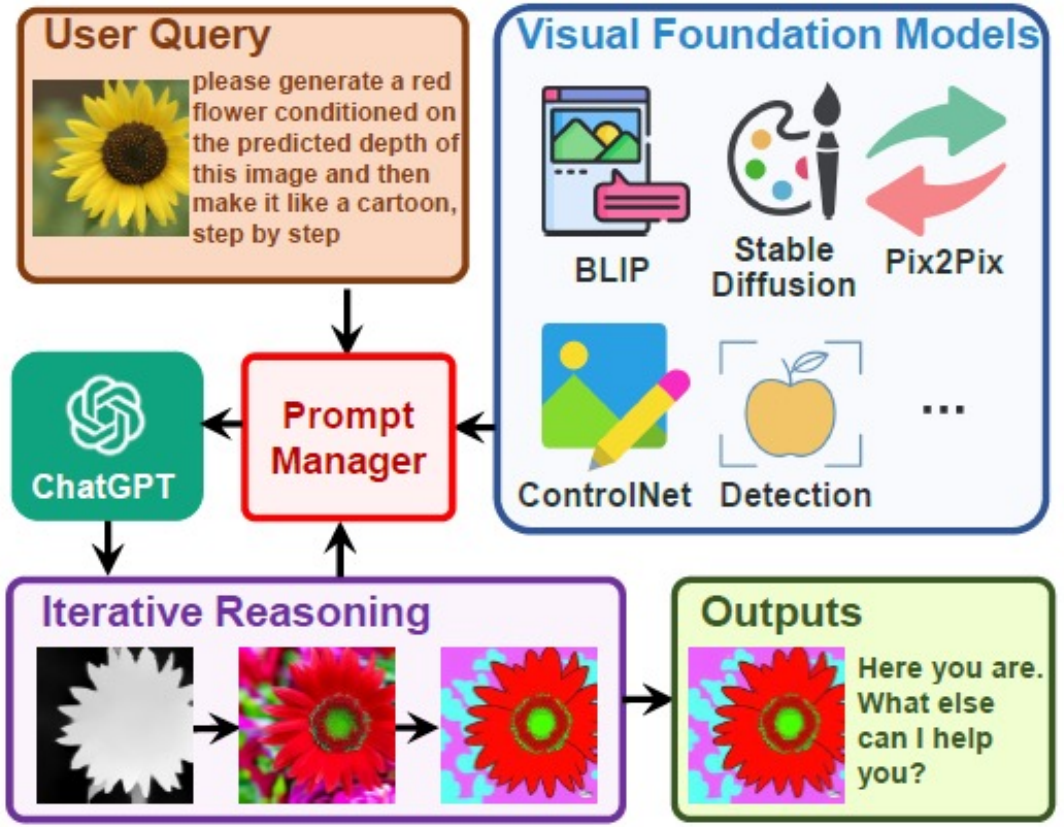
# 目录

# LVLM的能力扩充: 输出空间的扩展

- 从 LLM 到 LVLM:
  - 完成了输入空间的扩展
  - 通过图文对进行输入空间的对齐
  - 自然地通过LLM基座以文本方式进行输出
- 多模态大模型可以输出离散token以外的输出吗?
  - 连续型输出: 坐标, 标记框 ...
  - 其他模态: 图片, 音频, 3D 点云...

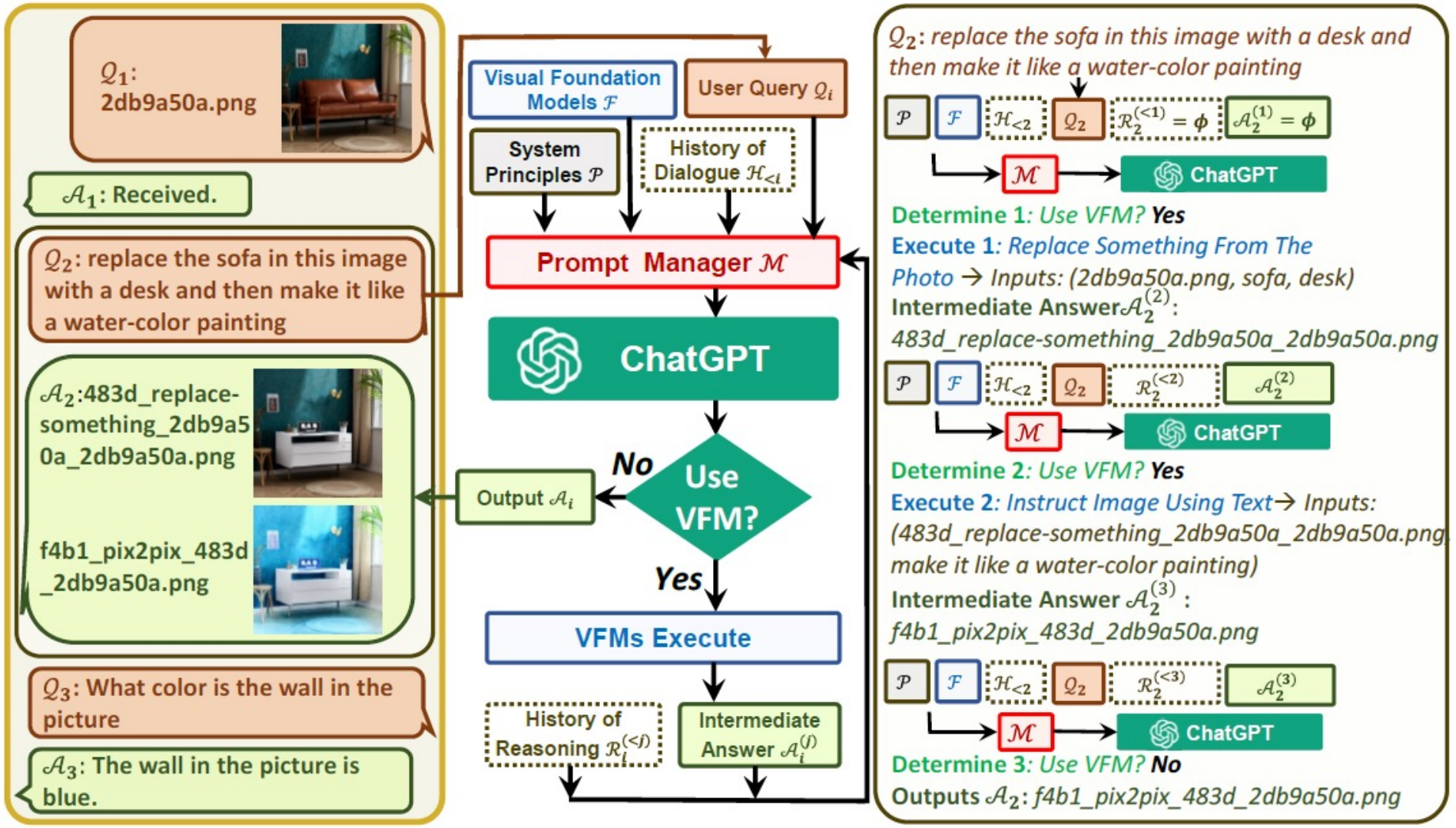
# Visual ChatGPT： 以Zero-Shot方式使用工具

- **通过文本指令来使用工具!**
- **基座：**ChatGPT
  - 泛用而灵活的系统
  - 局限于文本输入 / 输出
- **工具：**视觉基础模型 (VFM)
  - 具有特定方面的视觉能力
- **输出空间：**基于工具得到扩展
  - 图片, 物体标记框…
- **拓展方法:**
  - 通过 “**prompts** manager”
  - **Zero-shot** 拓展方式， 基于ChatGPT




Visual chatgpt: Talking, drawing and editing with visual foundation models， 2023

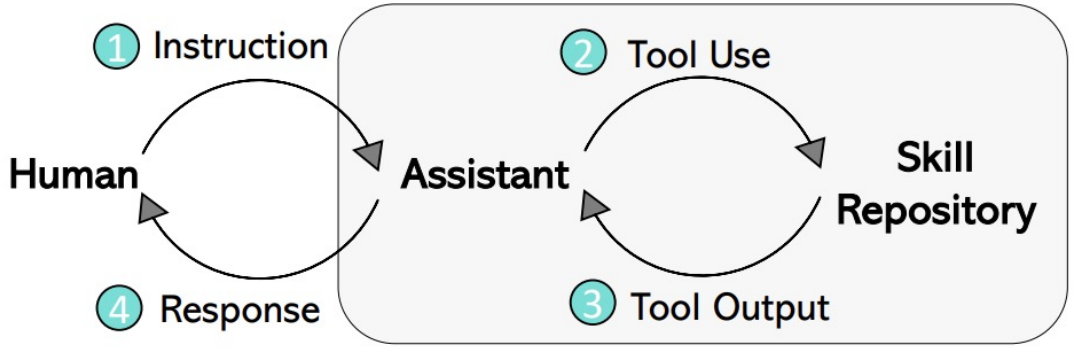
# Visual ChatGPT :以Zero-Shot方式使用工具




Visual chatgpt: Talking, drawing and editing with visual foundation models， 2023

# LLaVA-Plus: 训练模型学习工具的使用

- **基座：**LLaVA (或其他任意 LVLMs)
- **输出空间:**
  - 图片: 基于 Stable Diffusion
  - 图分割: 基于 SAM
  - 标记框: 基于物体检测器
  - ...




- **拓展方法：**
  - 4-轮对话的形式
  - **通过构建的数据训练模型学习**遵循使用工具的指令

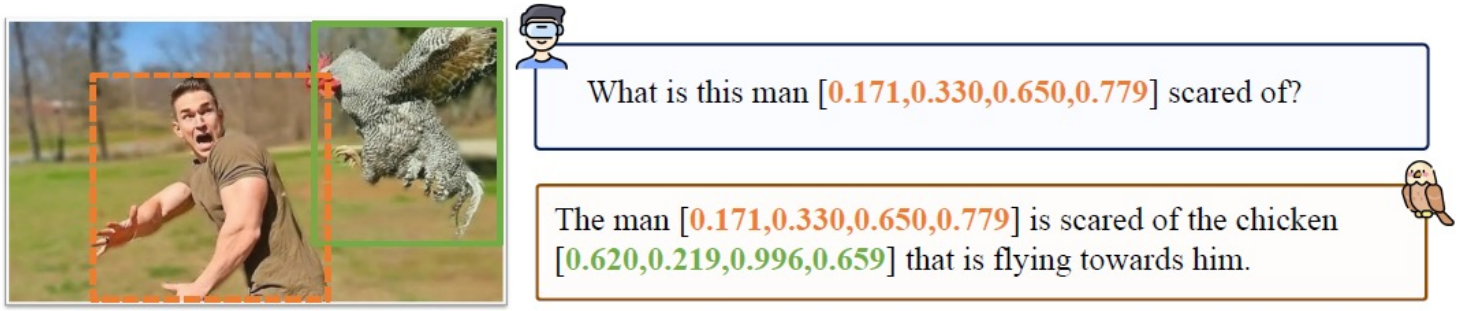
$$\text{Human}: \mathbf{I}_q <\backslash n> \mathbf{X}_q \text{<STOP>} \; \text{Assistant}: \mathbf{X}_{\text{skill\_use}} \text{<STOP>}$$

$$\text{Human}: \mathbf{X}_{\text{skill\_result}} \text{<STOP>} \; \text{Assistant}: \mathbf{X}_{\text{anwser}} \text{<STOP>}$$

Llava-plus: Learning to use tools for creating multimodal agents，2023

# Shikra：以文本表示连续的数值

- **输出空间：** 连续的坐标
- **拓展方法：**
  - 以 **自然语言的形式** 来表示连续的数值




Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic，2023

  - 指令遵循训练数据的构建:
    - 重构已有的数据: RefCOCO, Visual-7W, visual genome, Flickr30k entities
    - 生成的QA数据: 基于 Flickr30k entities 数据通过GPT-4生成
  - 训练阶段-1: 使用重构的数据
  - 训练阶段-2: LLaVA + 生成的QA数据

# Kosmos2：以扩展词表的形式进行Grounding

- **输出空间:** 标记框
- **拓展的方法:** 拓展词表
  - 位置 tokens: P*P tokens 来表示 P*P 个图片里的分块
  - 特殊 tokens: 以markdown里 **超链接** 形式进行表示
    - 文本描述 <box>标记框</box>
    - <grounding> 作为一个**开关**来指示模型是否需要进行grounding

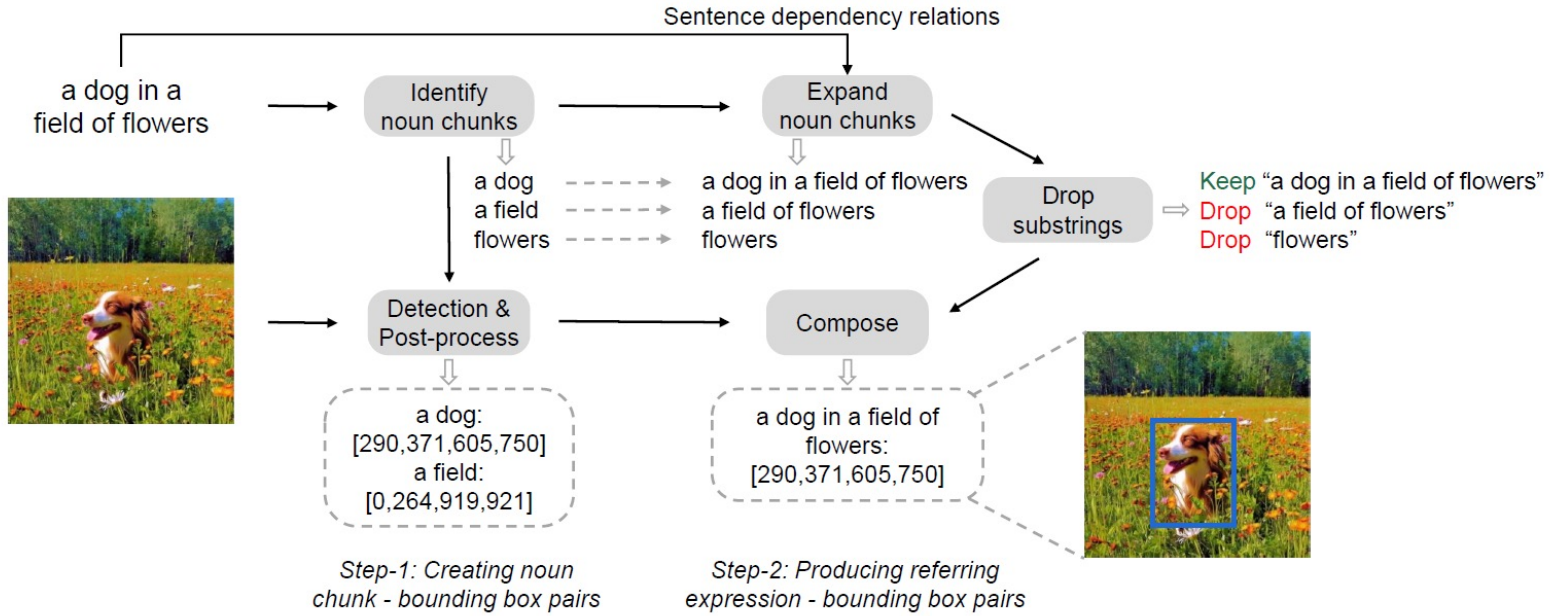
<s> <image> Image Embedding </image> <grounding>  It <box><loc$_{44}$><loc$_{863}$></box> seats next to  a campfire <box><loc$_{4}$><loc$_{1007}$></box> </s>

- 预训练: 图文对 + 文本数据 + GRIT
- 指令微调: LLaVA + unnatural instructions + GRIT

Kosmos-2: Grounding Multimodal Large Language Models to the World，2023

# Kosmos2: GRIT （Grounded Image-Text） 构建

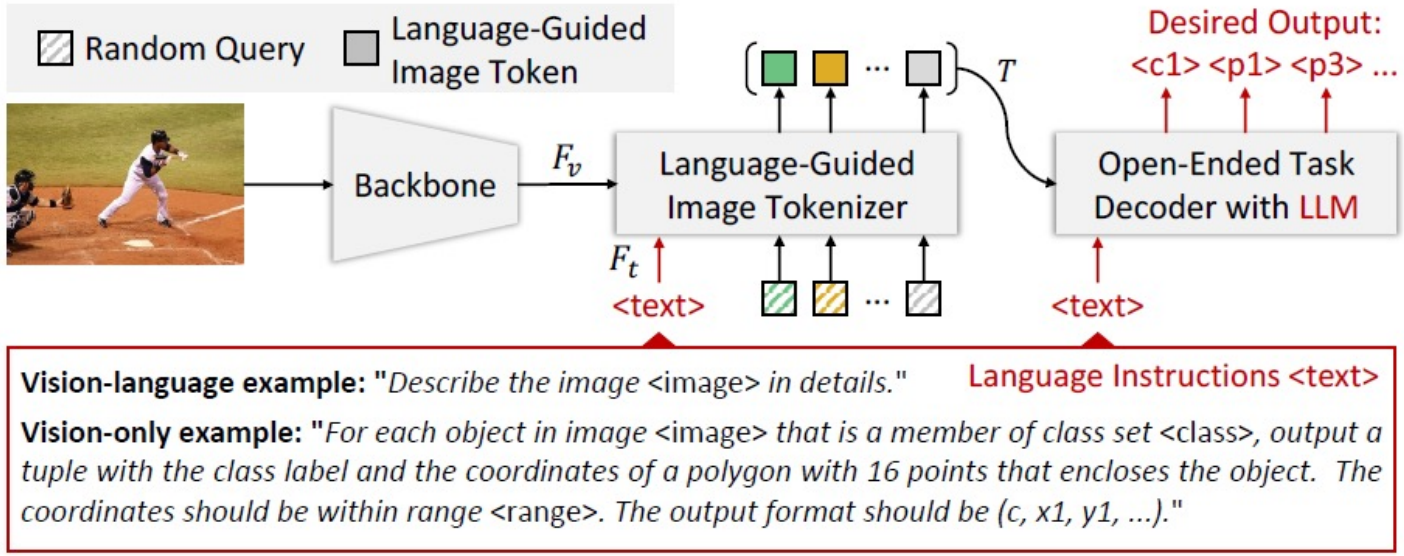
- 步骤-1 构造名词短语-标记框对: SpaCy + GLIP
- 步骤-2 构造referring-expression-标记框对
- 通过不断聚合语义树的节点将名词短语拓展到referring expression
- 舍弃被其他描述（referring expression）包含的项




**Kosmos-2: Grounding Multimodal Large Language Models to the World，2023**

# VisionLLM: 更丰富的词表扩充

- **输出空间:** 分类类别 + 坐标
- **拓展方法：**在输入指令里扩展: 任务描述 + 输出格式的定义

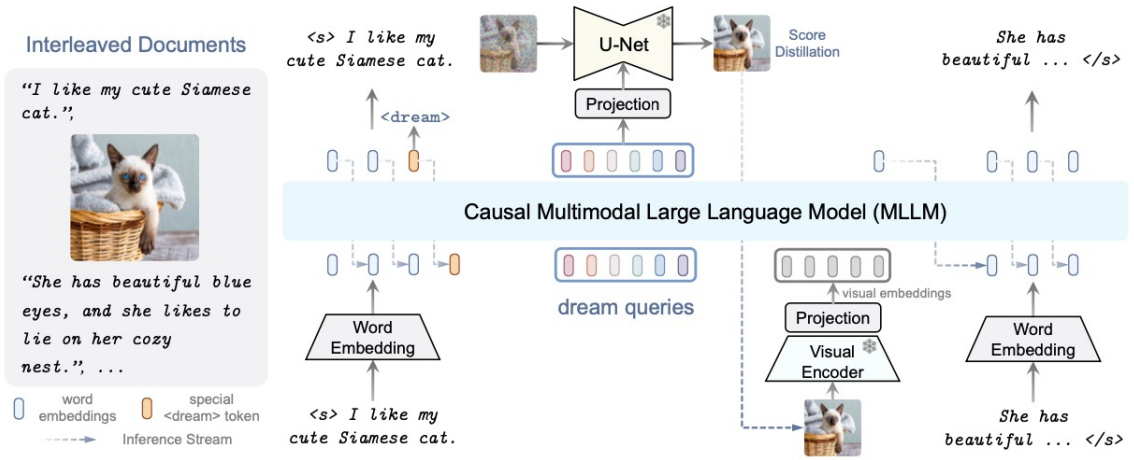



- 词表内进行扩充:
  - 512 位置 tokens: 表示坐标 + 类别 tokens: 作为类别的index
  - 输出tokens: 用来表述输出的形式，并在已知输出形式的情况下进行高效的解码

Visionllm: Large language model is also an open-ended decoder for vision-centric tasks， 2023

# DreamLLM：引入图文交错的输出形式

- **基座：** CLIP + Vicuna + Stable Diffusion
- **输出空间：** 与文本交错的图片
- **拓展方法：**
  - <dream> token 占位符指示产生图片的位置
  - 引入可学习的dream queries

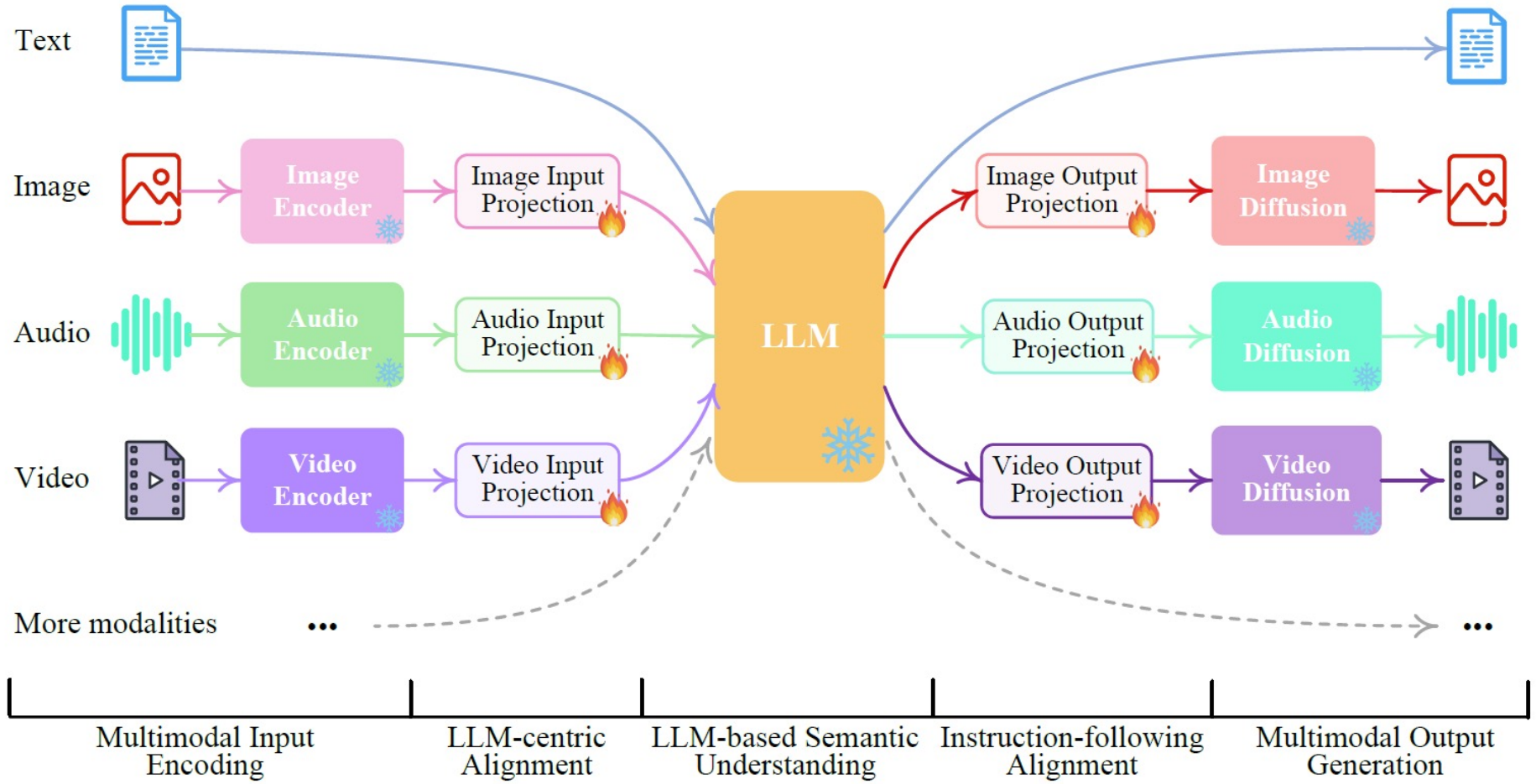



- **数据构建：**
  - MMC4中的多模态文档
  - 利用GPT-4从文档构建指令相关的QA对

**Dreamllm: Synergistic multimodal comprehension and creation，2023**

# Next-GPT: 任意模态的生成

- **输出空间:** 任意模态、模态交错的信息




Next-gpt: Any-to-any multimodal llm, 2023

# Next-GPT：任意模态的生成

- **输出空间：**任意模态、模态交错的信息
- **拓展方法：**在生成端引入模态信息占位符
  - **指示特定位置生成特定模态信息，**E.g. <IMG0><IMG1><IMG2><IMG3>
    指示图片生成，占位符对应的表示作为**对应模态解码器的输入**
- 指令微调数据集:
  - **文本 + X – 文本**: LLaVA, miniGPT-4, VideoChat
  - **文本 – 文本 + X**: 基于 X-描述 数据构造
  - **MosIT**: 构造的 5K 对话
    - 基于GPT-4的Self-instruct方法: 构造多轮、多模态、模态交互的对话
    - 搜集最匹配的对应模态数据: Youtube, StableDiffusion, Midjourney
    - 人工筛选，保证质量

**Next-gpt: Any-to-any multimodal llm, 2023**

# LVLM的能力扩充：特定任务上能力的增强

- 在特定任务上 LVLMs 和对应的 SOTAs 仍有差距 (数据来自 Qwen-VL):

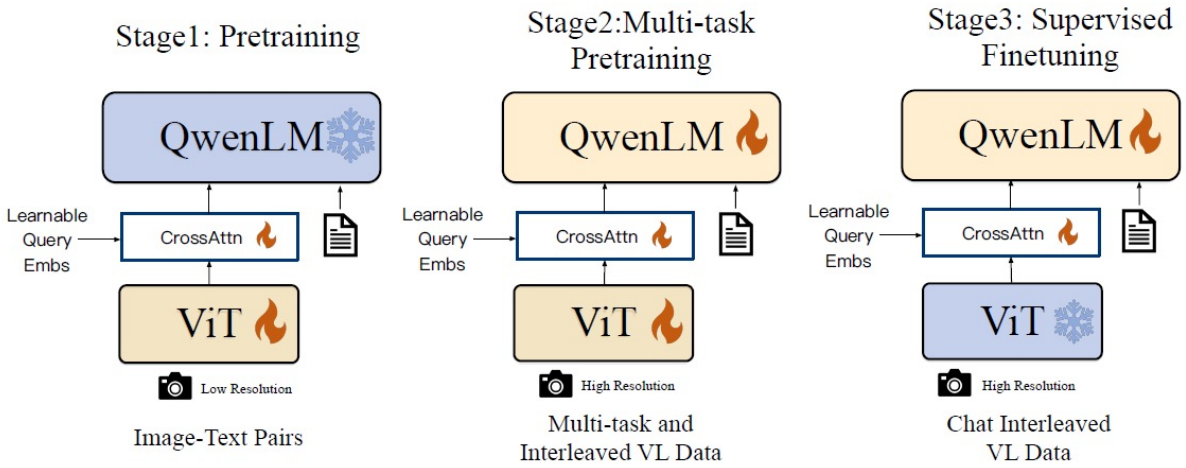
| Model | Nocaps | Flickr30K | VQA v2 | OKVQA | GQA | SciQA-Img | Vizwiz |
|---|---|---|---|---|---|---|---|
| BLIP-2 | 103.9 | 71.6 | 65.0 | 45.9 | 32.3 | 61.0 | 19.6 |
| Specialist SOTAs | 127.0 (PALI) | 84.5 (InstructBLIP) | 86.1 (PALI-X) | 66.1 (PALI-X) | 72.1 (CFR) | 92.5 (LLaVA) | 70.9 (PALI-X) |

  - **Zero-shot** LVLMs v.s **Fine-tuned** SOTAs
  - LVLM没有学习过特定任务的输出输入结构信息

- LVLM能在特定任务上缩小和SOTA的差距吗?
  - 主要关注的任务: VQA, Object Grounding, Image Captioning

# Qwen-VL：多任务学习

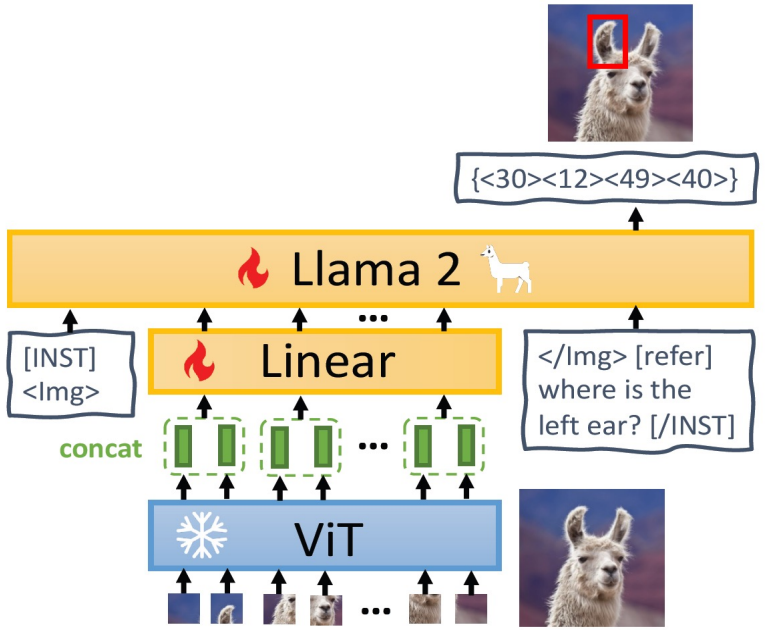
- **视觉编码器:** OpenCLIP ViT-bigG (448 px)
- **LLM：**Qwen-7B；　**连接模块:** 单层的cross-attention模块
- **3-阶段的训练框架：**
  - **预训练:** 大规模, 弱关联的图文对
  - **多任务学习:** 高质量数据 (VQA, Caption, Grounding, OCR)
  - **指令微调:** 基于指令遵循数据 (多模态 + 文本)




**Qwen-vl: A frontier large vision-language model with versatile abilities，2023**

# MiniGPT4-v2：额外引入任务指示符

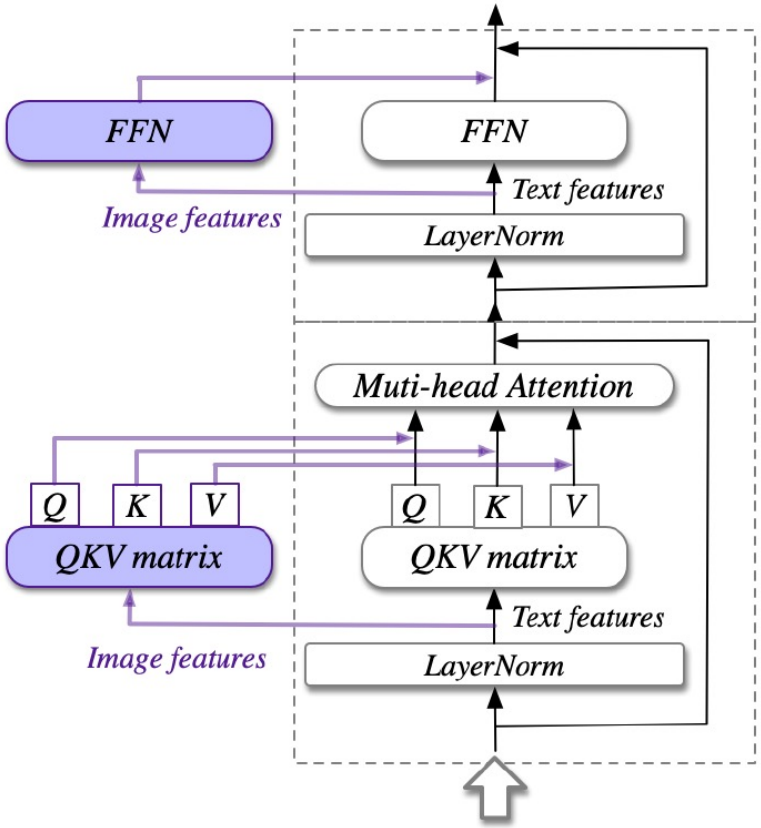
- **视觉编码器:** EVA-ViT (448 px)
- **LLM：** LLaMA-2
- **连接模块:** 拼接邻接的4个tokens进行Linear
- **3-阶段训练:**
  - **引入任务指示符**: [vqa], [grounding], [refer]..
  - **预训练:** 主要学习弱关联的图文对
  - **多任务学习:** 仅细粒度的任务**数据** (VQA + Caption + Grounding)
  - **指令微调:** 指令遵循数据（多模态 + 文本）




Minigpt-v2: large language model as a unified interface for vision-language multi-task learning，2023

# CogVLM:引入视觉专家模块

- **视觉编码器:** EVA2-CLIP-E (490 px)
- **LLM：** Vicuna-7B-v1.5 + 视觉专家模块;
- **连接模块:** MLP层
- **预训练：**
  - LAION + 基于 Kosmos2 构造的 grounding 数据
- **SFT对齐训练：**
  - LLaVA, LLaVAR, LRV-Instruction, 非公开数据
- 下游任务上的 **Fine-tuning:**
  - Captioning, VQA, visual grounding




CogVLM: Visual Expert for Pretrained Language Models，2023

# 小结

- 真正的多模态模型必然是全模态支撑的
- 以大语言模型作为大脑是目前的主流架构
- 编码端可以进行语义对齐
- 语义空间引入其他模态的词汇，扩充输出可能性
- 解码端引入其他工具，完成输出

- **训练数据的生成：多模态混合的数据样本还是远远小于文本模态**