上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

LLM-based Frameworks for Vision-centric Tasks
基于大语言模型的视觉中心任务

团队：视觉传感器

时间：2024.3.1

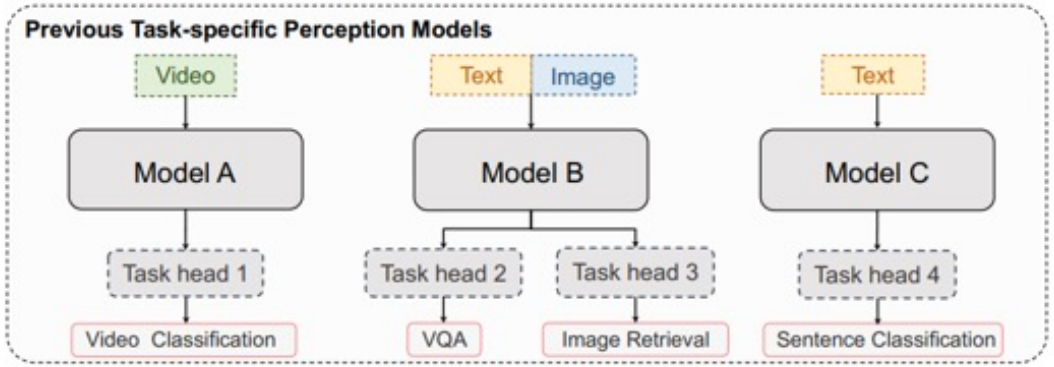
汇报人：孙桥

目录

背景

Vision-centric Tasks:

To enable the computer to understand what it sees (and react accordingly).

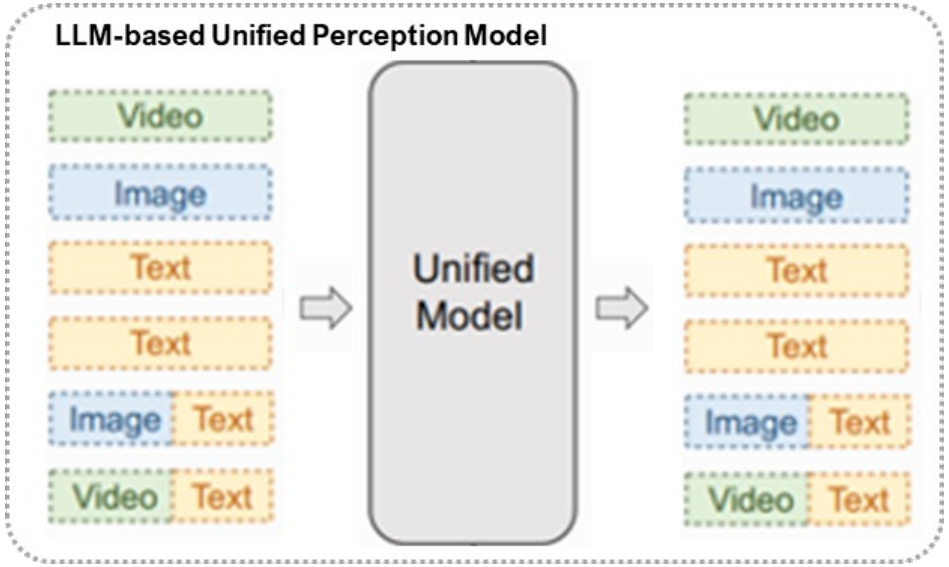


In the past:

- on a task-by-task basis
- specific Vision Foundation Models (VFMs)
- various different networks and loss functions
- modalities can be handled are simple

NLP: LLMs as a unified solution

LLMs -> Unified Vision Model ?

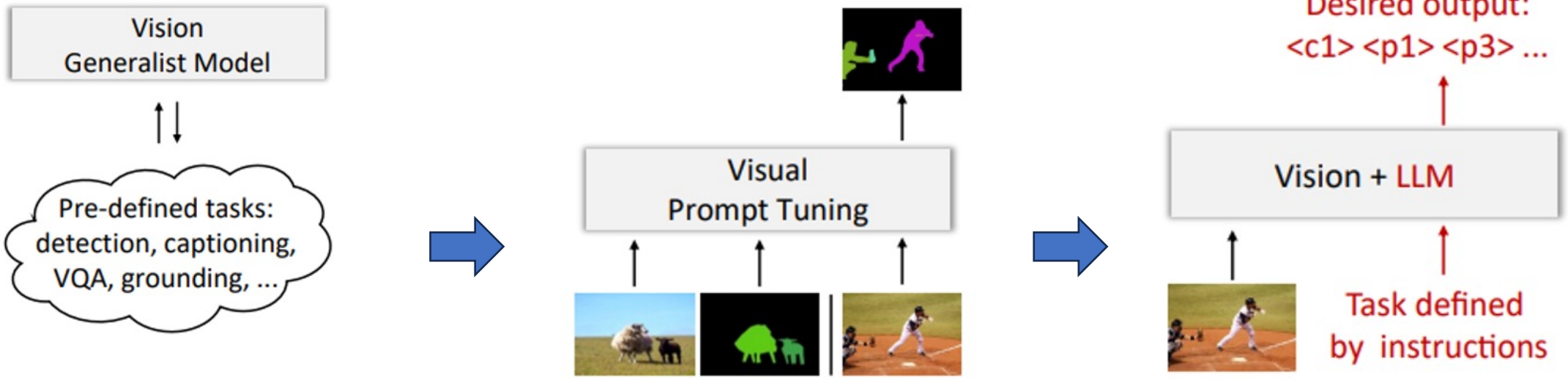


Obstacle :

Diversity and labelling gaps in CV tasks..

方法

Paradigm Evolution：



Vision Generalist Models：
Multiple tasks, shared architecture

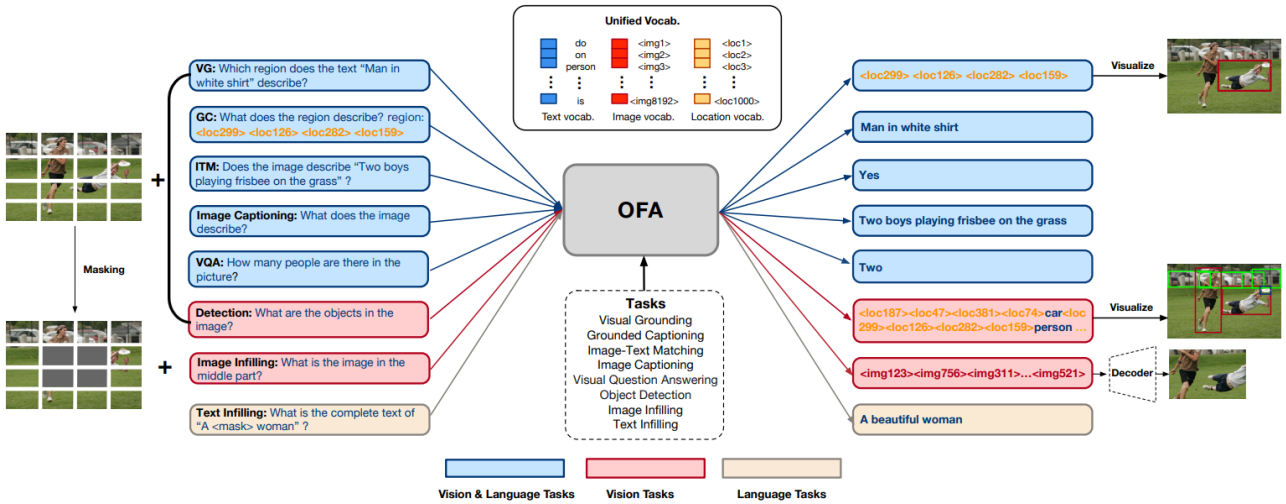
Visual Prompt Tuning：
Inconsistent format

Vision Generalist Models：
Flexibility & consistency

Vision Generalist Models： Seq2seq manner

Insight: the success seq2seq models in the field of NLP

Modeling diverse tasks as sequence generation tasks: OFA, Flamingo, and GIT.



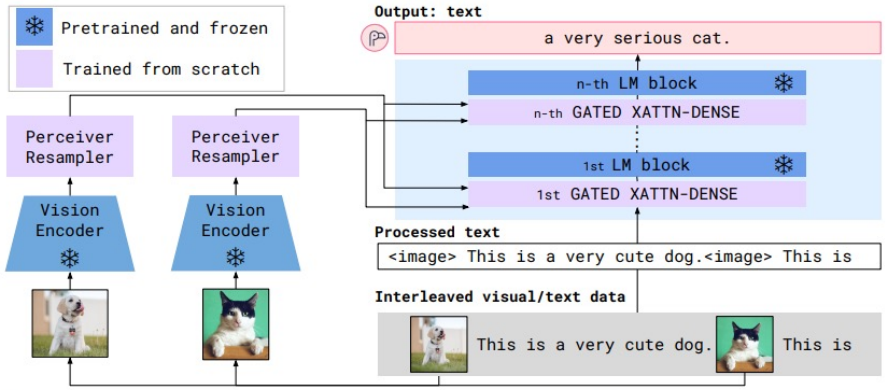


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

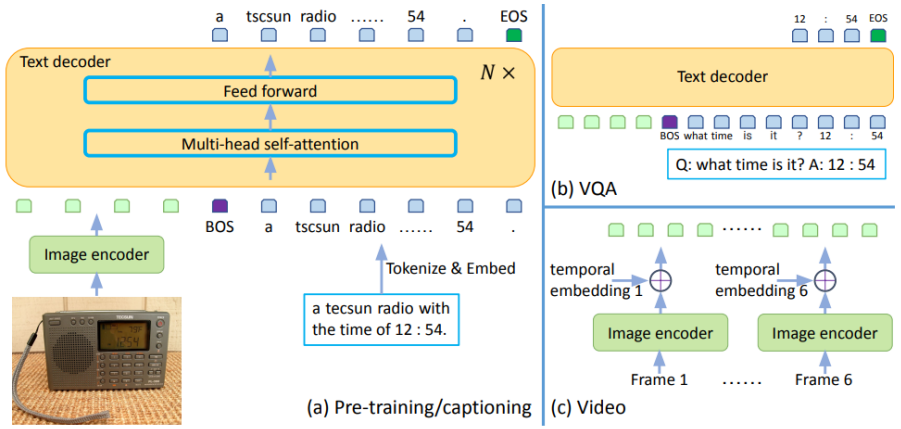
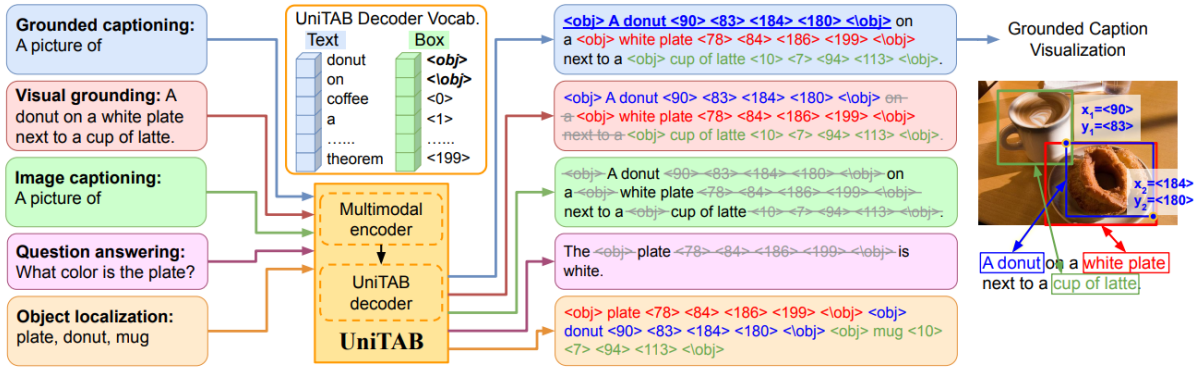


Figure 2: Network architecture of our GIT, composed of one image encoder and one text decoder. (a): The training task in both pre-training and captioning is the language modeling task to predict the associated description. (b): In VQA, the question is placed as the text prefix. (c): For video, multiple frames are sampled and encoded independently. The features are added with an extra learnable temporal embedding (initialized as 0) before concatenation.

Vision Generalist Models： Seq2seq manner, *with discrete coordinate tokens*

Extends this idea by using discrete coordinate tokens to encode and decode spatial information for more tasks: Unified-IO, Pix2Seq v2, and UniTab.



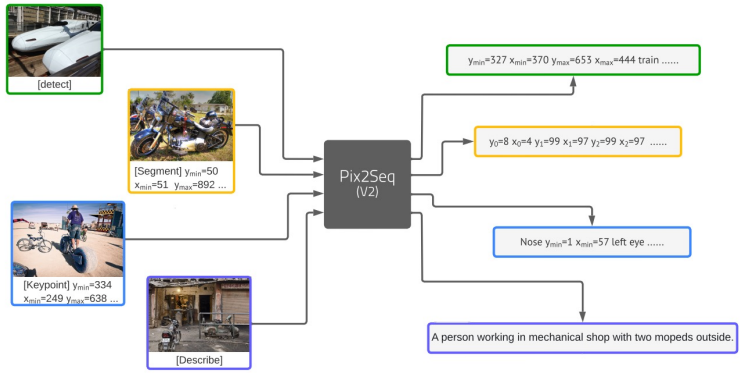


Figure 1: An illustration of the proposed framework. An image and a sequence of task prompt is given, the model produce a sequence of discrete tokens corresponding to the desired output.

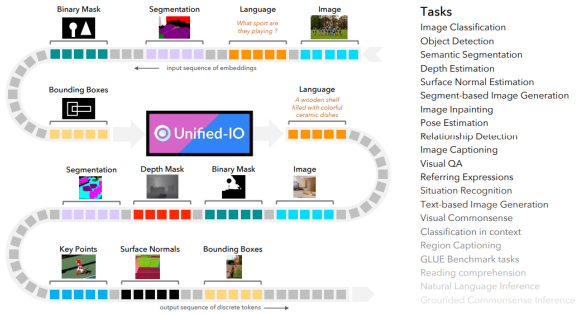


Figure 1: UNIFIED-IO is a single sequence-to-sequence model that performs a variety of tasks in computer vision and NLP using a unified architecture without a need for either task or modality-specific branches. This broad unification is achieved by homogenizing every task's input and output into a sequence of discrete vocabulary tokens. UNIFIED-IO supports modalities as diverse as images, masks, keypoints, boxes, and text, and tasks as varied as depth estimation, inpainting, semantic segmentation, captioning, and reading comprehension.

suffer from:

- slow inference speed
- performance degradation

due to:

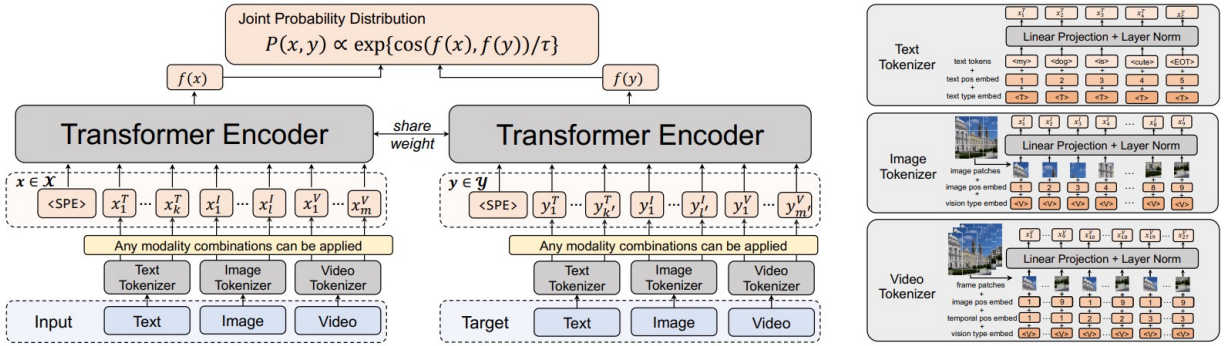
- the non-parallel auto-regressive decoding process.

Vision Generalist Models： Maximum likelihood target manner

Solve these issues by unifying different tasks using the maximum likelihood target for each input based on representation similarity, regardless of their modality.

Uni-Perceiver: formulates various perception tasks as finding the maximum likelihood target for each input through the similarity of their representations.

Making it possible to support both generation and non-generation tasks in a unified framework.



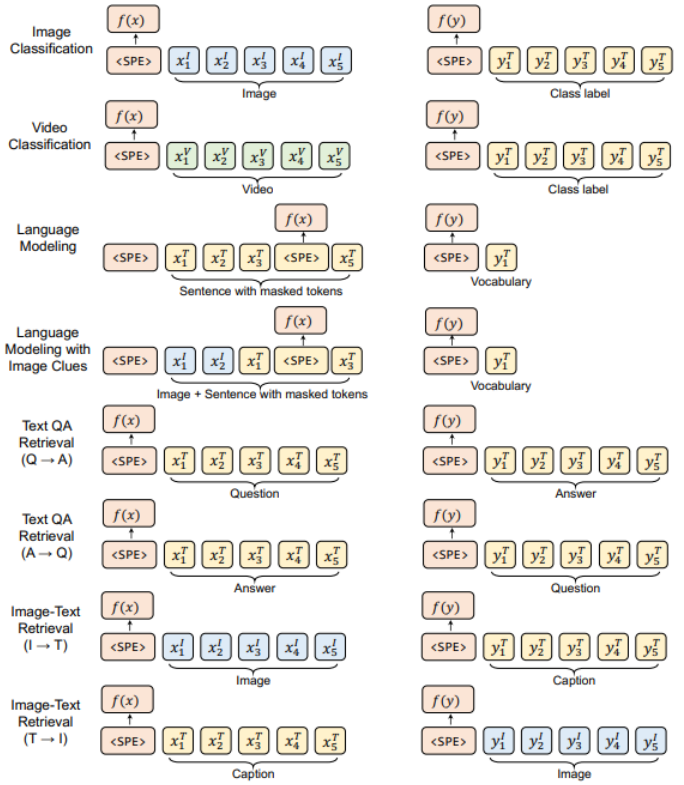


Figure 3. Input and target formats of pre-training tasks. For each task, the left column represents the format of input sequence x, and the right column represents the format of the target sequence y. $f(x)$ and $f(y)$ indicate the representations used for calculating the joint probability distribution as in Eq. (2). Here, we have omitted the tokenizer and encoder for concision.

Vision Generalist Models： Maximum likelihood target manner

Uni-perceiver v2: enhancing in versatility & performance

by adopting:

- an improved optimizer
- an unmixed sampling strategy

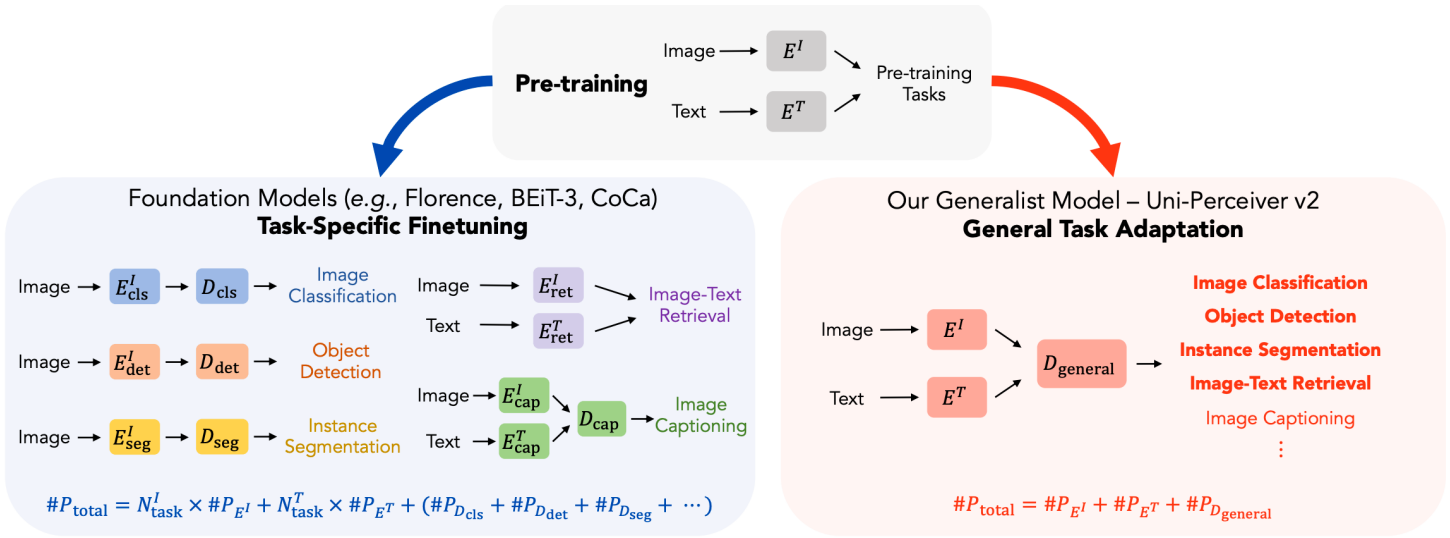


Figure 1. Comparison of foundation models and Uni-Perceiver v2. E^I and E^T denote the image encoder and text encoder, respectively. In existing foundation models, task-specific decoders $D_{\text{cls}}, D_{\text{det}}, \ldots$ are employed to tune E^I and E^T in different task-specific finetuning. The total number of parameters $\#P_{\text{total}}$ in adaptation grow with the number of visual/linguistic tasks, denoted as N^I_{task} and N^T_{task}, respectively. By contrast, our Uni-Perceiver v2 shares all parameters across various downstream tasks with a general decoder D_{general}, where no task-specific fine-tuning is incorporated. Better than previous generalist models, our method can also effectively handle pillar tasks such as image classification, object detection, instance segmentation, and image-text retrieval.

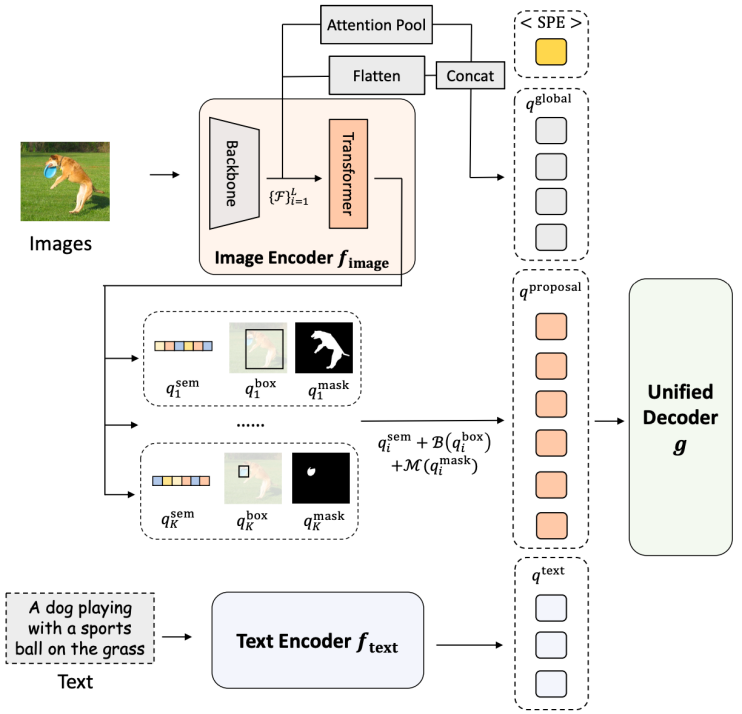


Figure 3. Architecture overview of our Uni-Perceiver v2.

Vision Generalist Models： Maximum likelihood target manner

Uni-Perceiver-MoE: handle task-interference issue — different tasks with shared parameters may conflict with each other

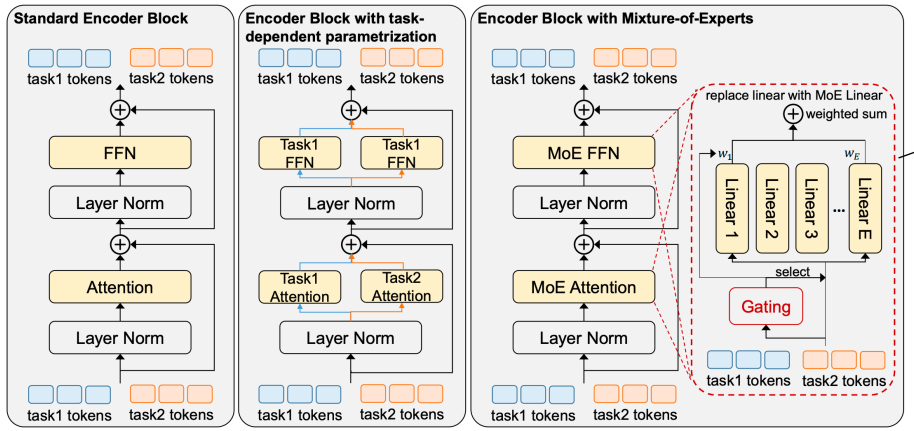


Figure 1: Comparisons of fully-shared standard encoder block, task-specific encoder block with task-dedicated parameters, and encoder block with efficient MoE parameterization.

Given any token x_i in the input sequence $X = \{x_i\}_{i=1}^{L}$,
a gate decision vector $\mathcal{G} \in \mathbb{R}^E$ that dispatches different input tokens to different experts,

$$\mathcal{G} = \text{top}_k\left(\text{softmax}\left(\mathbf{W}_g \cdot R(x_i) + \epsilon\right)\right), \quad k \ll E \text{ non-zero}$$

where $R(\cdot)$ defines a general routing strategy for gate decision
$\mathbf{W}_g$ is the trainable weights in gate decision and ϵ is the noise term.
output y_i is the weighted combination of each expert's computation on x_i as:

$$y_i = \sum_{e=1}^{E} \mathcal{G}_e \cdot \mathbf{W}_e \cdot x_i,$$

where $\mathbf{W}_e$ is the linear projection weights of the e-th expert and gate decision $\mathcal{G}_e$ determines how much the e-th expert contributes to the output y_i.

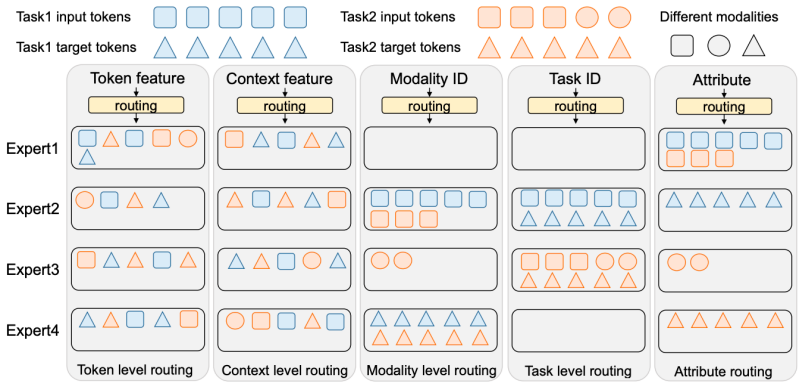


Figure 2: Comparisons of routing strategies with the top-1 gate decisions under 2-task training.

- **data-dependent variants:**

$$R_{\text{token}}(x_i) = x_i.$$

$$R_{\text{context}}(x_i) = \text{concat}(x_i, \text{attnpool}(X)),$$

  - Bad excellent memory efficiency...
  - model parallelism is required.

- **data-independent variants:**

$$R_{\text{modal}}(x_i) = \text{embed}(\text{id}_{\text{modal}}(x_i))$$

$$R_{\text{task}}(x_i) = \text{embed}(\text{id}_{\text{task}}(x_i)),$$

$$R_{\text{attr}}(x_i) = \text{layernorm}\left(\mathbf{W}_{\text{attr}} \cdot \text{attr}(x_i)\right).$$

- Still restricted by pre-defined tasks
- Cannot support flexible open-ended task customization based on language instructions

Excellent memory efficiency: only top-k experts need to be activated for all tokens with the same modality/task/attributes.
-> Can be merged in training and inference.

Vision Generalist Models： Text-to-image Manner utilizing Diffusion Models

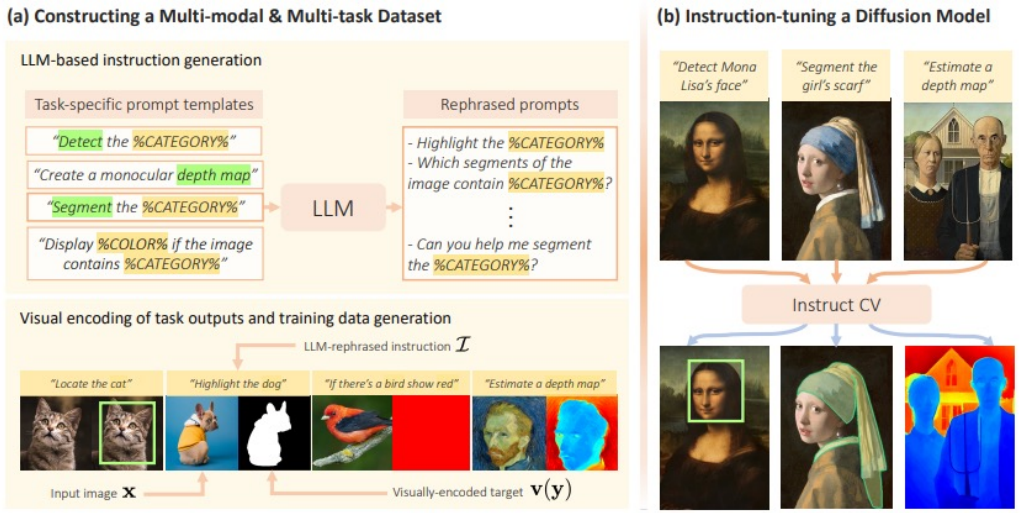


Figure 2: **Pictorial depiction of the InstructCV training pipeline.** (a) We pool multiple computer vision datasets to construct a multi-modal and multi-task set of image pairs, where the target of each task is visually encoded in the form of an output image. Starting with a set of task-specific prompt templates, we sample a new instruction for each training point by using an LLM to rephrase the template for the corresponding task. (b) Using the dataset in (a), we finetune a diffusion model to produce the output $\mathbf{v}(\mathbf{y})$ given an image $\mathbf{x}$ & an instruction $\mathcal{I}$.

instruction $\mathcal{I}$ on the input image $\mathbf{x}$

encoded task output $\mathbf{v}(\mathbf{y})$

instruction-tuning dataset $\mathcal{D}_{\mathcal{I}} = \{(\mathbf{x}_i, \mathbf{v}(\mathbf{y}_i), \mathcal{I}_i)\}_i$

finetuning the text-to-image diffusion model using $\mathcal{D}_{\mathcal{I}}$

(1) Semantic Segmentation. The target output $\mathbf{y}$ of this task is typically an assignment of a label or category to every pixel in an image. A natural choice of $\mathbf{v}(\mathbf{y})$ for the semantic segmentation task is a binary mask that labels pixels in the input image $\mathbf{x}$ belonging to the prompted category in $\mathcal{I}$.

(2) Object Detection. Here, the goal is to identify the spatial position of a category in an image using a bounding box, i.e., the label comprises bounding box coordinates $\mathbf{y} = [c_x, c_y, w, h]$. We define $\mathbf{v}(\mathbf{y})$ for object detection as the image $\mathbf{x}$ with a bounding box overlaid according to the coordinates in $\mathbf{y}$.

(3) Monocular Depth Estimation. The target $\mathbf{y}$ of this task is the depth value (i.e., distance relative to the camera) of each pixel in the RGB image $\mathbf{x}$. For this task, we define the visually-encoded target $\mathbf{v}(\mathbf{y})$ as an RGB image in which pixel colors encode the depth values. This encoding is done by converting depth values ranging from 0 to 10 meters (based on depth ranges in the NYUv2 dataset [32]) into the discrete space $[0, 1, \ldots, 255]$ for RGB image representation, i.e., $\mathbf{v}(\mathbf{y}) = \lfloor \mathbf{y} \times \frac{255}{10} \rfloor$. We then apply the same value across all three RGB channels to create a visual depth map.

(4) Image Classification. In multi-class image classification, the target label $\mathbf{y}$ is a categorical value indicating the object depicted in the image $\mathbf{x}$. To represent image classification in a Pix2Pix format, we resort to a color-coding methodology. To this end, we use a prompt template of the form: *"Display %color% if the image contains %category%"*. We sample random colors when filling in the template for individual training points. This steers the text-to-image model to produce an image consisting of the pure color block if the category specified in the prompt is visible in $\mathbf{x}$. (Note that this approach only enables us to predict if a specific category is in the input image $\mathbf{x}$. For multi-class classification, we need to use a series of prompts specifying all categories of interest one at a time.)

We use our instruction-tuning dataset $\mathcal{D}_{\mathcal{I}} = \{(\mathbf{x}_i, \mathbf{v}(\mathbf{y}_i), \mathcal{I}_i)\}_i$ to train a (conditional) diffusion model that conducts the vision task specified in the instruction $\mathcal{I}$ on the input image $\mathbf{x}$, producing a visually-encoded task output $\mathbf{v}(\mathbf{y})$. By finetuning the text-to-image diffusion model using $\mathcal{D}_{\mathcal{I}}$, we steer its functionality from a generative model to a language-guided multi-task vision learner.

Visual Prompt Tuning: Vision and language inputs as prompts

Insights:
Language instructions' power to unify various NLP tasks.

Vision and language inputs as prompts:
Flamingo, BLIP-2, MiniGPT-4, and LLaVA

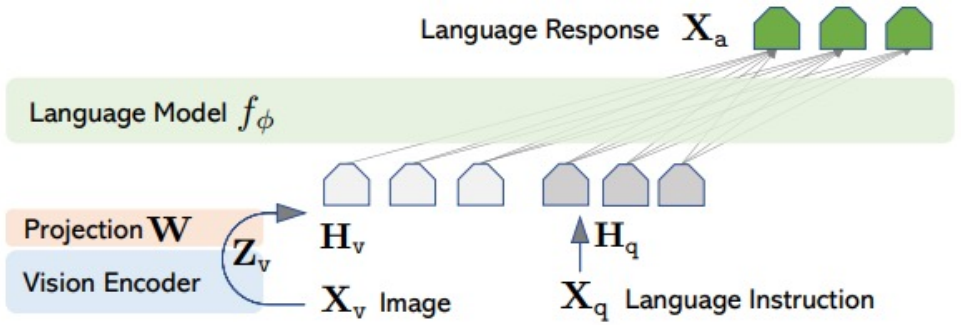


Figure 1: LLaVA network architecture.

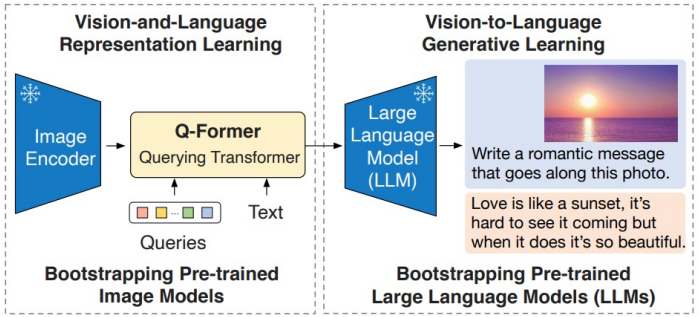


Figure 1. Overview of BLIP-2's framework. We pre-train a lightweight Querying Transformer following a two-stage strategy to bridge the modality gap. The first stage bootstraps vision-language representation learning from a frozen image encoder. The second stage bootstraps vision-to-language generative learning from a frozen LLM, which enables zero-shot instructed image-to-text generation (see Figure 4 for more examples).

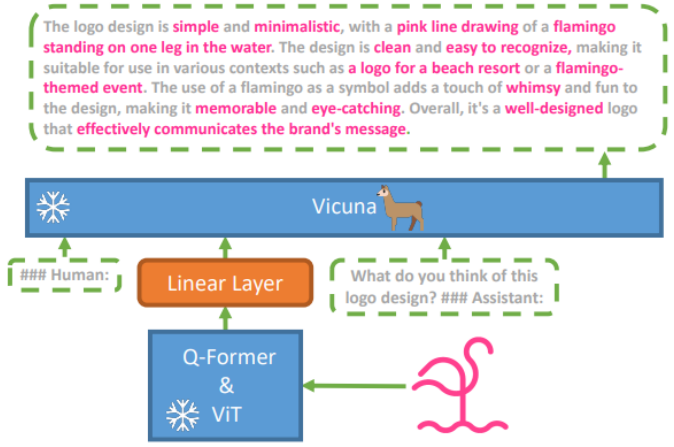


Figure 1: **The architecture of MiniGPT-4.** It consists of a vision encoder with a pretrained ViT and Q-Former, a single linear projection layer, and an advanced Vicuna large language model. MiniGPT-4 only requires training the linear projection layer to align the visual features with the Vicuna.

- **focus on:** image-to-text tasks
- **fail to:** address visual perception

Visual Prompt Tuning: Visual prompting frameworks for perception

As image inpainting tasks: Visual prompting via Image Inpainting, Painter, and SegGPT.

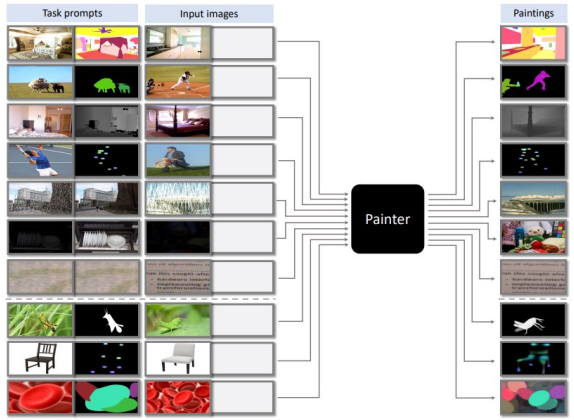


Figure 1. An illustration of the in-context inference of Painter. Painter is a generalist vision model, which can automatically perform vision tasks according to the input task prompts without the task specific heads. Painter can not only perform in-domain tasks with highly competitive performance, such as semantic segmentation (Row 1), instance segmentation (Row 2), depth estimation (Row 3), keypoint detection (Row 4), denoising (Row 5), deraining (Row 6), and image enhancement (Row7), but also be able to rapidly adapt to various out-of-domain vision tasks using simple prompts, such as open-category object segmentation, keypoint detection, and instance segmentation (Row 8-10).

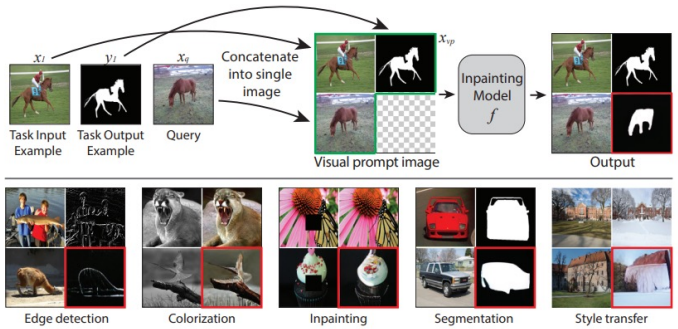


Figure 1: **Visual prompting via Image Inpainting.** *Top*: Prompting Image Inpainting Models. Given input-output example(s) (x_1, y_1) and image query x_q, we construct a grid-like single image called a *visual prompt* x_{vp}. The visual prompt is composed of the desired task example(s) and a new query image (all in green). The inpainting model goal is then to predict the masked region (red) such that it is consistent with the example(s). *Bottom*: an inpainting model can solve this way various computer vision tasks, given that it was trained on the right data. The model predictions are annotated in red.

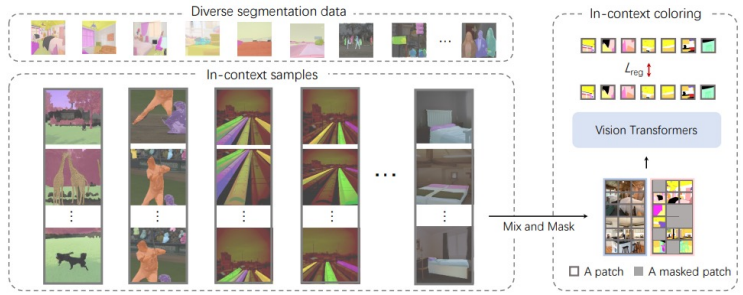


Figure 2: Illustration of overall training framework of SegGPT. We incorporate diverse segmentation data, including part, semantic, instance, panoptic, person, medical image, and aerial image segmentation, and transform them into the same format of images. We generate in-context samples that share similar contexts on-the-fly, *e.g.*, the overlapped colors shown in each column, which indicate the same category or the same instance. We adopt a general Painter [46] framework with in-context coloring as the training objective and a random coloring scheme for more flexible and generalizable training.

Strength:

- Good at segmentation tasks

Limitations:

- Image inpainting is inconsistent with the language instructions in LLMs
- Hard to define visual prompts to numerous real-world vision tasks
- Hard to leverage the reasoning, parsing ability, and world knowledge of LLMs

LLM-based Frameworks: LLM-centric unified vision, with image as a foreign language

Insight: use language instructions to softly entail all tasks and solve them with a shared LLM-based task decoder.

Better solution for integrating the strengths of LLMs with the various vision-centric tasks.

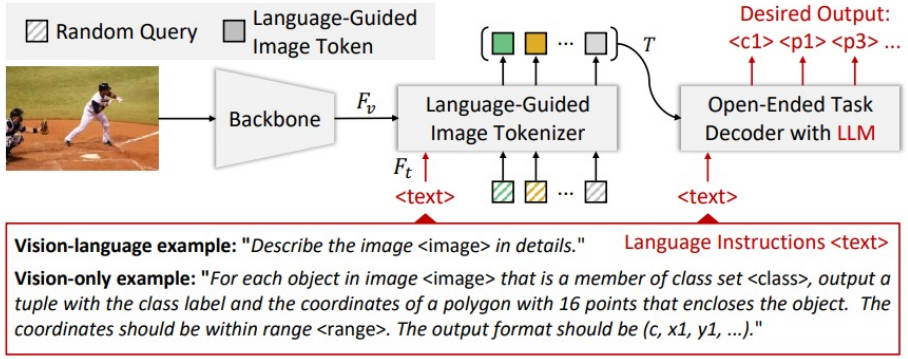


Figure 3: **Overall architecture of the proposed VisionLLM.** It consists of three parts: a unified language instruction designed to accommodate both vision and vision-language tasks, an image tokenizer that encodes visual information guided by language instructions, and an LLM-based open-ended task decoder that executes diverse tasks defined by language instructions.

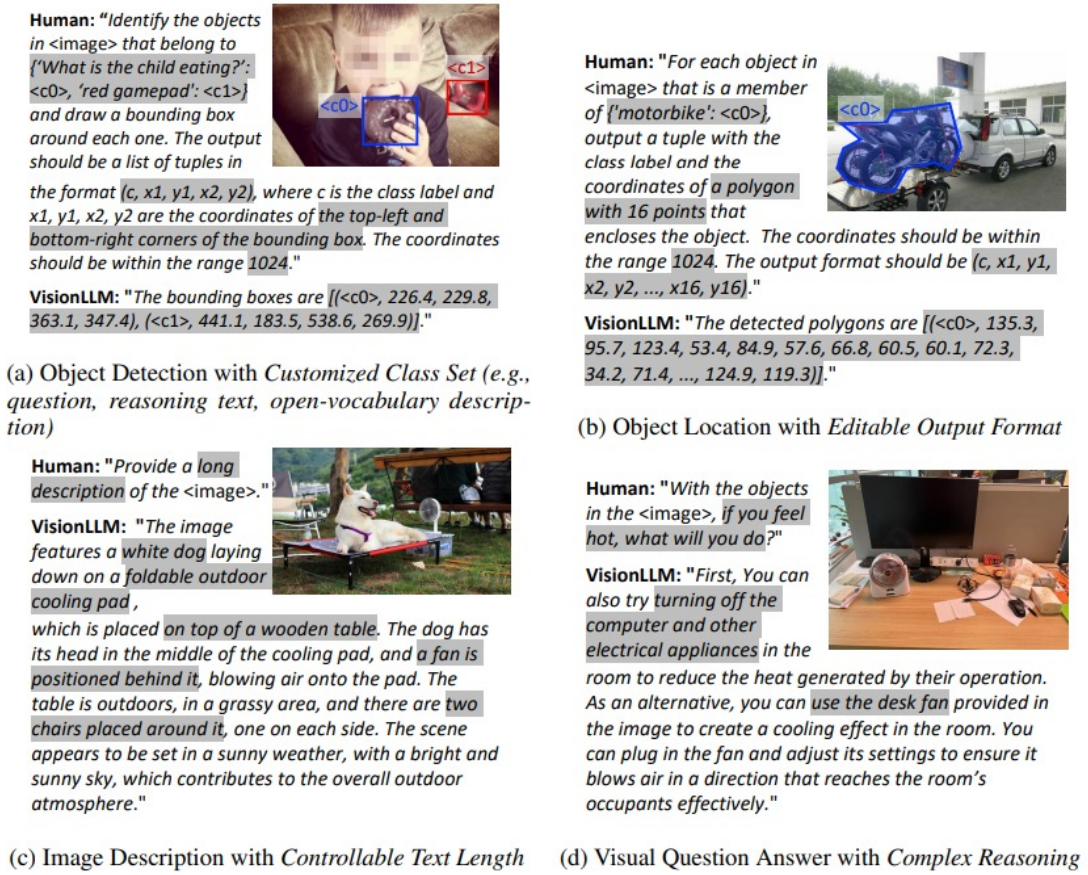


Figure 2: **Results and visualizations of our VisionLLM**. Guided by language instructions, our unified generalist framework showcases its effectiveness on diverse open-ended vision-centric tasks. The text marked with a gray background indicates the customized instructions and the desired outputs.

文献

LLM-based Frameworks: Prompt-manager-centric, with VFMs as plugins

Talking, Drawing and Editing with Visual Foundation Models

HuggingGPT

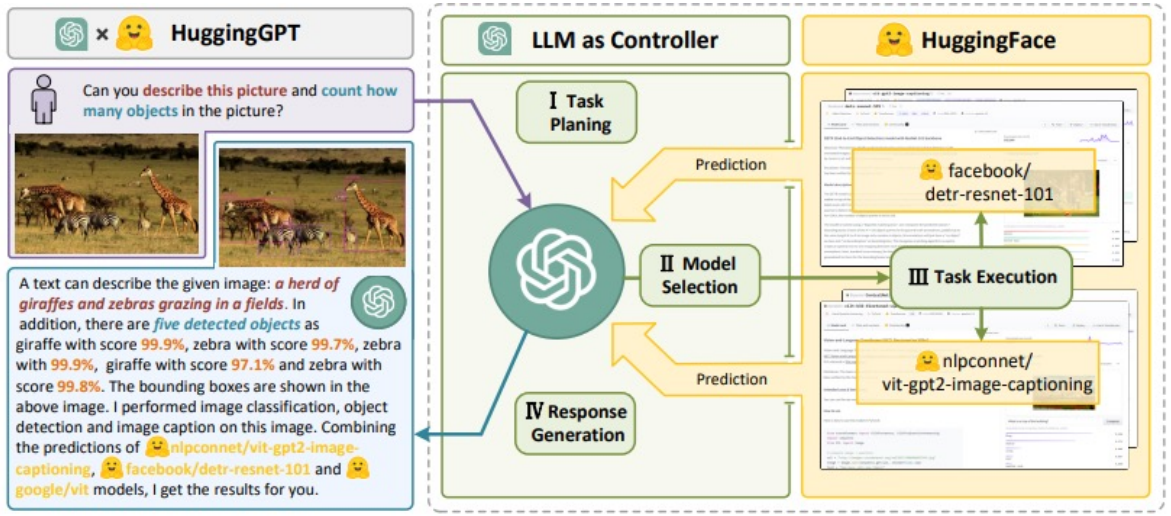


Figure 1: *Language serves as an interface for LLMs (e.g., ChatGPT) to connect numerous AI models (e.g., those in Hugging Face) for solving complicated AI tasks.* In this concept, an LLM acts as a controller, managing and organizing the cooperation of expert models. The LLM first plans a list of tasks based on the user request and then assigns expert models to each task. After the experts execute the tasks, the LLM collects the results and responds to the user.

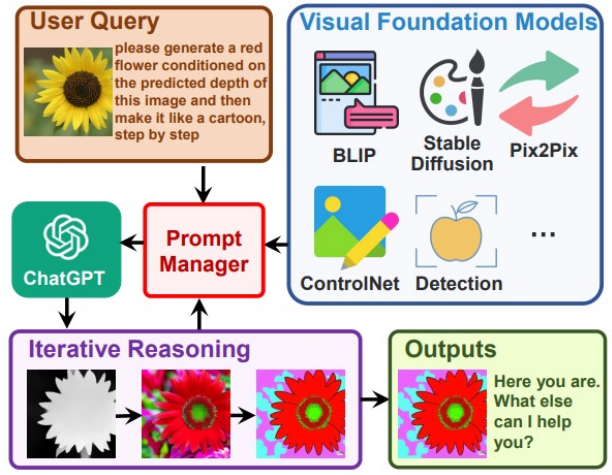


Figure 1. Architecture of Visual ChatGPT.

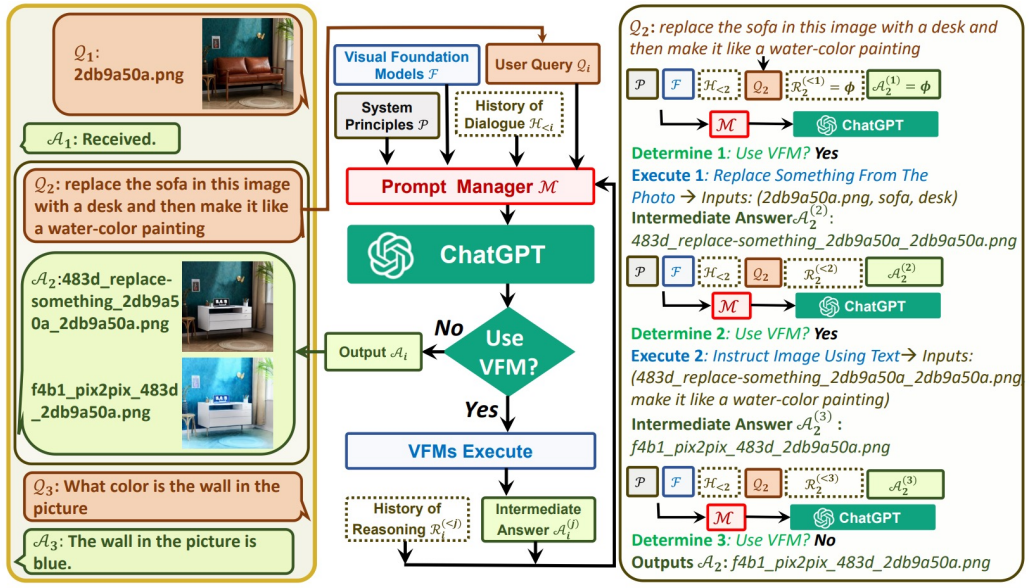


Figure 2. Overview of Visual ChatGPT. The left side shows a three-round dialogue, The middle side shows the flowchart of how Visual ChatGPT iteratively invokes Visual Foundation Models and provide answers. The right side shows the detailed process of the second QA.

LLM-based Frameworks: Reasoning & intention-driven – towards more comprehensive vision tasks

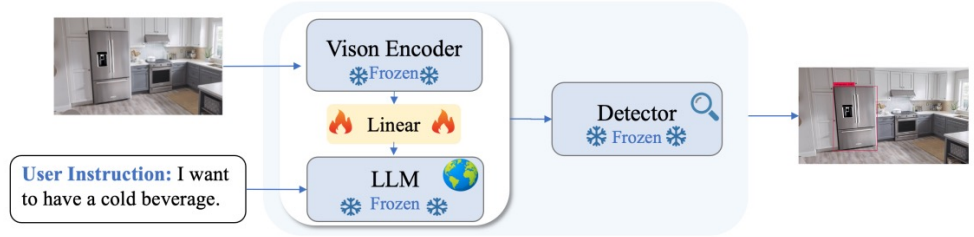


Figure 2: Framework of DetGPT. The multi-model model consisted of vision encoder and LLM interprets the user instruction, reasons over the visual scene, and finds objects matching the user instruction. Then, the object names/phrases are passed to the open-vocabulary detector for localization.

- **Visual encoder:** BLIP-2.
- **Language model:** Vicuna
- **Open-vocabulary detector:** GroundingDINO
- **Cross-modal alignment:** a linear projection layer (MiniGPT-4 like)

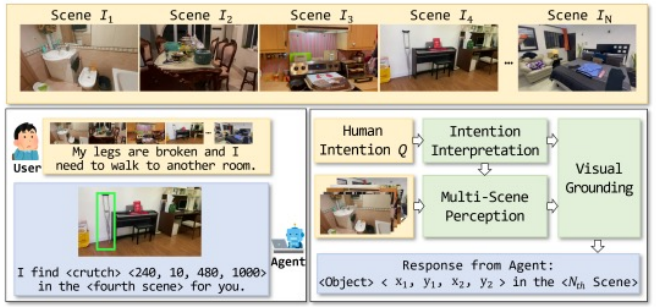


Figure 2: The illustration about the overall pipeline of our intention-driven visual grounding task, which mainly comprises intention interpretation, multi-scene perception and the subsequent visual grounding.

[Visual grounding (VG)] Understanding and Locating Open-World Objects Aligned with Human Intentions:

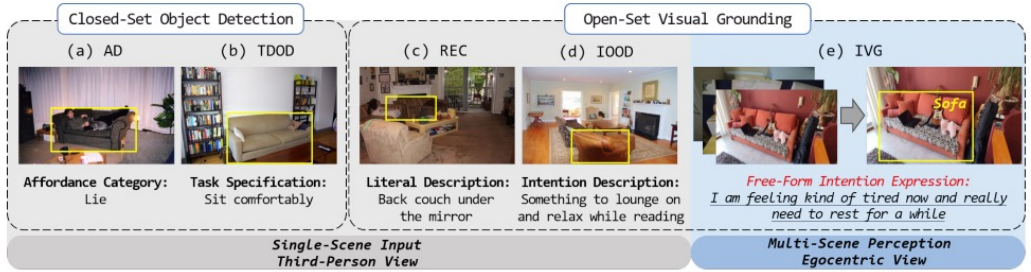


Figure 1: Task Comparison between Affordance Detection (AD), Task-Driven Object Detection (TDOD), Referring Expression Comprehension (REC), Intention-Oriented Object Detection (IOOD) and Intention-driven VG (IVG).

LLM-based Frameworks: Reasoning & intention-driven – towards more comprehensive vision tasks

LISA: REASONING SEGMENTATION VIA LARGE LANGUAGE MODEL – LLM outputs as seg-decoder's conditions

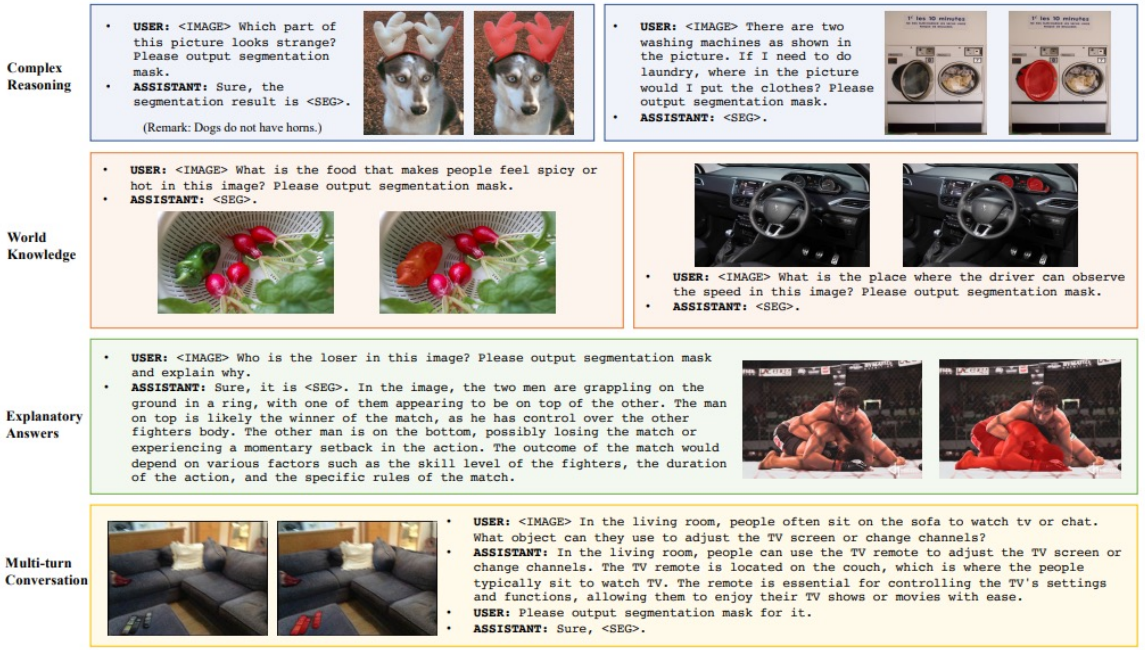


Figure 1: We unlock new segmentation capabilities for current multi-modal LLMs. The resulting model (LISA) is capable to deal with cases involving: 1) Complex Reasoning; 2) World Knowledge; 3) Explanatory Answers; 4) Multi-turn Conversation.

1) complex reasoning;
2) world knowledge;
3) explanatory answers;
4) multi-turn conversation.

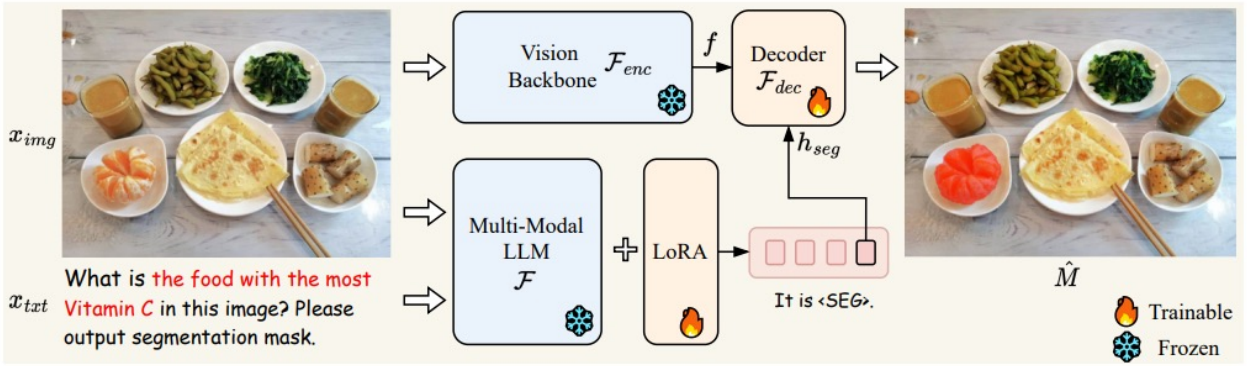


Figure 3: The pipeline of LISA. Given the input image and text query, the multi-modal LLM generates text output. The last-layer embedding for the <SEG> token is then decoded into the segmentation mask via the decoder. The choice of vision backbone can be flexible (e.g., SAM, Mask2Former).

LLM-based Frameworks: Interacting beyond language

Combining pointing and language instructions to accomplish complex vision-centric tasks.

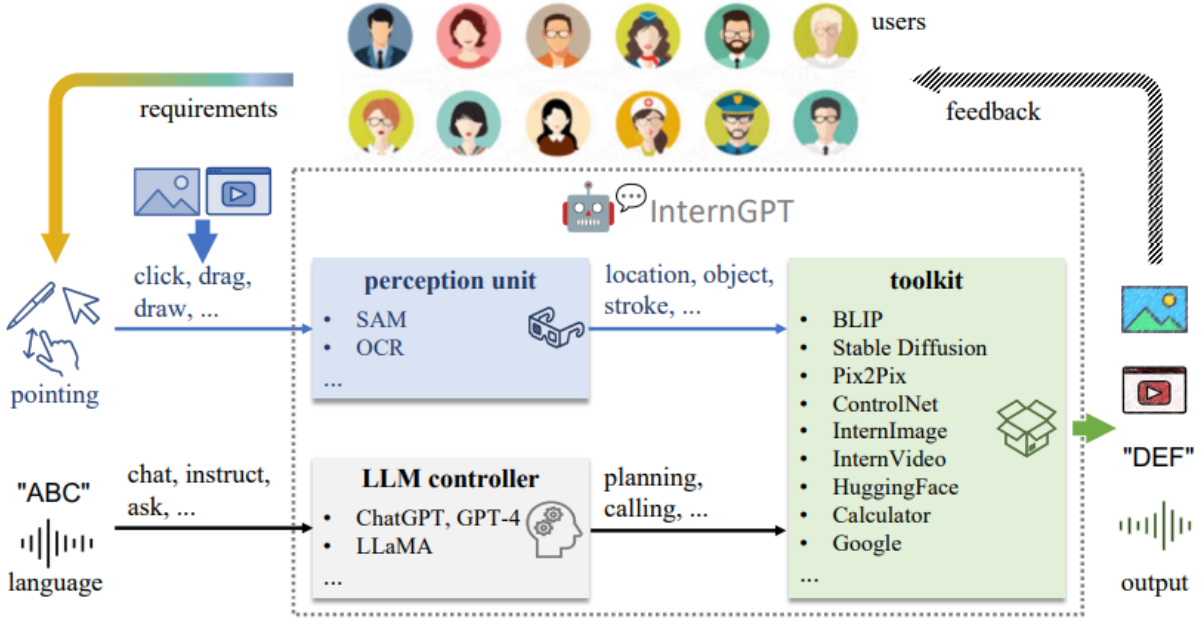
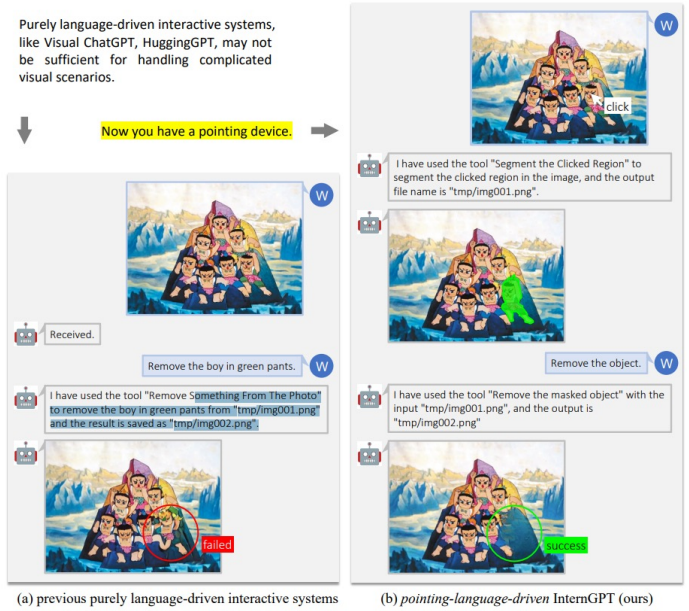


Figure 3: **Overall architecture of InternGPT**. It has three main components: perception unit, LLM controller, and open-world toolkit.

Operates at 3 levels:

- Level 1: basic interaction.
- Level 2: language-guided interaction.
- Level 3: pointing-language enhanced interaction.



(a) previous purely language-driven interactive systems (b) *pointing-language-driven* InternGPT (ours)

Figure 1: **Advantage of our *pointing-language-driven* interactive system.**

结论

Advantages:

- better handles multiple tasks,
- better process on multiple modalities,
- can directly conduct zero-shot inference on novel tasks that do not appear in the pre-training stage,
- enables more effective and sophisticated interactions between humans and machines,
- paving the way for novel techniques that blur the lines between human and machine intelligence.

Limitation:

- Accuracy & performance
- Stability
- Efficiency & real-time scenes
- Inconsistency between training & inference

Potential:

- Embodied & RL: generalist agents
- 3D & motion: special design

参考文献

- Alayrac, J. B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., ... & Simonyan, K. (2022). Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, *35*, 23716-23736.
- Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., ... & Wang, L. (2022). Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*.
- Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., ... & Yang, H. (2022, June). Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning* (pp. 23318-23340). PMLR.
- Chen, T., Saxena, S., Li, L., Lin, T. Y., Fleet, D. J., & Hinton, G. E. (2022). A unified sequence interface for vision tasks. *Advances in Neural Information Processing Systems*, *35*, 31333-31346. Gan, Y., Park, S., Schubert, A., Philippakis, A., & Alaa, A. M. (2023). InstructCV: Instruction-Tuned Text-to-Image Diffusion Models as Vision Generalists. *arXiv preprint arXiv:2310.00390*.
- Yang, Z., Gan, Z., Wang, J., Hu, X., Ahmed, F., Liu, Z., ... & Wang, L. (2022, October). Unitab: Unifying text and box outputs for grounded vision-language modeling. In *European Conference on Computer Vision* (pp. 521-539). Cham: Springer Nature Switzerland.
- Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., & Duan, N. (2023). Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.

参考文献

- Liu, Z., He, Y., Wang, W., Wang, W., Wang, Y., Chen, S., ... & Qiao, Y. (2023). Internchat: Solving vision-centric tasks by interacting with chatbots beyond language. *arXiv preprint arXiv:2305.05662*.
- Zhu, J., Zhu, X., Wang, W., Wang, X., Li, H., Wang, X., & Dai, J. (2022). Uni-perceiver-moe: Learning sparse generalist models with conditional moes. *Advances in Neural Information Processing Systems*, *35*, 2664-2678.
- Li, H., Zhu, J., Jiang, X., Zhu, X., Li, H., Yuan, C., ... & Dai, J. (2023). Uni-perceiver v2: A generalist model for large-scale vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2691-2700).
- Zhu, X., Zhu, J., Li, H., Wu, X., Li, H., Wang, X., & Dai, J. (2022). Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 16804-16815).
- Chen, Y., Zhang, S., Han, B., & Jia, J. (2023). Lightweight in-context tuning for multimodal unified models. *arXiv preprint arXiv:2310.05109*.
- Li, J., Li, D., Savarese, S., & Hoi, S. (2023). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.
- Zhu, D., Chen, J., Shen, X., Li, X., & Elhoseiny, M. (2023). Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

参考文献

- Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2024). Visual instruction tuning. *Advances in neural information processing systems*, *36*.
- Bar, A., Gandelsman, Y., Darrell, T., Globerson, A., & Efros, A. (2022). Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, *35*, 25005-25017.
- Wang, X., Zhang, X., Cao, Y., Wang, W., Shen, C., & Huang, T. (2023). Seggpt: Segmenting everything in context. *arXiv preprint arXiv:2304.03284*.
- Wang, X., Wang, W., Cao, Y., Shen, C., & Huang, T. (2023). Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6830-6839).
- Pi, R., Gao, J., Diao, S., Pan, R., Dong, H., Zhang, J., ... & Zhang, L. K. T. (2023). DetGPT: Detect What You Need via Reasoning. *arXiv preprint arXiv:2305.14167*.
- Wang, W., Zhang, Y., He, X., Yan, Y., Zhao, Z., Wang, X., & Liu, J. (2024). Beyond Literal Descriptions: Understanding and Locating Open-World Objects Aligned with Human Intentions. *arXiv preprint arXiv:2402.11265*.
- Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., & Jia, J. (2023). Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*.

感谢观看

https://www.shlab.org.cn
Shanghai Artificial Intelligence Laboratory