



# A Survey on Multimodal Large Language Models

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun,  
Tong Xu, and Enhong Chen. A survey on multimodal large  
language models. arXiv preprint arXiv:2306.13549, 2023

# 引言

尽管大语言模型（LLM）在大多数自然语言处理（NLP）任务中表现出了令人惊讶的样本零/少样本推理性能，但因为它们只能理解离散文本，本质上对视觉“视而不见”。同时，大型视觉模型（LVM）具有一定的视觉能力，但缺乏推理能力。多模态大模型是LLM与LVM相向而行的产物。

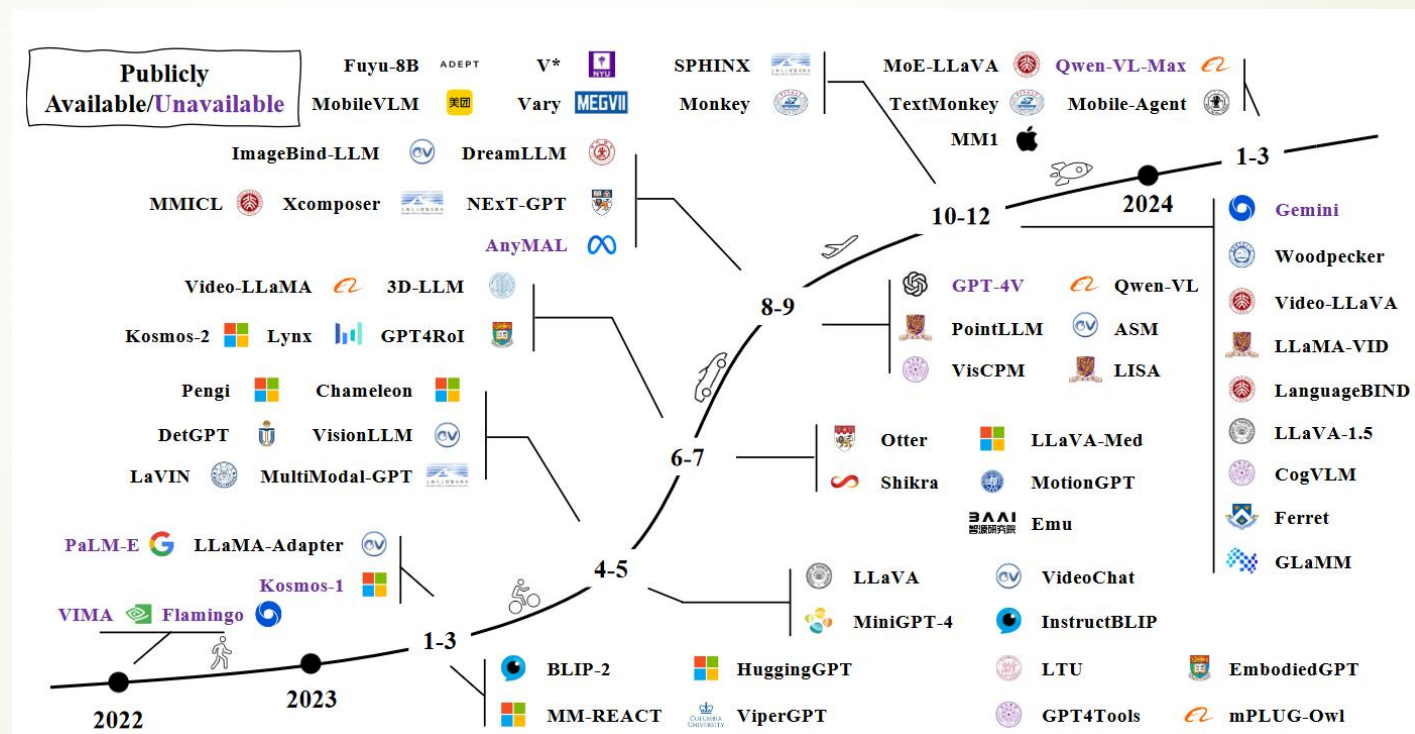


图1 代表性MLLM时间线

# 引言

多模态大模型（Multimodal Large Language Models, MLLM），将大模型作为大脑，融合不同模态数据，以处理多模态任务。在MLLM前，多模态学习可以分为判别式和生成式。

判别式模型的目的是学习从输入数据到输出标签的映射关系，常面向分类、回归任务；

生成式模型试图理解数据是如何生成的，包括它的概率分布和内在结构，常面向数据生成、数据增强、风格转换任务。

本文从以下角度展开综述：

- 基本表述与相关概念：MLLM的架构、训练策略、数据、评估；
- MLLM的研究趋势：支持更高粒度、模式、语言、场景；
- MLLM的研究要点：多模态幻觉、上下文信息、思维链、辅助视觉推理；
- 讨论现有的挑战与展望。

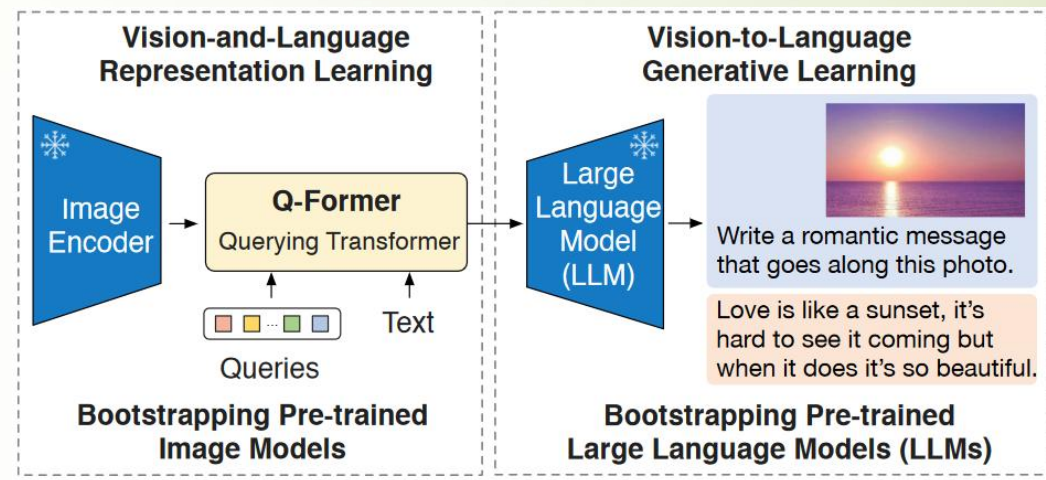


图2 生成式MLLM举例

# 主流MLLM架构

经典的MLLM结构包括三个部分：预训练模态编码器、预训练大模型，以及连接它们的模态接口。

其中，预训练模态编码器用于将多模态信号进行预处理，LLM根据处理后的信号信息进行推理，模态接口将不同模态的语义表示整合到一个统一的表示空间中，以便后续联合处理。

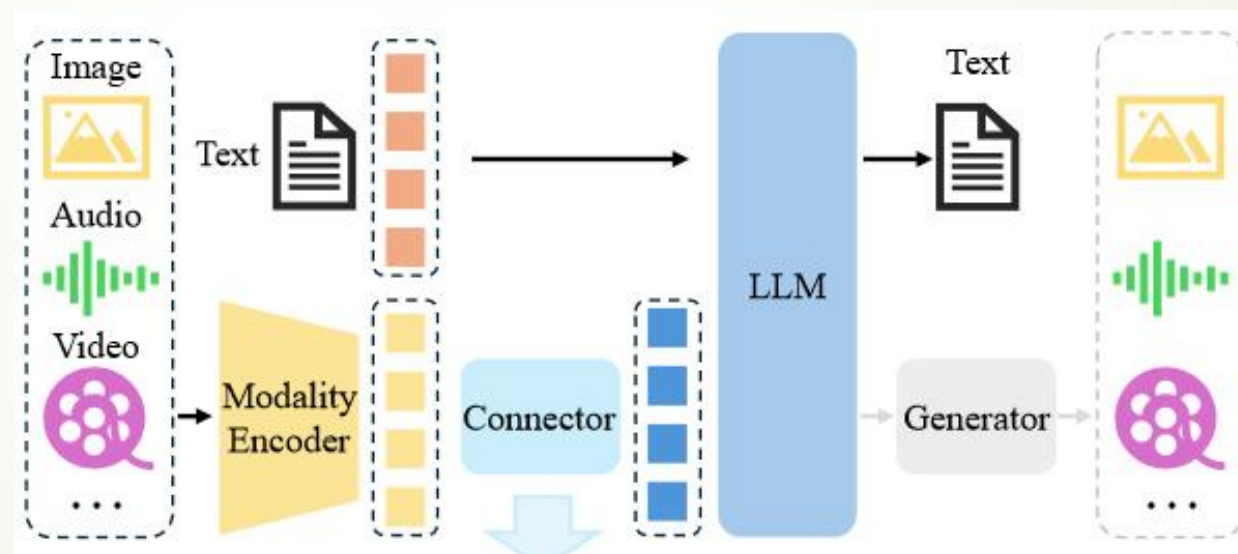


图3 一种多模态大模型架构示意图

# 模块1：模态编码器

模态编码器：将原始多模态信息压缩为紧凑的表示

一般先对模态编码器进行预训练，使得后续与LLM对齐时相对容易。

CLIP：将图片和文本编码联合构成多模态嵌入空间

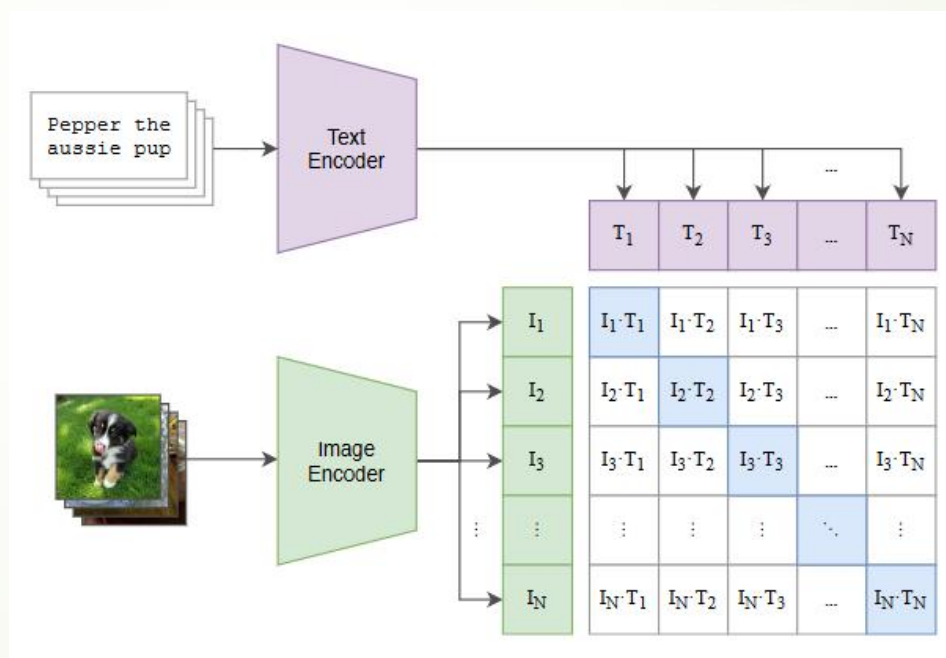


图4 CLIP构建多模态嵌入空间示意图



# 模态编码器的解析度

使用更高分辨率的模态编码器可以显著提升性能，具体方法分为直接缩放和块分割方法。

**直接缩放法**包括对编码器进一步调参、替换更高分辨率编码器、双编码器机制分别处理高低分辨率图像、通过交叉注意力将高分辨率特征注入到低分辨率分支中。

**块分割方法**将高分辨率图像切割成块，并重用低分辨率编码器。分块的子图像与下采样的高分辨率图像相结合，可以分别捕获局部和全局特征。

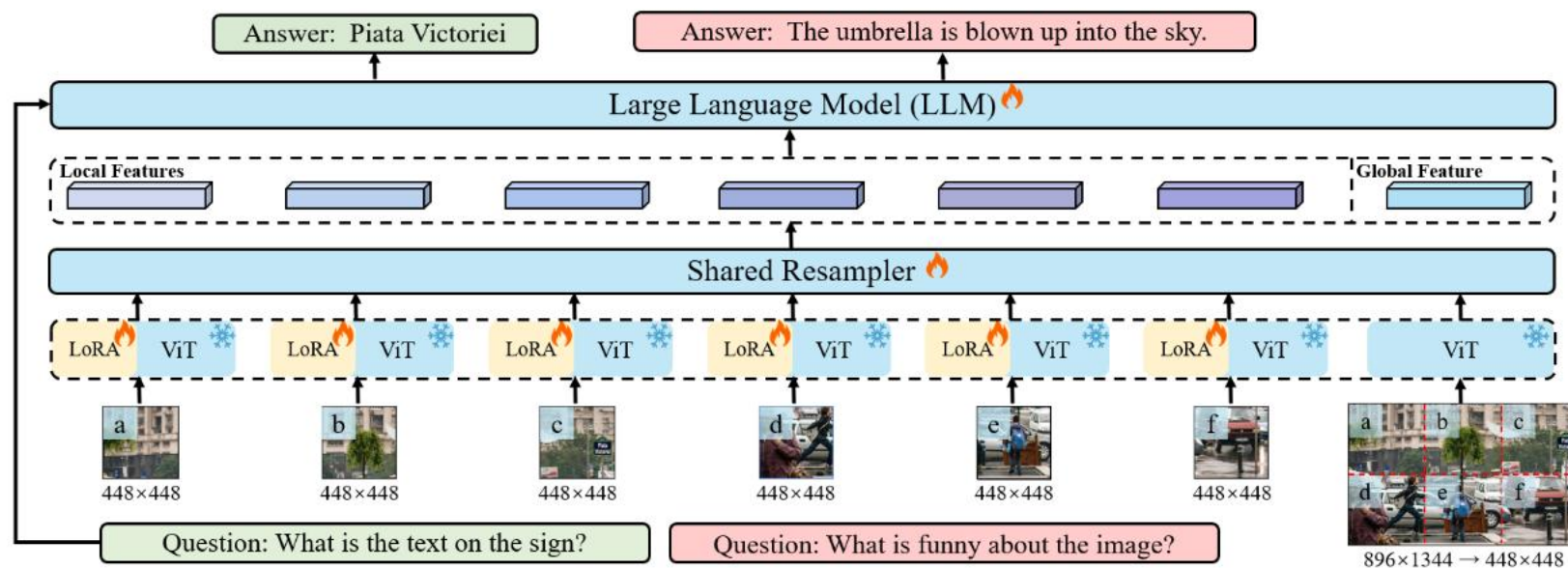


图5 块分割方法示意图

## 模块2：预训练大模型

参数量越大，效果越好；

部分较小LLM可在移动端部署：MobileVLM系列使用缩小规模的LLaMA（称为MobileLLaMA 1.4B/2.7B），能够在移动处理器上进行高效推理。

混合专家模型：通过将稀疏MoE层替换Transformer的FFN层，以获得更好的预测性能。通常包括以下部分：  
一个门控机制和一套门控输出机制：  
合并、平衡专家的选择；  
一套专家选择机制，根据门控输出选择专家进行预测；  
一套训练机制。

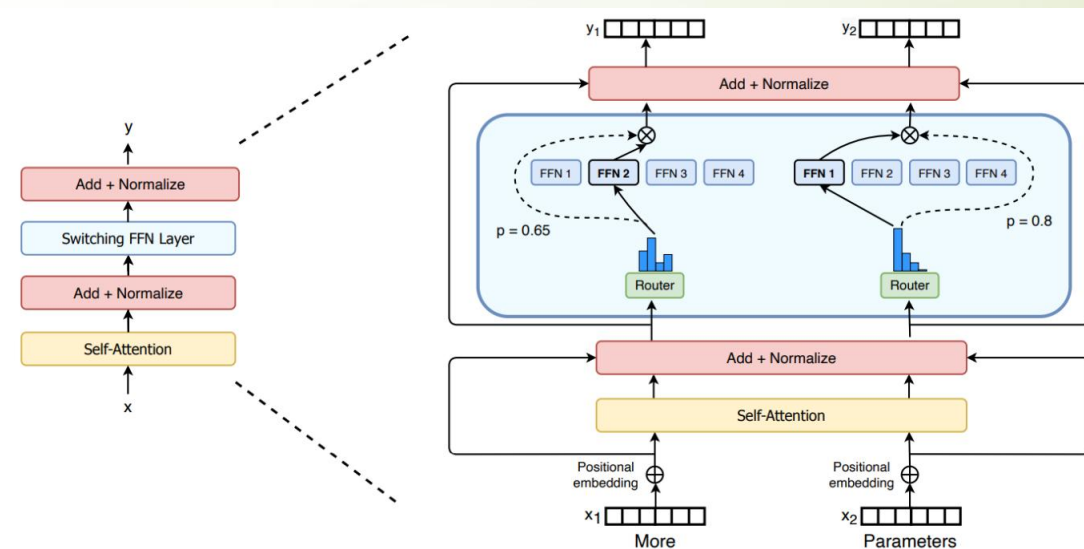


图6 Switch Transformers的MoE示意图

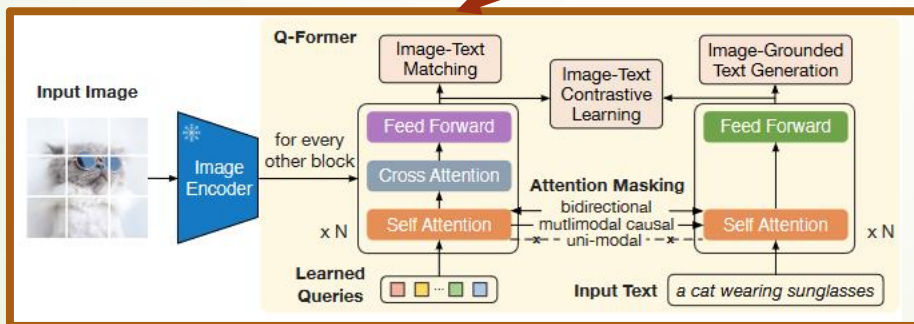
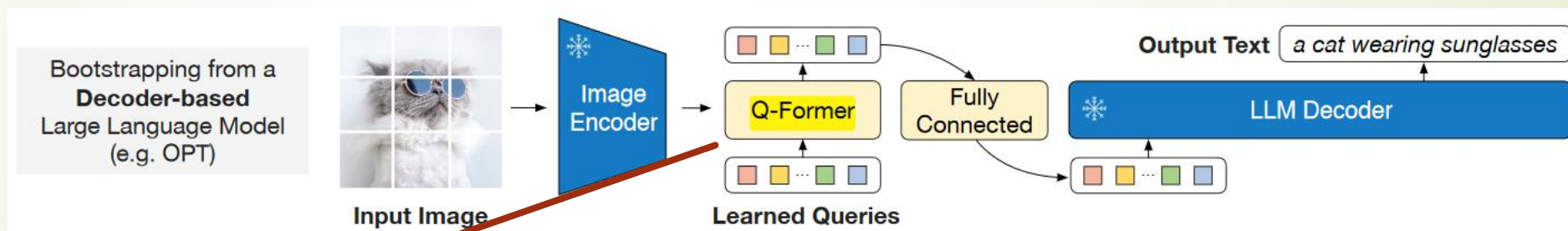
## 模块3：模态接口

模态接口：为了弥合LLM的文本输入限制与其他形式信息的差距。

一般方法有：引入可学习的连接器（令牌级或特征级融合），或者将图像翻译为语言文本送入LLM，也可以直接利用预先训练的专家模型。

模态接口的整体参数量一般占用全部参数的不到1%。

令牌级融合连接器：编码器的输出转化为令牌，与文本令牌连接。例如通过可学习的查询向量，结合自注意力、交叉注意力。



特征级融合连接器：引入额外模块，如交叉注意力层、模态专家网络，或部分参数（如QKV的权重矩阵）由预训练LLM初始化。

图7、8 BLIP-2涉及的模态接口



# 训练策略

MLLM的完整训练分为预训练、指令调优和对齐调节。

**预训练：**一般冻结部分已经预训练的模块（如编码器和LLM），学习可学习的模态接口，从而在不丢失预训练知识的前提下，学习模态协调的关系。数据方面数量大，应能够协调不同模式、提供世界知识，分为粗粒度数据（来自互联网、粗糙、更多）和细粒度数据（描述更长、更准确、更少）。

**指令调优：**通过自然语言指令，使模型能够泛化到未见任务。

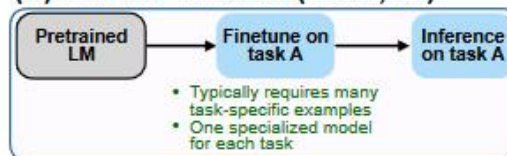
**预训练-微调：**基于大量数据集，面向特定任务训练；

**提示词：**减少对大数据集的依赖，通过提示工程完成专门的任务；

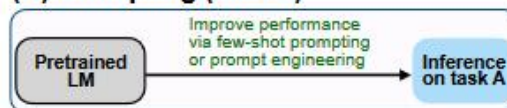
**指令调优：**可增强模型的泛化能力，与多任务提示词相关。

**训练数据方面，**质量和数量同样重要，包括数据的干净程度与指令的复杂程度。

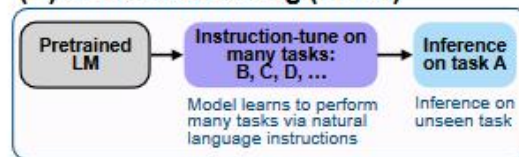
(A) Pretrain-finetune (BERT, T5)



(B) Prompting (GPT-3)



(C) Instruction tuning (FLAN)



```
{  
  "instruction": "我有一个api序列, 告诉我它的ttp技术编号:  behavior-processes-calls:{'filesystem;FindResourceExA;','filesystem;NtOpenFile;','filesystem;CopyFileA;','misc;RtlDosPathN  
'process;NtOpenSection;','synchronization;NtOpenMutant;','filesystem;NtCreateFile;','  
'registry;NtOpenKey;','system;NtClose;','synchronization;NtCreateMutant;','registry;  
'','registry;RegSetValueExA;','system;NtDelayExecution;','filesystem;NtQueryAttribute  
'filesystem;FindFirstFileExW;','misc;GetSystemInfo;','misc;HeapCreate;','filesystem;  
'NtProtectVirtualMemory;','filesystem;NtQueryInformationFile;'};procdump-yara:set();pro  
  "input": "",  
  "output": "对应的ttp编号是: T1082,T1012,T1222,T1547.001"  
},
```

图9、10 三种调优方法对比（上）  
与预训练-微调指令示例（下）

# 训练策略

**对齐调节：**用于与人类的特定偏好相协调。主要有**人类反馈强化学习（RLHF）**和**直接偏好优化（DPO）**。

**RLHF：**利用强化学习算法使LLM与人类偏好保持一致，并以人类注释作为训练循环中的监督。

对预训练模型进行初步微调；

设定首选配对的奖励偏好；

$$\mathcal{L}(\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log(\sigma(r_\theta(x, y_w) - r_\theta(x, y_l)))]$$

在强化学习中，采用**近端策略优化（PPO）**和**KL散度**分别控制更新幅度和策略波动。

$$\begin{aligned} \mathcal{L}(\phi) = & -\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\phi^{RL}(y|x)} \left[ r_\theta(x, y) \right. \\ & \left. - \beta \cdot \mathbb{D}_{KL} \left( \pi_\phi^{RL}(y|x) || \pi^{REF}(y|x) \right) \right] \end{aligned}$$

**DPO：**利用简单的二元分类损失从人类偏好标签中学习。

$$\begin{aligned} \mathcal{L}(\phi) = & -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_\phi^{RL}(y_w|x)}{\pi^{REF}(y_w|x)} \right. \right. \\ & \left. \left. - \beta \log \frac{\pi_\phi^{RL}(y_l|x)}{\pi^{REF}(y_l|x)} \right) \right] \end{aligned}$$



# 模型评估

分为封闭集和开放集。

**闭集评估：**针对答案选项已经预先定义，并且限制在有限集合内的问题。模型需要从给定的选项中选择正确的答案。其基准指标可以是在特定任务或数据集上的准确率，也可以是其他评价指标。通常分为零次调优（zero-shot）和微调（fine-tuning）两种设置。零次调优指模型在已有数据集上训练，处理未见任务；微调在特定数据集上训练，获取更好的性能。

**开集评估：**更加灵活的问题和回答，模型需要以聊天机器人的形式进行对话，对话内容可以是任意的。其标准可以分为人工手动评分、GPT评分和案例研究。

# MLLM研究趋势

**粒度支持：**开发在模型输入和输出方面更细粒度支持的MLLM。在输入方面，研发支持用户提示的更细粒度控制的模型，例如从图像到区域、像素；在输出方面，随着输入支持的发展，模型能够给出更高精度和更细粒度的结果。

**模态支持：**MLLM对模态的支持程度不断增加。研究支持更多种模态内容输入的MLLM，例如3D点云。此外，扩展MLLM以生成更多模态的回答，如图像、音频和视频等。

**语言支持：**一些工作致力于开发非英语的模型，以覆盖更广泛的用户群体。

**场景/任务扩展：**除了开发通用的MLLM外，一些研究着眼于更具体的场景，并将MLLMs扩展到具有特定专业知识的下游任务中。例如扩展到医学领域，通过灌输医学领域的知识来解决医学领域的任务。



# MLLM与多模态幻觉

多模态幻觉是指 MLLM 产生的响应与图像内容不一致的现象。分为：

存在幻觉：最基本的形式，这意味着模型错误地声称图像中某些物体的存在；

属性幻觉：以错误的方式描述某些物体的属性，例如：无法正确识别颜色，通常与存在幻觉相关；

关系幻觉：是一种更复杂的类型，指的是对物体之间关系的错误描述，例如相对位置和相互作用。

评估指标：

CHAIR：检测幻觉事物的比例；

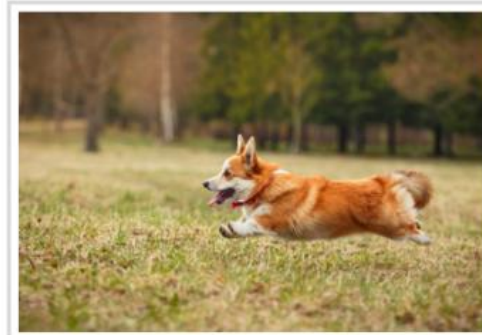
POPE：给出多个二元选择进行测试；

HaELM：利用纯文本模型自动评价文本幻觉。

其他还包括MME、FaithScore、AMBER等评价指标与方法。

8

Please describe this image in detail.



The image features a cute, **red dog** running across a grassy field ... Around the dog, there are **several other dogs** visible in the background ...

图11 MLLM的幻觉现象

# MLLM与多模态幻觉

多模态幻觉的缓解方法：

**事前纠正：**一个直观且直接的解决方案是收集专门的数据（例如负面数据）并使用这些数据进行微调，从而产生具有较少幻觉反应的模型。

**事中纠正：**在架构设计或功能表示方面进行改进，探讨幻觉产生的原因，并设计相应的补救措施来减轻幻觉的产生过程。

**事后纠正：**在输出生成后纠正幻觉，如利用专家模型补充上下文信息，或对于不确定性较高的结果重新生成回答。

# 延伸学习

## 1. 多模态上下文学习 (ICL)

给出少量示例与可选附加指令，LLM观察并对比已有的例子，在类比中学习。

目的是利用少量示例，使模型抽象出通用的理解，以提升处理新任务的能力，加强不同模态信息之间的联系。

通常与指令调优相结合，一般可直接增加在推理阶段。

## 2. 多模态思维链 (CoT)

CoT是“一系列中间推理步骤”。学习CoT能力的方法有微调、少样本学习、零样本学习。微调构建一系列思维链数据集；少样本学习手工制作一些上下文示例；零样本学习提供设计指令，例如“让我们逐步思考”。CoT的核心是分解任务为子任务。

结构上，CoT又可以分为单链推理和树形推理方法，链长度分为自适应和预定义。

# 延伸学习

## 3. LLM辅助视觉推理

### 视觉推理结合LLM的优势：

强大的泛化能力：得益于大规模预训练中获得的丰富的开放世界知识，LLM辅助的系统可以轻松地将泛化到未见过的对象或概念上，展现出显著的零样本、少样本性能。

具备新的功能：借助LLMs的强大推理能力，这些系统能够执行复杂任务。例如给定一张图片，某些系统可以解释图片背后的含义，或解释一张搞笑的meme为什么引人发笑。

更好的交互性和控制能力：与传统的视觉推理模型相比，LLM-based系统能够通过用户友好的界面（例如点击和自然语言查询）进行细致的控制。

### LLM在其中扮演的角色：

作为控制器：LLMs作为中心控制器，将复杂任务分解为更简单的子任务/步骤，并将这些任务分配给适当的工具/模块。


作为决策者：在多层迭代中解决复杂任务，LLM负责总结当前上下文和历史信息，决定当前步骤的信息是否足以回答问题或完成任务。

作为语义精炼器：利用其丰富的语言学和语义知识，LLMs通常被指导整合信息成为连贯流畅的自然语言句子，或根据不同的特定需求生成文本。





# 挑战和展望

- 
1. 增强长上下文多模态信息处理能力；
  2. 提升模型遵循复杂指令的能力；
  3. 优化 M-ICL 和 M-CoT 等技术；
  4. 开发基于MLLM的智能实体代理；
  5. 增强模型安全性。



谢谢收看！