

Fine-grained text or logos in images → difficult for CLIP-based encoder.

Example 1: Ichiran Ramen — fails to read restaurant name.

Example 2: Fridge image — misidentifies “strawberry yogurt.”

Root cause: model treats image as disconnected patches, lacking global context.

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



ICHIRAN Ramen [source]

Annotation A close-up photo of a meal at **ICHI-RAN**. The chashu ramen bowl with a spoon is placed in the center. The ramen is seasoned with **chili sauce**, **chopped scallions**, and served with **two pieces of chashu**. Chopsticks are placed to the right of the bowl, still in their paper wrap, not yet opened. The ramen is also served with nori on the left. On top, from left to right, the following sides are served: a bowl of **orange spice** (possibly garlic sauce), a plate of **smoke-flavored stewed pork with chopped scallions**, and a cup of **matcha green tea**.



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 **Page non-fat yogurt**, and **one of the smaller cups is Page blueberry yogurt**. The brand and flavor of the other smaller cup are unknown. Towards the back, there is a container with an unknown content.

Question 1	What's the name of the restaurant?	What is the brand of the blueberry-flavored yogurt?
Question 2	Describe this photo in detail.	Is there strawberry-flavored yogurt in the fridge?

Table 6: Challenging examples from LLaVA-Bench (In-the-Wild), we provide extremely-detailed annotation for each image for an accurate evaluation. Some questions require the model to extract details from high resolution image and to have a broad knowledge coverage.

Multimodal Reasoning on Science QA

Dataset: 21 K multimodal science questions.

Baselines: GPT-3.5, LLaMA-Adapter, MM-CoT, etc.

LLaVA alone: 90.9 % accuracy.

LLaVA + GPT-4 (Judge mode): 92.5 % — new state of the art.

Shows strong complementarity between visual and language reasoning.

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

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.

Ablation Studies and Key Insights

Using pre-last-layer CLIP features $\rightarrow +0.96$ % gain.

“Reasoning-first” training converges faster.

Removing Stage 1 pre-training $\rightarrow -5.1$ % accuracy drop.

Larger Vicuna (13 B vs 7 B) $\rightarrow +1$ % gain.

Visual features	Before	Last
Best variant	90.92	89.96 (-0.96)
Predict answer first	-	89.77 (-1.15)
Training from scratch	85.81 (-5.11)	-
7B model size	89.84 (-1.08)	-

Table 8: Design choice ablations (%). The difference with the best variant is reported in red text.

Takeaways and Future Directions

First work to apply *instruction tuning* to vision + language.

Uses GPT-4 as a data generator and teacher.

LLaVA achieves ~85 % of GPT-4 multimodal performance while fully open-source.

SOTA on Science QA (92.5 %).

Future: higher resolution, richer tasks, and real user interactions.

LLaVA: Large Language and Vision Assistant

Visual Instruction Tuning

NeurIPS 2023 (Oral)

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

► University of Wisconsin-Madison ► Microsoft Research ► Columbia University

*Equal Contribution



🔥[NEW!] LLaVA-1.5 achieves SoTA on 11 benchmarks, with just simple modifications to the original LLaVA, utilizes all public data, completes training in ~1 day on a single 8-A100 node, and surpasses methods that use billion-scale data.

LLaVA represents a novel end-to-end trained large multimodal model that combines a vision encoder and Vicuna for general-purpose visual and language understanding, achieving impressive chat capabilities mimicking spirits of the multimodal GPT-4 and setting a new state-of-the-art accuracy on Science QA.

NExT-GPT:Any-to-AnyMultimodal LLM

Authors: Shengqiong Wu et al., 2024

Wu, Shengqiong, et al. "Next-gpt: Any-to-any multimodal llm." *Forty-first International Conference on Machine Learning*. 2024.

What the paper claims

- **Goal:** Build one LLM-centric system that can take any mix of inputs (text/image/video/audio) and output any mix back—“any-to-any.”
- **Key idea:** Keep powerful modality encoders/decoders frozen and only train small projection layers + light LLM adapters. They report only ~1% of parameters are updated.
- **Training tricks:**
 - (1) Encoding-side alignment so non-text modalities become “LLM-understandable”;
 - (2) Decoding-side alignment so the LLM can trigger image/audio/video decoders via learned signal tokens;
 - (3) MosIT instruction tuning for multi-turn, modality-switching dialogues.

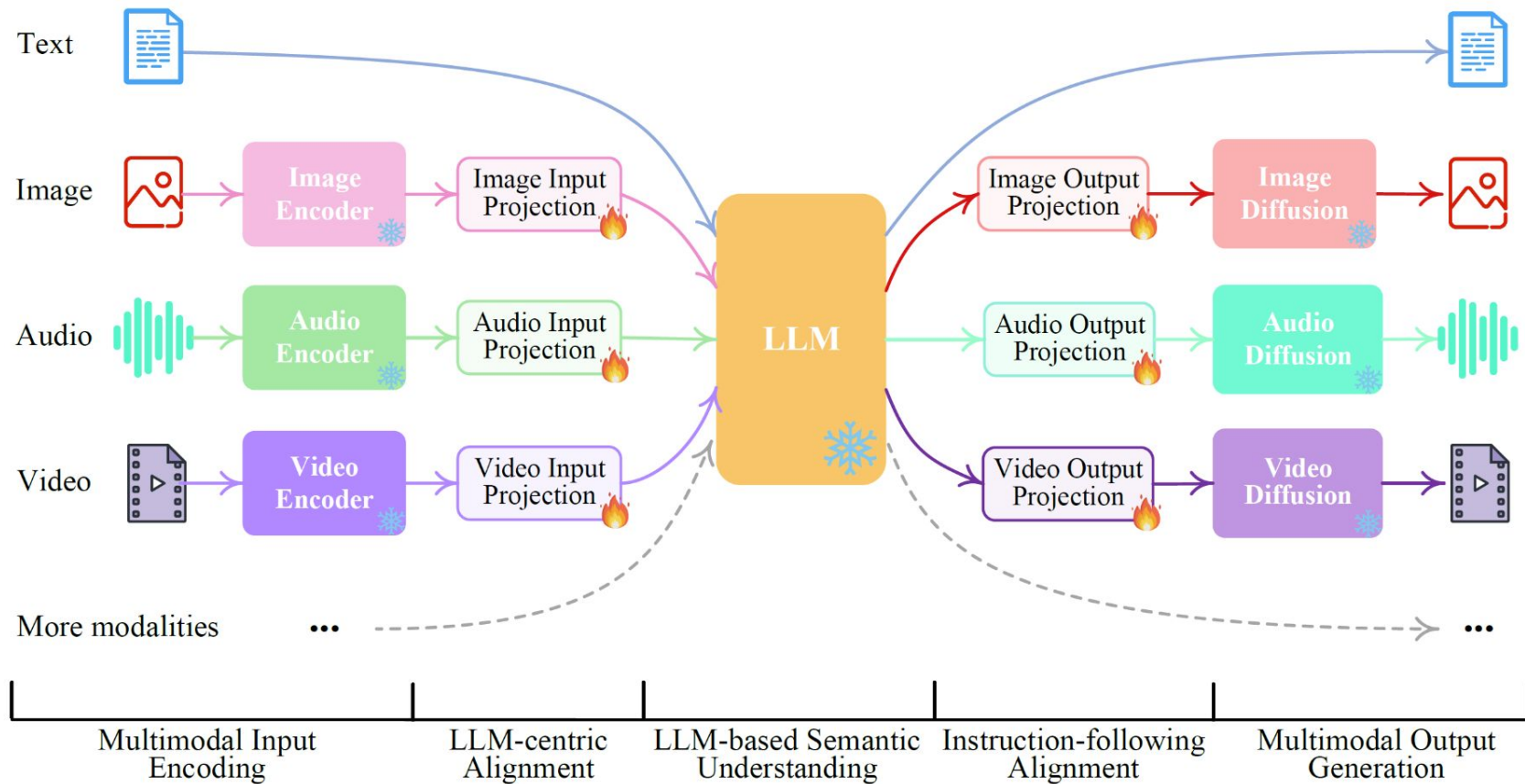


Figure 1. By connecting LLM with multimodal adaptors and diffusion decoders, NExT-GPT achieves universal multimodal understanding and any-to-any modality input and output. ❄️ and 🔥 represent the frozen and trainable modules, respectively.

Core method

1. **Multimodal encoding.** Uses **ImageBind** (unified encoder) for image/audio/video; a small **input projection** maps each modality into a **language-like** space the LLM can read.
2. **LLM reasoning.** Core is **Vicuna-7B**. The LLM emits (a) normal **text** and (b) special **modality signal tokens** that tell which generators to fire.
3. **Multimodal generation.** Output projection adapts those signal tokens for the appropriate **diffusion decoders**: **Stable Diffusion** v1.5 (images), **Zeroscope** (video), **AudioLDM** (audio). Decoders are kept frozen.

Core method

Table 1. Summary of NExT-GPT system configuration. Only 1% of parameters need updating during fine-tuning.

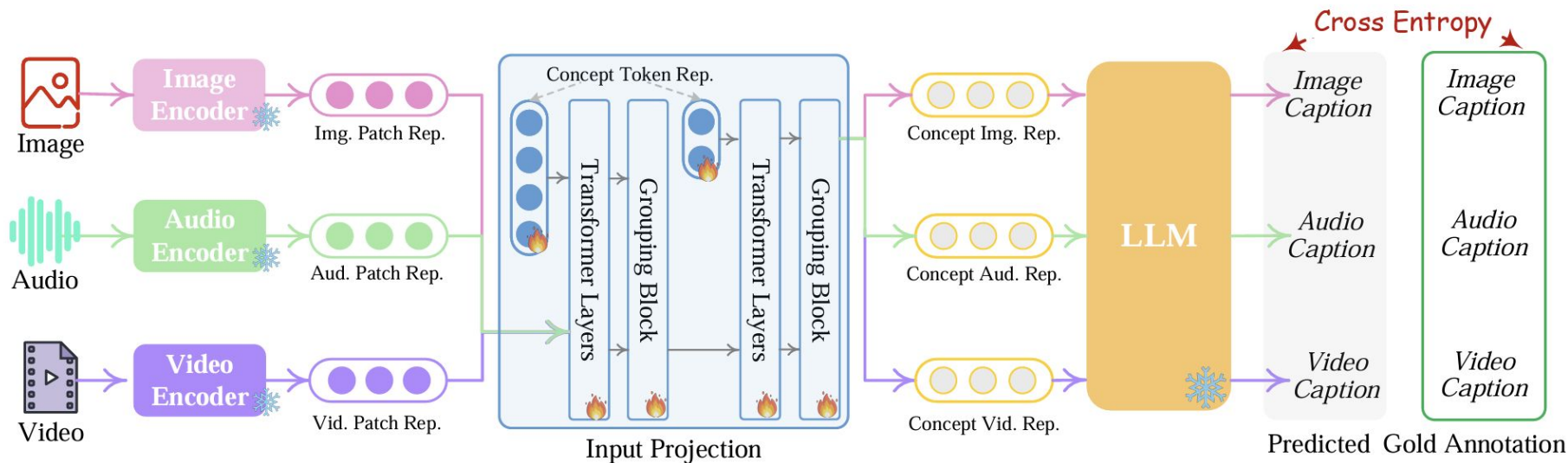
	Encoder		Input Projection		LLM		Output Projection		Diffusion	
	Name	Param	Name	Param	Name	Param	Name	Param	Name	Param
Text	—	—	—	—			—	—	—	—
Image	ImageBind	1.2B❄️	Grouping	28M🔥	Vicuna	7B❄️	Transformer	31M🔥	SD	1.3B❄️
Audio					(LoRA	33M🔥)	Transformer	31M🔥	AudioLDM	975M❄️
Video							Transformer	32M🔥	Zeroscope	1.8B❄️

What's actually trained?

Only the **input** and **output projection stacks** (plus tiny LoRA on the LLM during instruction tuning).

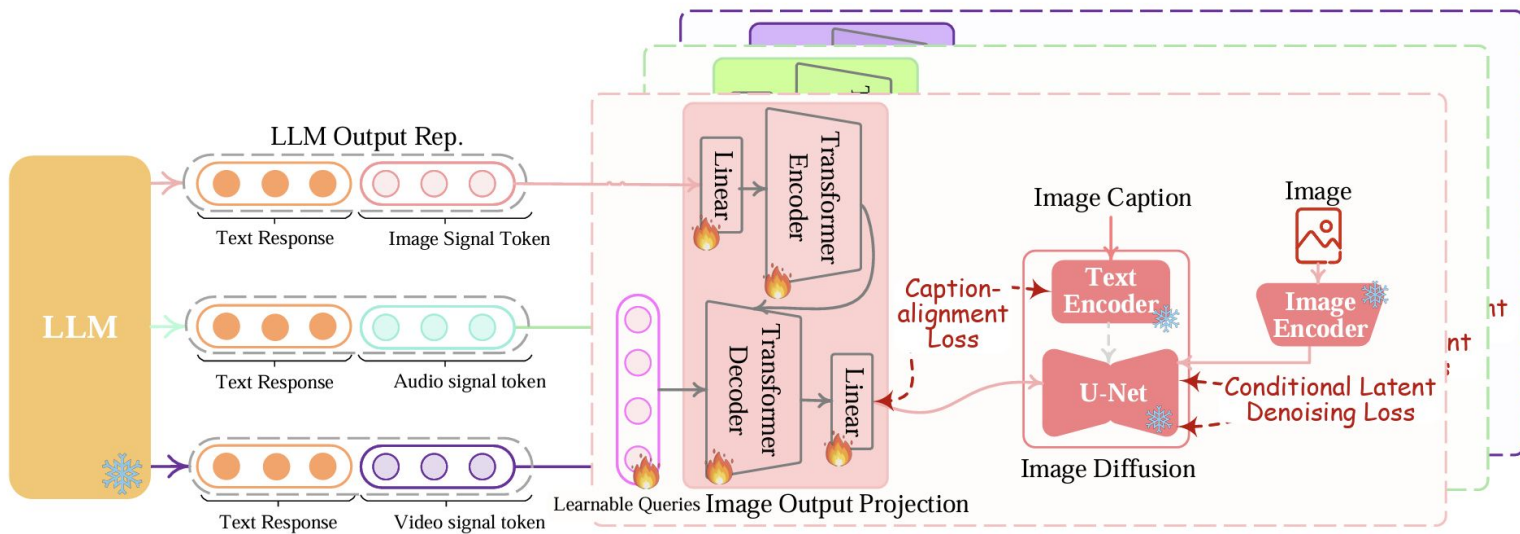
Encoding-side LLM-centric Alignment

- Problem: encoders give **patch/grid** features that don't line up with token semantics.
- Fix: a **grouping mechanism** aggregates patches into higher-level **concept tokens** before feeding the LLM.



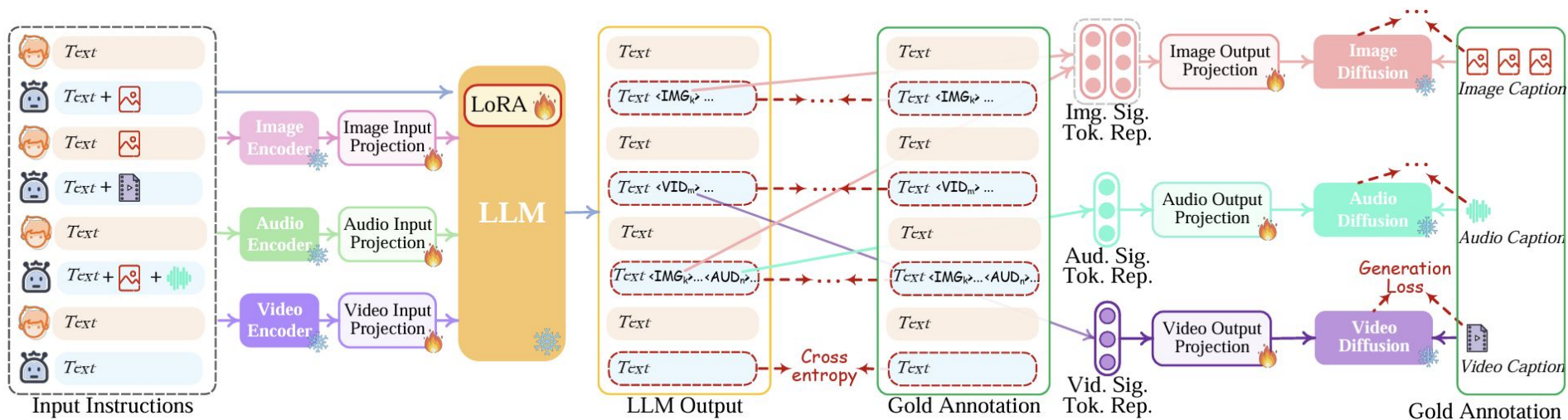
Decoding-side Instruction-following Alignment

- The LLM learns to output **signal tokens**—[IMGi], [AUDi], [VIDi]—which mean “generate image/audio/video now.”
- Those **soft token reps** condition the frozen diffusion models; training also pulls token reps toward the text-condition embeddings used by the decoders



Modality-switching Instruction Tuning

MosIT = curated **multi-turn, modality-switching dialogs** covering >100 topics. Built from **caption corpora** (COCO/WebVid/AudioCaps), templated prompts via **GPT-4**, and paired media, total **~5K dialogs**.



Zero-shot evaluation of image captioning with CIDEr($\uparrow$) score

Table 2. Zero-shot evaluation of image captioning with CIDEr ($\uparrow$) 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	<u>123.1</u>	82.4	102.2	-	33.4	33.9	36.0	-
LLaVA (Liu et al., 2023b)	LLaMA-2-7B-Chat	120.7	<u>82.7</u>	-	-	-	-	36.2	-
mPLUG-Owl (Ye et al., 2023b)	LLaMA-7B	117.0	80.3	<u>119.3</u>	-	39.0	-	46.6	<u>34.0</u>
Emu (Sun et al., 2023)	LLaMA-7B	-	-	<u>117.7</u>	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	<u>48.1</u>	-	60.9	-
NExT-GPT	Vicuna-7B	123.7	84.5	124.9	<u>66.7</u>	48.4	52.1	<u>58.0</u>	57.5

The table shows that NExT-GPT (Vicuna-7B) matches or outperforms previous open-source multimodal models on most zero-shot benchmarks, especially for image captioning (NoCaps, COCO) and several VQA tasks. Overall, it achieves the strongest or second-strongest results across captioning, question answering, and comprehensive evaluations.

Comparison of video reasoning

Table 3. Comparison of video reasoning tasks on MSRVT (Xu et al., 2016), MSVD-QA and MSRVT-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	MSRVT-QA	NExTQA	AudioCaps
Codi (Tang et al., 2023)	-	<u>74.4</u> *	-	-	-	<u>78.9</u> *
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	-	<u>29.6</u>	-
Video-LLaVA (Lin et al., 2023)	Vicuna-7B	-	70.7	<u>59.2</u>	-	-
Emu (Sun et al., 2023)	LLaMA-7B	-	32.4	14.0	6.8	-
NExT-GPT	Vicuna-7B	76.2 *	<u>64.5</u>	61.4	50.7	81.3 *

NExT-GGPT gets the best video captioning and audio captioning scores, and it also performs strongly on video QA, beating most earlier models.

Results on text-to-image/audio/video generation

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

Model	Image	Audio	Video
	FID (↓)	FAD (↓)	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 [†]	<u>11.18</u>	<u>1.74</u>	<u>30.96</u>

Human Evaluation

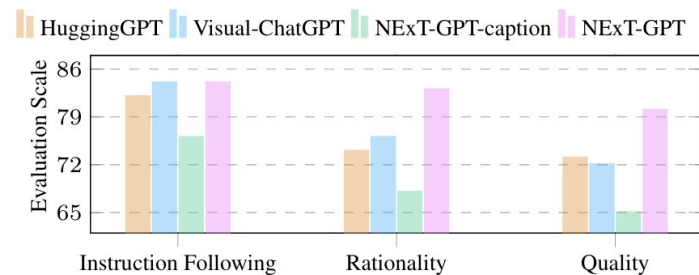


Figure 5. Human Evaluation (1-100 scale, results are on average) of NExT-GPT in comparison with pipeline baselines.

Humans rated NExT-GPT highest overall in following instructions, reasoning, and response quality compared with the pipeline baselines.

NExT-GPT generates the best overall images, audio, and videos among the compared models (lowest FID/FAD, highest CLIPSIM). Even its zero-shot version stays very competitive.



What is the unusual part of this video?





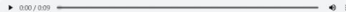
The unusual part of this video is that a dog is riding a skateboard. It is not common to see a dog on a skateboard, as dogs are typically not able to ride skateboards or bicycles. This scene is unique and interesting because it showcases a dog's ability to adapt and participate in human-made activities, demonstrating its intelligence and adaptability.



That is pretty cool. How about adding some music to make this scene look more funny?



Certainly! Here's a synthesized music track that could enhance the humor and lightheartedness of the scene:

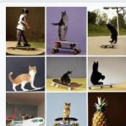




Do you have any other similar unusual scenes? Show them.



I have a set of such kinds of scenes:



(A)



I am in a bad mood, so bad so sad.



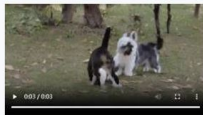
I am sorry to hear about that. Sometimes, it is hard to shake off a bad mood. Have you tried listening to some lifting music or watching a comedy to cheer you up?



Can you show something to cheer me up?



Sure, here's a video of a cute puppy playing in the park. It always makes me feel better.



(B)



I need to prepare my presentation for my history class, can you give some inspiration?



Of course! History is a fascinating subject. One approach you could take is to create a timeline presentation. You can use images, videos, and text to show the progression of events and highlight key moments in history. ...



Can you show me some examples?



Certainly! Here are a few examples of timeline presentations and historical figures :1.2.3.



(C)

Figure 4. Qualitative examples showcasing the interpretative and generative capabilities of NExT-GPT across diverse modalities or their combinations.

Evaluating Object Hallucination in Large Vision-Language Models

Authors: Yifan et al., 2023

Li, Yifan, et al. "Evaluating object hallucination in large vision-language models." *arXiv preprint arXiv:2305.10355* (2023).

Why Study Object Hallucination in LVLMs?

- **LVLMs** = LLMs + Vision Encoders
→ powerful multimodal understanding.
- Problem: **Object hallucination** = LVLM generates objects not present in the image.
- Consequence: Degrades accuracy and reliability in real-world applications (e.g., autonomous driving).
- **Goal:** Systematically evaluate object hallucination in modern LVLMs.

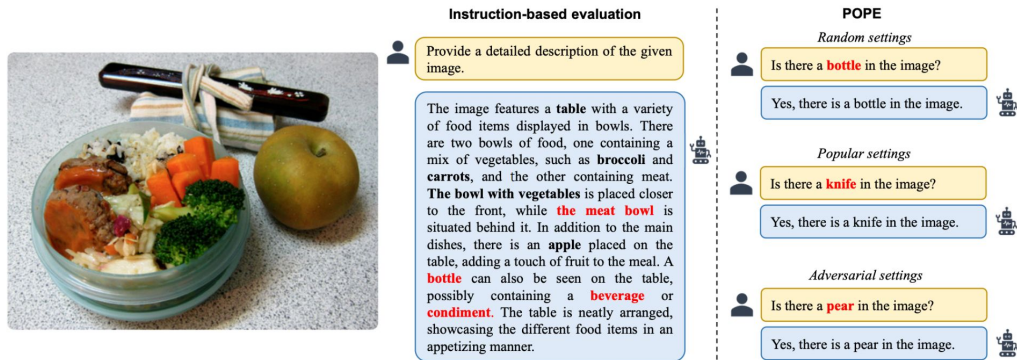


Figure 1: Cases of object hallucination in LVLMs. **Bold** objects are ground-truth objects in the annotations and **red** objects are hallucinated objects by LVLMs. The left case is from the traditional instruction-based evaluation method, and the right cases are from three variants of POPE.

Limitation of CHAIR

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 VLPs 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 VLPs (OSCAR, VinVL, BLIP, and OFA) are collected from Dai et al. (2023b). The best results in each block are denoted in bold.

CHAIR (Caption Hallucination Assessment with Image Relevance)

Measures the proportion of objects mentioned in the generated captions that do **not** actually exist in the image.

-CHAIR_I: (# Hallucinated Obj Instances in Caption) / (Total Obj Instances)

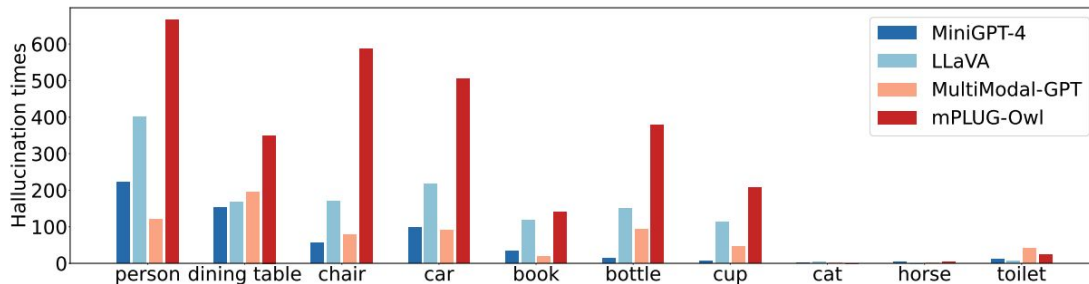
-CHAIR_s: (# Captions with ≥ 1 Hallucination) / (Total Captions)

Findings:

- Most LVLMs exhibit **more severe hallucination** than traditional VLPs (e.g., BLIP, VinVL).
- The evaluation results are **highly sensitive to the prompt**—even the same model can show **doubled hallucination rates** under different prompts.
- **Caption length** also affects the metric, making the evaluation **unstable**

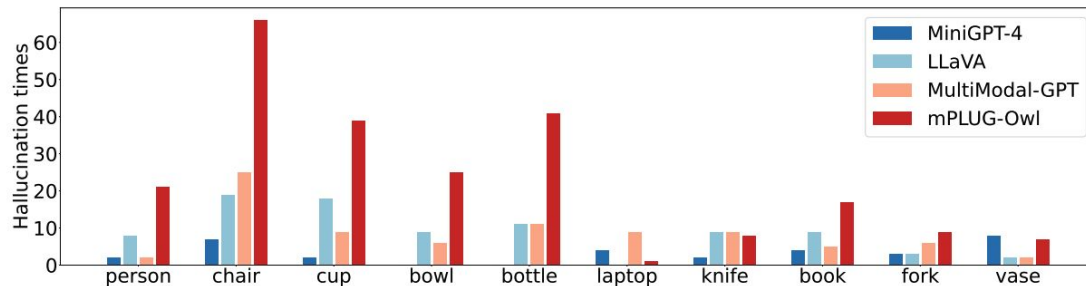
What cause object hallucination?

Unbalanced object distribution



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

Frequently co-occurring object groups



(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.

Polling-based Object Probing Evaluation (POPE)

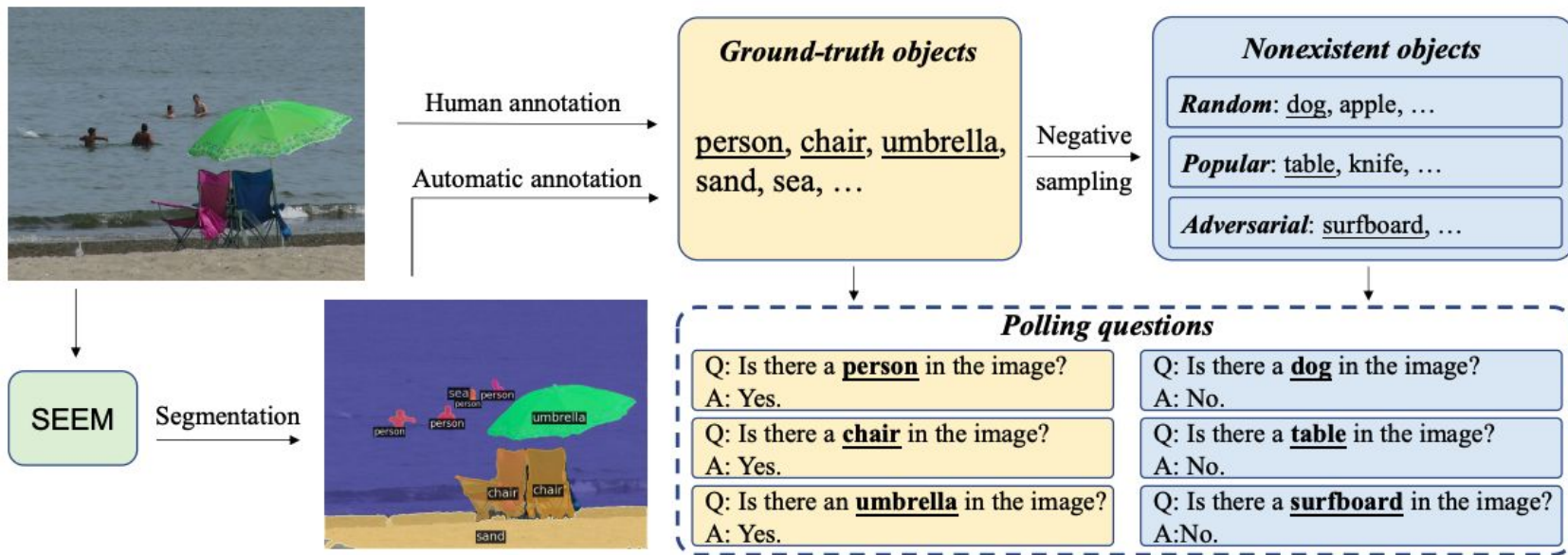


Figure 3: Overview of the POPE pipeline. Given an input image, POPE first extracts ground-truth objects in the image either from human annotations or with the help of automatic segmentation tools like SEEM. Then, POPE conducts negative sampling for nonexistent objects in the image under Random/Popular/Adversarial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

Negative Sampling in POPE

Three Strategy of Sampling

- Random sampling: randomly sample objects that do not exist in the image
- Popular sampling: select top k most frequently occurring objects in the dataset but not present in the image
- Adversarial sampling: select top k most frequently co-occurring objects with the ground-truth



POPE

Random settings



Is there a **bottle** in the image?

Yes, there is a bottle in the image.



Popular settings



Is there a **knife** in the image?

Yes, there is a knife in the image.



Adversarial settings



Is there a **pear** in the image?

Yes, there is a pear in the image.



Evaluation on MSCOCO

Dataset	POPE	Model	Accuracy	Precision	Recall	F1 Score	Yes (%)
MSCOCO	<i>Random</i>	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
	<i>Popular</i>	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
	<i>Adversarial</i>	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.

Setup:

- Built POPE on the **MSCOCO validation set**.
- Selected **500 images** with > 3 ground-truth objects.
- Constructed **6 Yes/No questions** per image.

Findings:

- **InstructBLIP** achieves the **highest F1 score**, showing the least hallucination.
- Performance **drops from Random → Popular → Adversarial**

Advantages of POPE: Stability

- POPE is less sensitive to instruction format.

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.

Advantages of POPE: Scalability

- Extend POPE to datasets without manual annotations.
- Model performance trends remain consistent with annotation-based POPE
- A **performance gap** appears (e.g., MiniGPT-4: F1 71.37 $\rightarrow$ 62.70)
due to **finer-grained segmentation** from SEEM, which makes the task harder.

Dataset	POPE	Model	Accuracy	Precision	Recall	F1 Score	F1 Score (Truth)	Yes (%)
MSCOCO	<i>Random</i>	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
	<i>Popular</i>	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
	<i>Adversarial</i>	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.

Advantages of POPE: Consistency

- Whether the Yes/No reflects the model's perception of objects?

Model	InstrcutBLIP	Mini-GPT-4
Number of "Yes"	664	961
Number of objects in captions	664	1034
Number of "No"	1303	1445
Number of objects in captions	0	5

Limitations

- Due to limited computational resources, experiments were conducted on **subsets of the validation data**, which may cause performance fluctuations arising from **data distribution biases**.
- The use of **automatic segmentation tools** may introduce **inconsistencies in object labeling** compared with human annotations, potentially affecting evaluation accuracy.
- Because **POPE relies on keyword matching** to interpret LVLM responses, evaluation errors can occur when **models do not explicitly output the expected terms**.

