# Visual Instruction Tuning

Published by: Haotian Liu, Chunyuan Li, Qingyang Wu, Yong Jae Lee - Dec 2023

Presented by Tejus Singh

# Instruction Tuning - Form of fine-tuning (occurs after pretraining)

- Standard fine-tuning: You feed in a sequence; loss applies to all tokens
  - **Result: Model learns to continue the text in a task-specific domain**
  - Example:
    - **During training**: Input: “The cat is black”
    - **During inference**: “The cat is” → Model output: “black”

- Instruction tuning: Training examples contain instruction, input, and desired output; Loss only applies to the output tokens.
  - **Result: Model maps instructions to appropriate outputs**
  - Example:
    - **During training**: Input: “Summarize the following: I ate so much, and then went to the mall, and then when I got back I slept at home. Response: I ate, went to the mall, slept at home.”
    - **During inference**: “Summarize the following: This cat here is black, and it likes to eat fish” → Model output: “This black cat eats fish.”

# Visual Instruction Tuning

- Vision models are trained on caption-style data and cannot follow instructions
- Language models are useful in following instructions but are blind to images

- Solution: Treat images as part of the input and train model on instruction-response pairs that requires understanding of both image and text

Example:

Training: Input “What is the person doing?” + Image, Response: “Person is riding a bike.” Model learns to output the response to the input.

- This results in an instruction-following model that can align both text and image representations in the same pipeline.

# Tangent on Data Generation

Problems

- Data for multi-modal instruction-following tasks is difficult to source.
- Separately, image-text contrastive models (like CLIP) could provide text captioning for images. LLMs like GPT-4 could perform reasoning from text.

Solutions

- They used the COCO dataset which provided captions $X_C$ for images $X_V$. The dataset also included bounding box annotations for objects.
- They then fed $X_C$ and bounding box annotations to GPT-4, prompting it to generate interesting questions about the scenario described in the image.

# Creating text prompts from captions via GPT-4

Starting with an **image, caption, bounding box** info from COCO:

- They constructed prompts to GPT-4 in an instruction-following conversation.
  - Prompt to gpt: “Given following image description: <Man riding on bicycle at night with cars nearby.> Design a conversation between you and a person asking about this photo. Ask diverse questions and give corresponding answers.
  - GPT-4 output: “Conversation: Q: …. A ….”

- Dataset: each training sample is a triplet containing <image, instruction(s), answer(s)>
- Then the LLM is trained on these triplets, to output the answers.

# LLaVA Architecture

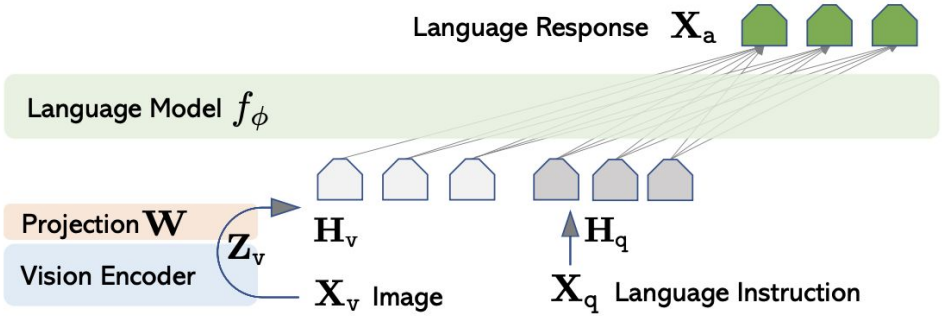


Figure 1: LLaVA network architecture.

- They uses Vicuna as their base LLM, and pretrained CLIP visual encoder.
- CLIP encoded images $X_V$ into set of visual feature vectors $Z_V$
- Using a learned Matrix W, $Z_V$ was brought into Vicuna's word embedding space.

$$H_V = W\, g(X_V) \text{ where } g(X_V) \text{ gives us } Z_V$$

- Finally, $H_V$ and $H_q$ were concatenated together. From there everything is standard transformer stuff.

Note, Vicuna is an instruction-tuned variant of LLaMA

# Learning Objective for visual fine-tuning

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

- $X_{instruct}$ contains the image data $X_V$ and questions $X_q$
- Calculate P(getting the right answer $X_i$ given image data $X_V$, all previous instructions (including current) $X_{instruct,<i}$, and all previous answers $X_{a,<i}$)
- L denotes the total number of tokens in each sequence, once the input has been tokenized.
- The model only outputs answer tokens (and an <end> tag) and only these contribute to the loss.

In the objective function, why is the probability conditioned on previous answers?

# Perform Instruction tuning in two steps

Stage 1: “Pre-training” for feature alignment

- The LLM and visual encoder weights are kept frozen. Projection matrix W is trained to maximize likelihood (from prev slide)
  - Remember, W is used to bring the encoded image features into Vicunaa’s word embedding space.

Stage 2: “Fine-tuning”

- The LLM weights and W are trained to maximize likelihood

Notice, the visual encoder was never trained. Why?

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. |
| 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. |

# Evaluation of LLaVA

Methodology:

1. Create <image, image text description, question>.
2. Get candidate model answers from <image, question>.
3. Generate “benchmark answers” using text-only gpt-4 with <image description, question>.
4. Use gpt-4 as a judge. Input: description, question, (candidate answer + benchmark answer). Output: Scores (helpfulness, relevance, accuracy, detail, 1-10)
   a. Note: Judge does not see the images, only the textual image descriptions

*Comparison with other vision-language models*

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

# (Fatal?) Flaws

- Firstly, getting the ground truth answer from gpt-4 and then using gpt-4 as a judge is flawed unless you hold gpt-4 as a benchmark for question answering.
- More importantly, the data LLaVA trained on was also generated by gpt-4 so when its responses are evaluated by a gpt-4 judge and compared to a gpt-4 benchmark answer…

If we were looking to answer: “Does LLaVA approximate gpt-4?” then the evaluation is appropriate.

If we were trying to answer: “Is LLaVA a good vision-language reasoning model?” the evaluation doesn’t work well for this.

# Visual Reasoning Results

| | 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) |

Relative scores wrt gpt-4 on LLaVA-bench

Weaknesses:

- Small dataset prevented LLaVA from learning nuanced details about specific products or items
- Sometimes, LLaVA cannot identity complex details, and treats images more like bags of features.

# Scientific Reasoning Results

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

- Tested on ScienceQA, where some tasks contained images, others didn't, task is to get the right answer for the posed science question.
- gpt-4 failed to provide response where image analysis was required, so they complemented with LLaVA (LLaVA + GPT-4 complement)
- When LLaVA and gpt-4 disagree, they ask gpt-4 again, to choose a final answer using both model outputs (LLaVA + GPT-4 judge)
- Chain-of-Thought variant reduced training time by half but didn't improve results.

# Takeaways

- One of the first instruction-tuned multimodal model using generated data
- Strong multi-modal instruction following model
  - Could follow instructions, reason, follow conversation, etc
- Simple architecture (vision encoder + Vicuna base LLM)
- Efficient way to generate data
- Open sourced the research
- Strong results on Q and A benchmarks and image based reasoning tasks (relative to GPT-4-based evaluation)