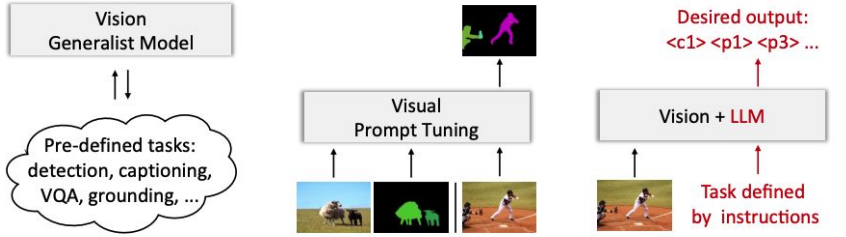
VisionLLM & VisionLLMv2

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

- Introduction
  - Problem
    - Pre-training -> Fine-tuning Paradigm
    - Visual Prompt Tuning
      - Segment Anything Model
    - Vision Language Model
      - Not vision-centric tasks
      - Generate next token
  - Solve
    - Address vision-centric tasks(object detection, segmentation segmentation ~~)
    - Instruction based vision-centric tasks available

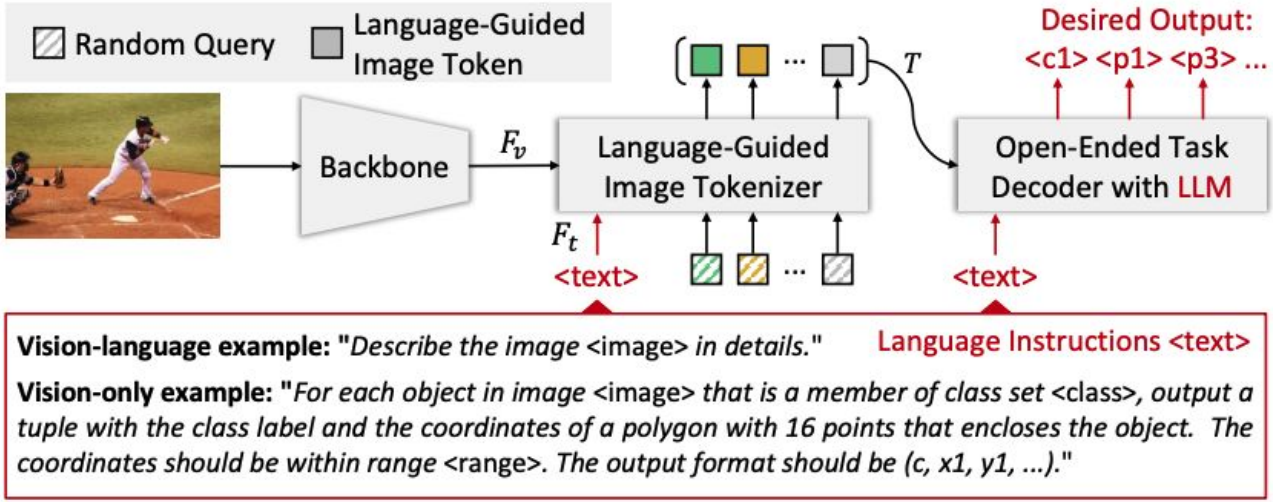


(a) Vision generalist models [59, 61, 83] are constrained by the format of pre-defined tasks. (b) Visual prompt tuning [26, 64, 62] are inconsistent with the format of LLMs. (c) VisionLLM (ours) can *flexibly manage vision-centric tasks using language instructions like LLMs*.

Figure 1: **Comparison of our VisionLLM with popular paradigms.** Unlike current vision generalist models that depend on pre-defined task formats and visual prompt tuning models that are inconsistent with large language models (LLMs), VisionLLM leverages the power of LLMs for open-ended vision tasks by using language instructions.

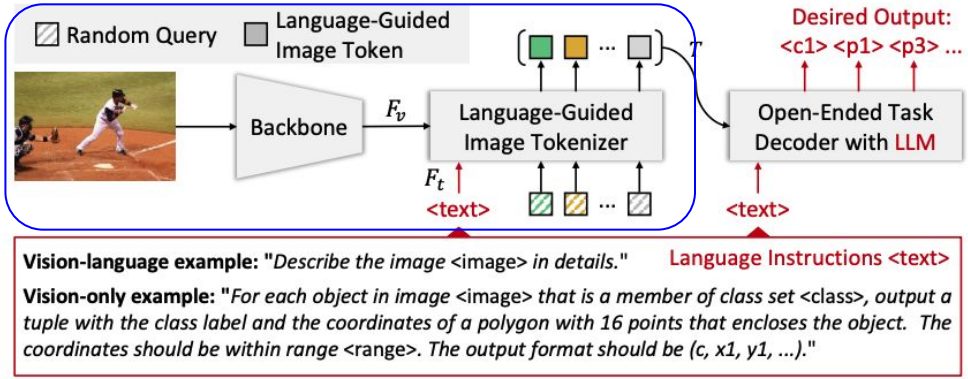
VisionLLM: Large Language Model is also an Open-Ended Decoder for Vision-Centric Tasks

- Overall Architecture
  - Backbone
  - Language Guided Image Tokenizer
  - Open-Ended Task Decoder with LLM



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

- Image Tokenizer
  - Compose
    - Backbone
    - Language Guided Image Tokenizer
  - Same with DETR Architecture
    - Backbone - Resnet50
    - Language Guided Image Tokenizer - Transformer Encoder
      - Query = BERT(Token) + nn.Parameter(n, dim)
      - Key, Value = Output of Resnet50



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

- Image Tokenizer
  - Compose
    - Backbone
    - Language Gui
      Image Tokeniz
  - Same with DETR Ar
    - Backbone - Re
    - Language Gui
      - Query =
      - Key, Val

VisionLLM considers images as a kind of foreign language and converts them into token representations. Unlike previous works [17, 60, 34] that utilize fixed-size patch embeddings to represent images, we introduce the language-guided image tokenizer to flexibly encode visual information that aligns with task-specific language prompts or instructions.

Specifically, give an image $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ with height H and width W, we first feed it to the image backbones (*e.g.*, ResNet [23]) and extract visual features F_v of four different scales. Additionally, we leverage a text encoder (*e.g.*, BERT [16]) to extract the language features F_l from given prompts. The language features are then injected into each scale of visual features through cross-attention [55], yielding multi-scale language-aware visual features, enabling the alignment of features across modalities.

Afterward, we propose to adopt a transformer-based network (*e.g.*, Deformable DETR [82]) with M random-initialized queries $Q = \{q_i\}_{i=1}^{M}$ to capture the high-level information of images. We build the transformer-based network on top of the multi-scale language-aware visual features to extract M image tokens $T = \{(e_i, l_i)\}_{i=1}^{M}$, each of which is represented by an embedding e_i and a location l_i, denoting the semantic and positional information of the token. This design not only represents the images independent of input resolution but also extracts the visual representation that is informative with respect to the language prompts.

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

- Unified Language Instruction
  - Vision-Language Tasks
    - Image Captioning : “The image is <image>. Please generate a caption for the image: ”
    - VQA : “The image is <image>. Please generate an answer for the image according to the question: <question>”.

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

- Unified Language Instruction
  - Vision-Only Tasks
    - propose a unified output format represented as a tuple (C, P), where C denotes the class index in the category set <class>, and $P = \{x_i, y_i\}_{i=1}^{N}$ represents N points that locate the object. To align with the format of word tokens, both the class index C and the coordinates of points x_i, y_i are transformed into discretized tokens. Specifically, the class index is an integer starting from 0, and the continuous coordinates of the points are uniformly discretized into an integer within the range [-<range>, <range>]. For object detection and visual grounding tasks, the point number N is equal to 2, representing the the top-left and bottom-right points of object's bounding box. In the case of instance segmentation, we employ multiple ($N > 8$) points along the object boundary to represent an instance mask [69]. Other perception tasks such as pose estimation (keypoint detection) can also be formulated as language instructions in this way.
  - Prompt example
    - An example of language instruction for the instance segmentation task is as follows: "*Segment all the objects of category set* <class> *within the* <range> *of the image and generate a list of the format (c, x1, y1, x2, y2, ..., x8, y8). Here, c represents the index of the class label starting from 0, and (x1, y1, x2, y2, ..., x8, y8) correspond to the offsets of boundary points of the object relative to the center point. The image is:* <image>".

classification tokens is flexibly provided in the category set <class> of language instructions, such as {"person":<c0>, "car":<c1>, "black cat":<c2>,...}. This design allows our model to select

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

- Experiment Results

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

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

- Experiment Results

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

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

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

- Training Schedule
  - Stage 1
    - Training Deforamable DETR & BERT
    - Only Detection
  - Stage 2
    - Freeze Deformable DETR, Training BERT & LLM
    - Detection & Instance Segmentation & Visual Grounding & VQA & Captioning

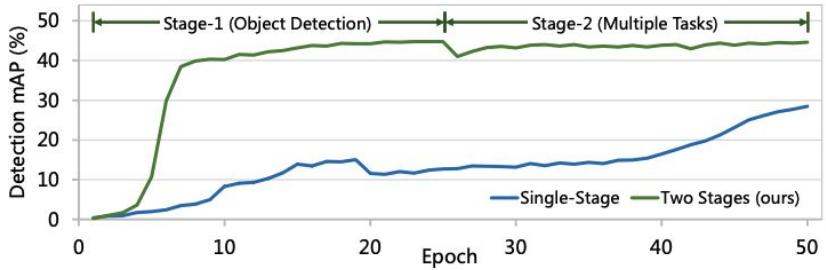


Figure 5: **Comparison of two training schedules for VisionLLM.** We found that a two-stage training from easy to hard converges faster than a single-stage training.

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

- Prompt

A.1 Object Detection

Example 1. "*Please examine the image and identify all objects in the category set* `<class>`*. For each object, specify its location within the range* `<range>` *by determining the top-left and bottom-right corners of its bounding box. To indicate the object's class and location, provide the output in the format (c, x1, y1, x2, y2), where 'c' represents the class index starting from 0, and (x1, y1, x2, y2) correspond to the offsets of the bounding box corners relative to the center point. The image is:* `<image>`"

Example 2. "*Identify all the objects in the image that belong to the category set* `<class>` *and predict a bounding box around each one. The output should be a list in the format (c, x1, y1, x2, y2), where c represents the index of the class label starting from 0, and x1, y1, x2, y2 are the offsets of the top-left and bottom-right corners of the box relative to the center point. The coordinates should be within* `<range>`*. The image is:* `<image>`"

Example 3. "*For each object in the image that is a member of the category set* `<class>`*, output a tuple with the index of class label starting from 0 and the offsets of corners relative to the center point that encloses the object. The offsets should be in the order of top-left and bottom-right corners of the rectangle and should be within* `<range>`*. The output format should be (c, x1, y1, x2, y2). The image is:* `<image>`"

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

- Prompt

A.2 Instance Segmentation

Example 1. *"Segment the objects from the image with class labels from* `<class>` *and output their coordinates within range* `<range>`*. The coordinates should be given as the boundary points relative to the center point, and the output format should be (c, x1, y1, x2, y2, ..., x20, y20), where c is the index of the class label that starts from 0. The image is:* `<image>`*"*

Example 2. *"Segment all the objects from the category set* `<class>` *in the provided image and output a tuple (c, x1, y1, x2, y2, ..., x14, y14) for each, where c is the index of the class label in the category set that starts from 0, and (x1, y1, x2, y2, ..., x14, y14) correspond to the offsets of boundary points on the instance mask relative to the center point which should be within* `<range>`*. The image is:* `<image>`*"*

Example 3. *"In the provided image, please segment all the objects in category set* `<class>` *within the range* `<range>` *by providing their coordinates in the (c, x1, y1, x2, y2, ..., x24, y24) format, where 'c' denotes the index of the class label starting from 0, and (x1, y1, x2, y2, ..., x24, y24) stand for the offsets of boundary points relative to the center point. The image is:* `<image>`*"*

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

- Prompt

A.3 Visual Grounding

Example 1. "*Please find the object in the category set* {<expression>:<cls0>} *within the range* <range>. *Please provide the output in the format (c, x1, y1, x2, y2), where c is the class index starting from 0, and (x1, y1, x2, y2) are the offsets of the top-left and bottom-right corners of the bounding box relative to the center point. The image is:* <image>"

Example 2. "*Given the input image, category set* {<expression>:<cls0>}, *and the range* <range>, *please locate the object in the image and output the corresponding coordinates in the tuple (c, x1, y1, x2, y2), where c is the index of the class label starting from 0, and (x1, y1, x2, y2) are the offsets of the top-left and bottom-right corners of the rectangle relative to the center point. The image is:* <image>"

Example 3. "*For each object in the image that belongs to the* {<expression>:<cls0>} *category set, please provide the class label (starting from 0) and the offsets from the center of a bounding box that encloses the object. The corner offsets should be in the order of top-left and bottom-right, and within the range* <range>. *The output should be in the format (c, x1, y1, x2, y2). The image is:* <image>"

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

- Prompt

A.4 Image Captioning

Example 1. "*The image is* <image>. *Write a caption:* "

Example 2. "*The image is* <image>. *Please describe this image:* "

Example 3. "*With the objects in the* <image>, *please generate a caption for the image:* "

A.5 Visual Question Answering

Example 1. "*The image is* <image>. *Please generate an answer according to the question:* <question>. "

Example 2. "*The image is* <image>. *Please answer the question* <question> *according to the image.* "

Example 3. "*With the objects in the* <image>, <question>. "

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

- Figure

Instruction: "*Identify the objects in the image that belong to {'person': <c0>, ..., 'frisbee': <c29>, ...} 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' represents the index of the class label starting from 0, and x1, y1, x2, y2 are the offsets of the top-left and bottom-right corners of the box relative to the center point. The coordinates should be within* <range>. *The image is:* <image>"

Instruction : "*Identify the objects in the image that belong to {'frisbee': <c0>} 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' represents the index of the class label starting from 0, and x1, y1, x2, y2 are the offsets of the top-left and bottom-right corners of the box relative to the center point. The coordinates should be within* <range>. *The image is:* <image>"

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

- Figure

Instruction: "*Identify the objects in the image that belong to {'person': <c0>, 'bicycle': <c1>, ..., 'backpack': <c24>, ..., 'toothbrush': <c79>} 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' represents the index of the class label starting from 0, and x1, y1, x2, y2 are the offsets of the top-left and bottom-right corners of the box relative to the center point. The coordinates should be within* <range>. *The image is:* <image>"

Instruction: "*Identify the objects in the image that belong to {'bicycle': <c0>, 'surfboard': <c1>, 'sports ball': <c2>, 'backpack': <c3>, 'the man wearing blue T-shirt': <c4>} 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' represents the index of the class label starting from 0, and x1, y1, x2, y2 are the offsets of the top-left and bottom-right corners of the box relative to the center point. The coordinates should be within* <range>. *The image is:* <image>"

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

- Figure

Instruction: *"The image is:* <image>. *Analyze the image in a comprehensive and detailed manner."*

VisionLLM: *"The image features a busy city street with a large red double-decker bus driving down the road. There are several people walking on the sidewalk, some of whom are carrying backpacks. The bus is positioned towards the left side of the image. There are several people walking along the sidewalk, with some closer to the bus and others further away. The overall atmosphere suggests a busy urban environment with people going about their daily routines."*

Instruction: *"The image is:* <image>. *Write a detailed description of the given image."*

VisionLLM: *"The image features a large, open-air market filled with a variety of fruits and vegetables. There are numerous mangoes, watermelons and apples displayed in the market, with some fruits placed closer to the foreground and others further back. The market is bustling with activity as people can be seen browsing and purchasing the fresh produce. The market is likely a popular destination for those seeking a fresh and healthy diet."*

Instruction: *"The image is:* <image>. *Write a short description of the given image."*

VisionLLM: *"The image features a lively open-air market with a variety of fruits and vegetables. People are busy browsing and buying fresh produce."*

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

- Figure

Instruction: "*The image is:* <image>. *Please generate an answer according to the question: what reasons could there be for the cat squatting on the shoes?*"

VisionLLM: "*There could be several reasons. One possibility is that the cat is seeking warmth or comfort, as shoes can provide a cozy and warm surface for the cat to rest on. Another reason could be that the cat is seeking attention or affection from its owner, who is standing nearby. Cats are known to be curious creatures, and they may be seeking attention or attention from their owners.*"

Instruction: "*The image is:* <image>. *What do you think of the working environment here?*"

VisionLLM: "*The working environment in this office space appears to be open, spacious, and well-equipped. There are several desks set up for employees, providing ample workspace for each individual. The office features a modern and contemporary design, which suggests a creative and collaborative work environment. The presence of several chairs and the organization of the office, including the laptops and monitors, indicate that the space is well-maintained and conducive to productivity. Overall, the office appears to be a pleasant and efficient work environment for its employees.*"

(b) Visual question answering with reasoning

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Introduction

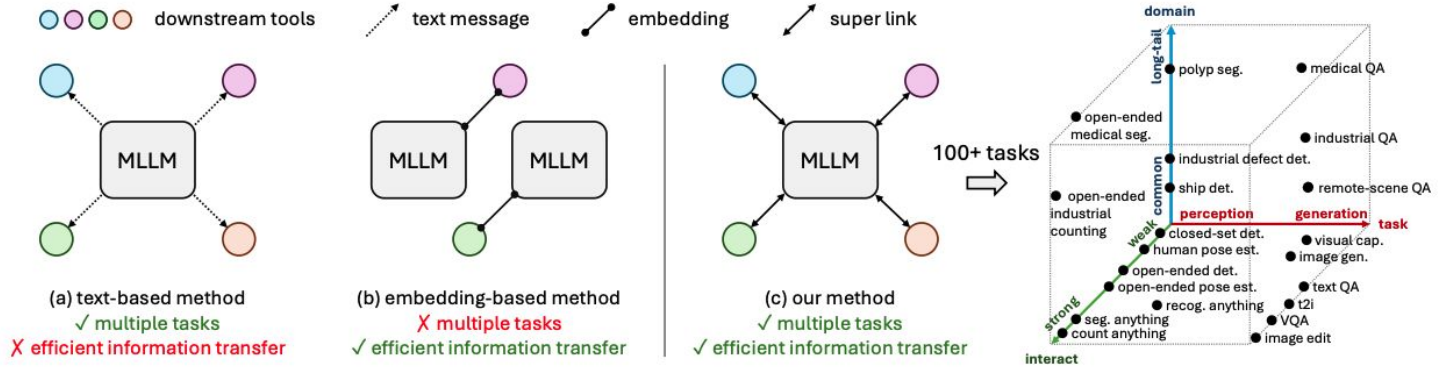
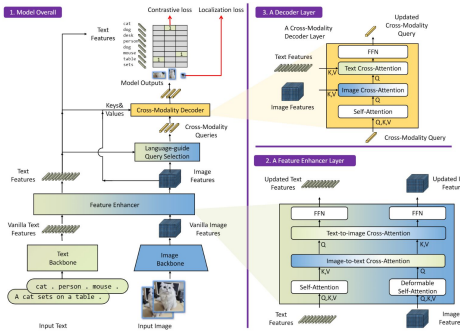
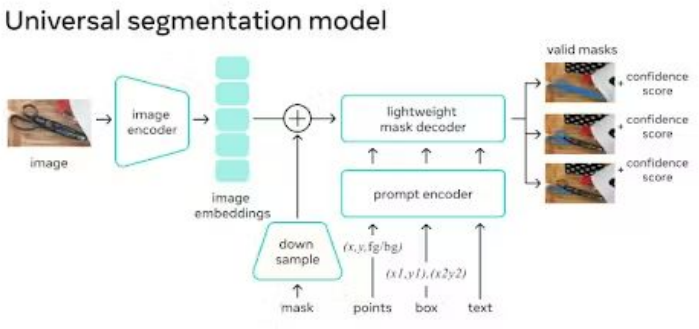


Figure 1: **Illustration of three information transmission methods.** (a) Text-based method shows MLLM connected to various downstream tools via text messages, capable of handling multiple tasks but suffering from inefficient information transfer. (b) The embedding-based method displays a connection using learnable embeddings, which facilitates efficient information transfer but lacks support for multitasking. (c) Our method employs a "super link" technique, where a unified MLLM interfaces with multiple task decoders through super links, supporting over 100 diverse tasks.

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Introduction
  - Vision Generalist Model
    - Unified Vision Model
      - Pix2Seq-D, SEEM, Semantic-SAM, Grounding DINO…
    - Visual Prompting
      - Segment Anything
    - Diffusion Model as Interface
      - InstructCV, InstructDiffusion





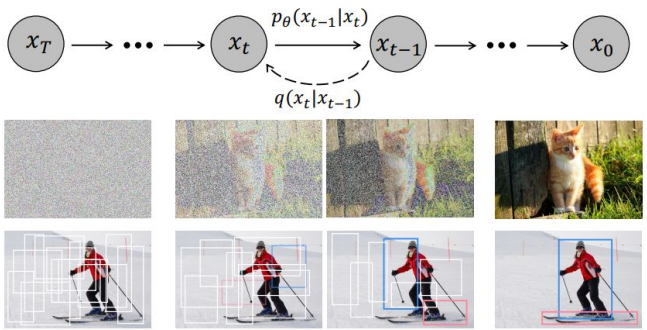


Figure 3. The framework of Grounding DINO. We present the overall framework, a feature enhancer layer, and a decoder layer in block 1, block 2, and block 3, respectively.

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Model Design - (image encoder, LLM, task-specific decoders, super link)
  - Tokenization
    - Text prompt -> used in LLM
    - Image input
      - (3, 336, 336) -> Vision Transformer(CLIP-L) -> (576 + 1, C)
      - 576 = (336 / 14)^2, 1 = cls token
    - Visual prompt
      - point, box, scribble, mask -> binary mask
      - concatenate with input image along the channel dimension
      - process three convolution layers to downsample by factor of 14
      - adding to image input feature
      - grid sampling is used to extract features within the masked regions
      - averaged to form the feature of the visual prompt (1, C)

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Model Design - (image encoder, LLM, task-specific decoders, super link)
  - LLM
    - Just use Vicuna-7B

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

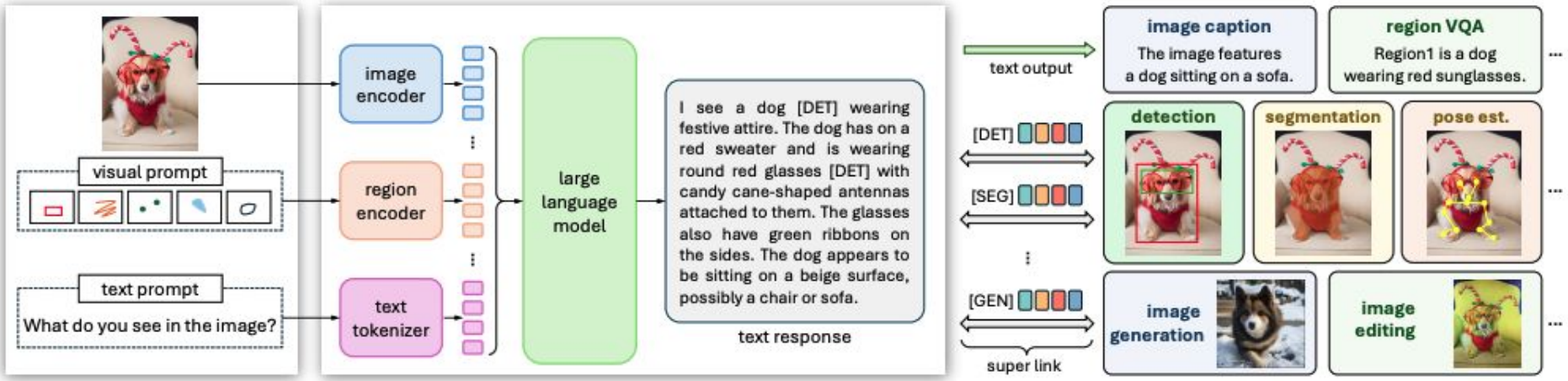
- Model Design - (image encoder, LLM, task-specific decoders, super link)
  - Task-specific decoders
    - Use Grounding DINO for object-level localization
    - add a mask decoder for segmentation
    - UniPos for pose estimation(key point decoder)
    - Stable Diffusion, Instructpix2pix to generate or edit images

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Model Design - (image encoder, LLM, task-specific decoders, super link)
  - Super Link Technique
    - Text-only output tasks(VQA, …) -> just vision language model
    - Vision Specific Task
      - Routing Token : [DET], [POSE], [SEG], [GEN], [EDIT]
      - Super-Link Queries
        - Q -> R(N X C), N is the number of queries, randomly initialized
        - super link queries are appended
          after the input embeddings of the routing token
        - appended token is sent into the specific decoders as s condition
          to perform the downstream tasks

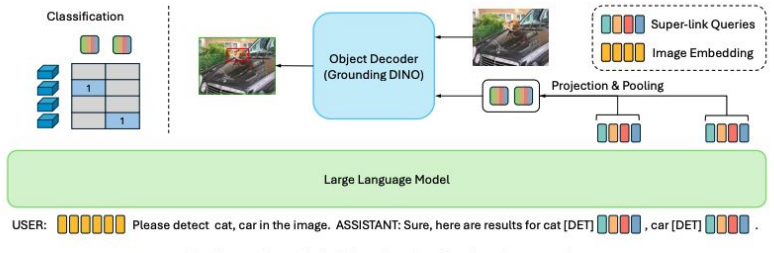
VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Model Design - (image encoder, LLM, task-specific decoders, super link)

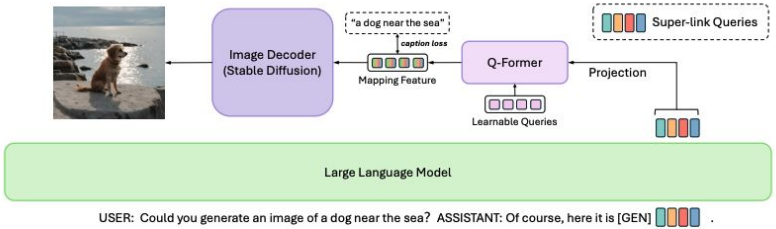


VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Model Design - (image encoder, LLM, task-specific decoders, super link)



(a) Connecting with object decoder for visual perception.



(b) Connecting with image decoder for visual generation.

Figure A13: **Architecture details for connecting LLM with task-specific decoder via super-link queries.** (a) Connecting with object decoder. We first extract the per-category features by performing projection and pooling on the hidden states of corresponding super-link queries. Then these features are sent into the object decoder as text features. (b) Connecting with image decoder. We add a Q-Former for projecting the features of super-link queries to the feature space of Stable Diffusion.

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Model Design - (image encoder, LLM, task-specific decoders, super link)

Example1: Text Prompt + Visual Prompt for Interactive Segmentation.
USER: <image> Could you please segment all the corresponding objects according to the visual prompts as region1 <region>, region2 <region>?
ASSISTANT: Sure, these objects are region1 [SEG], region2 [SEG].

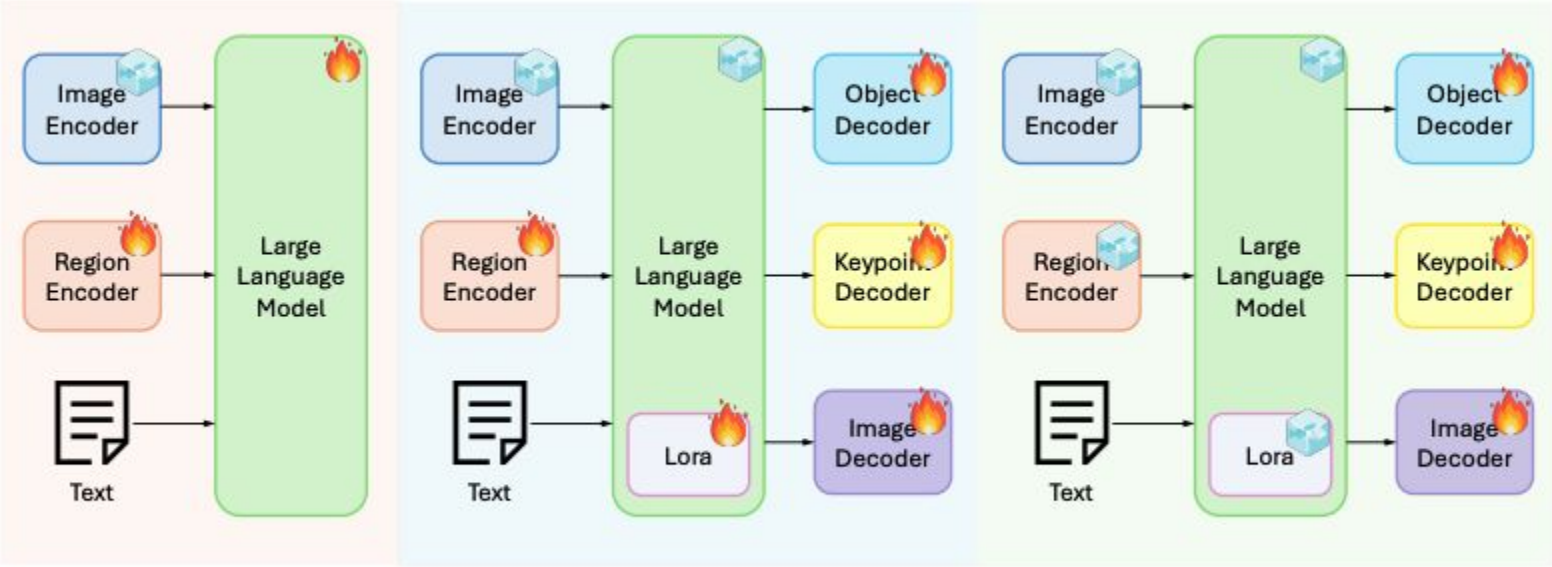
Example 2: Text Prompt for Text-to-Image Generation.
ASSISTANT: Of course, here it is [GEN].

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Training Stage
  - Stage - 1 : Multi modal Training (VisionLLM v2-Chat)
    - LLaVA pre-training and instruction tuning
    - training projection
  - Stage - 2 : Multi-capacity Fine-Tuining
    - integrate task-specific decoders into the model
      and perform multi-task joint training
    - instruction data, COCO(object detection), ADE20K(semantic segmentation)
  - Stage - 3 : Decoder-only Fine Tuning
    - Training only all decoders

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

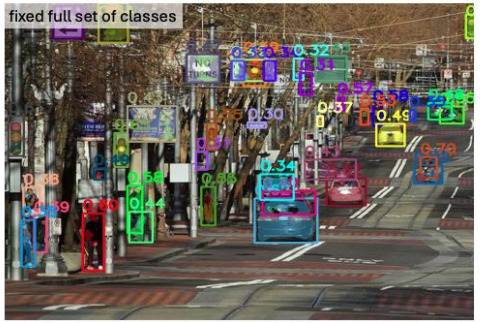
- Training Stage



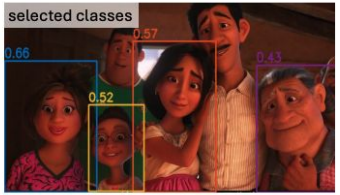
VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Qualitative Results
  - Visual Perception
    - Closed set

Please conduct object detection to any [List of COCO classes] that may be present.

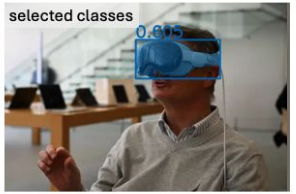


Please conduct object detection to any [List of COCO classes] that may be present.



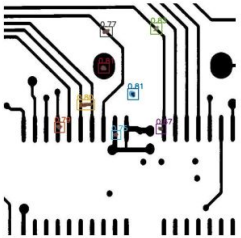
Can you carry out object detection on this image and identify the women it contains?

I'm trying to detect bottles and forks in the image. Can you help me?

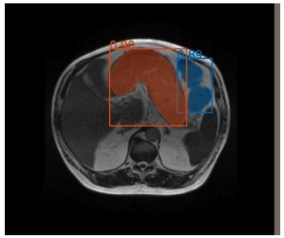


Are you capable of identifying Apple Vision Pro within this image?

(a) Remote Sensing



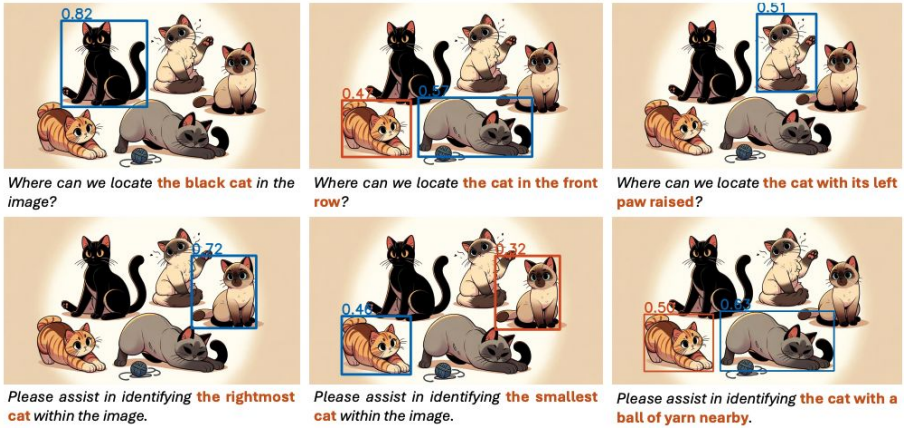
(b) PCB

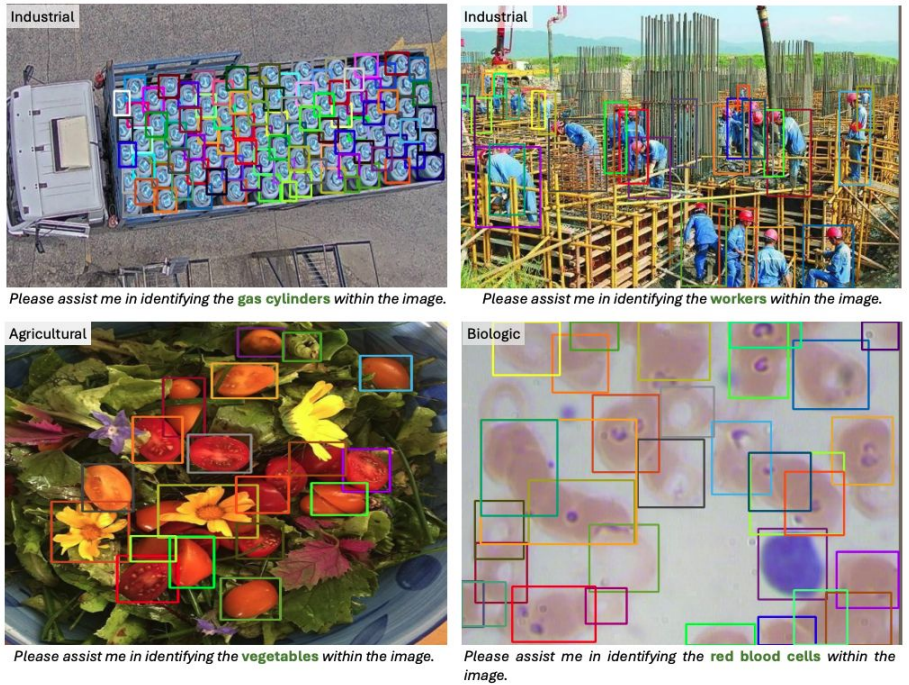


(c) Medical

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

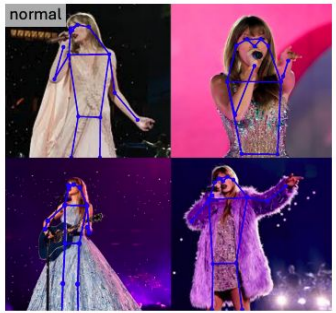
- Qualitative Results
  - Visual Perception
    - Open set





VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Qualitative Results
  - Visual Perception
    - Keypoint



*I need your expertise to locate any **person** in this image. Can you pinpoint the keypoint locations of **[List of 17 COCO keypoints]** ?*

*I need your expertise to locate any **person** in this image. Please analyze this image and find the keypoint of **the right elbow**?*

*Detect any **person** in the given image. Can you pinpoint the keypoint locations of the **nose, left_eye, right_eye, left_ear, right_ear** ?*

*Please perform object detection on this image for identifying **elephants**. Detect the keypoint positions of the **List of 17 AP-10K keypoints**].*

*Can you detect **horses** within the image? Can you pinpoint the keypoint locations of **[List of 17 AP-10K keypoints]** ?*

VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks

- Qualitative Results
  - Visual Perception
    - Grounded caption

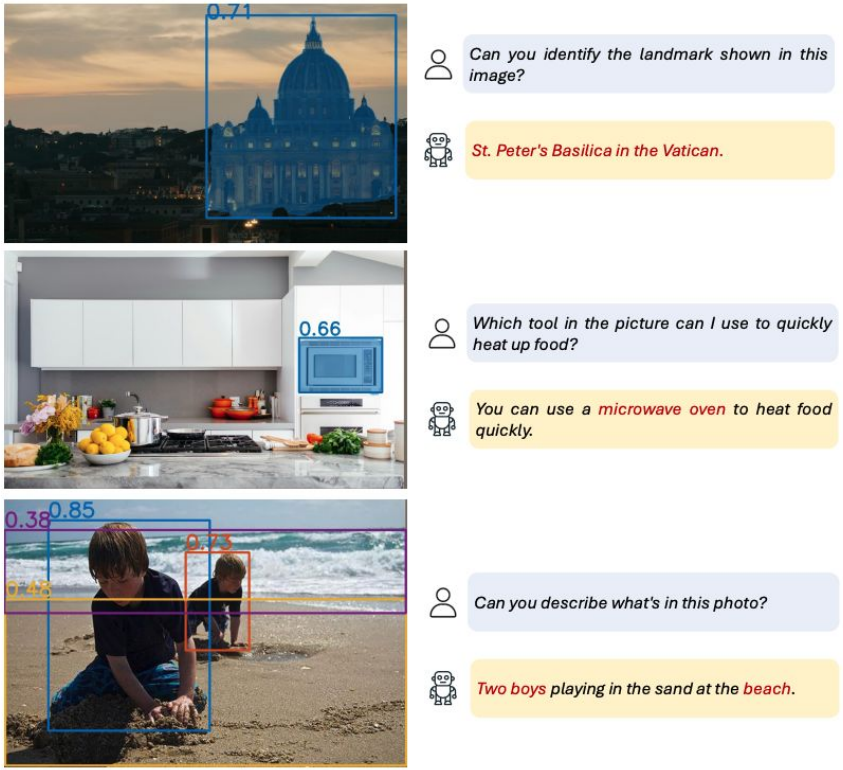


Figure A6: **Grounded caption.** The model accurately locates objects based on user prompts, outputs bounding boxes, and provides answers to user queries.