多模态大模型发展历程

➢ 基础篇

以transformer架构及拓展

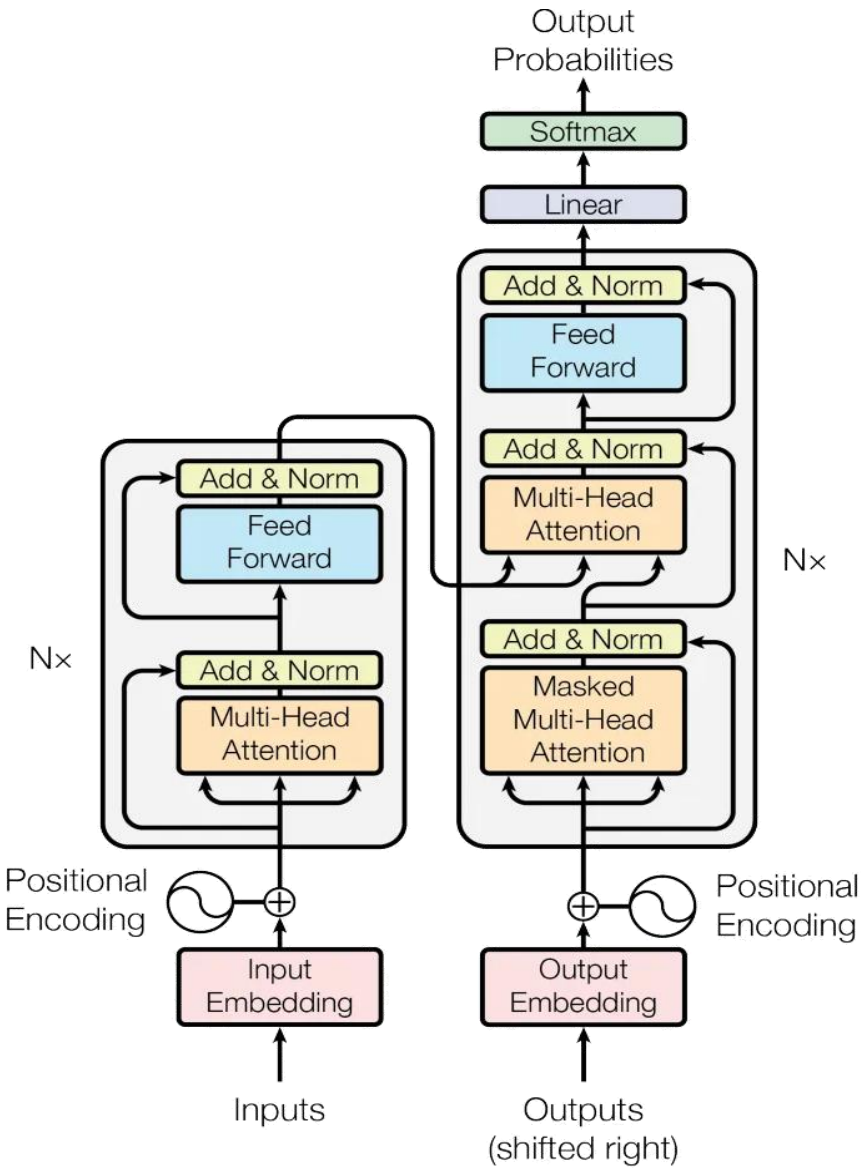
➢ 模型篇

以GPT、CLIP、BLIP、llava、Qwen-VL等系列

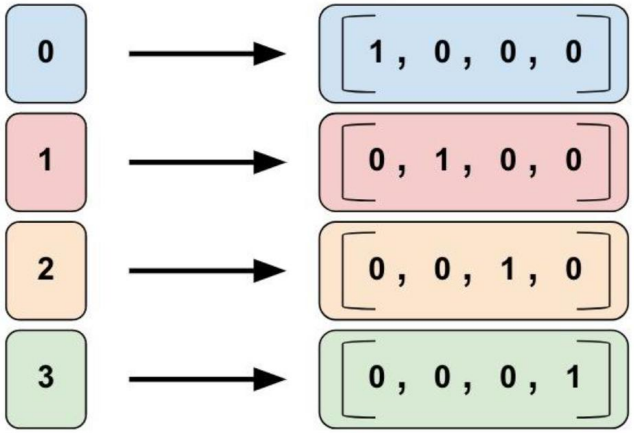
➢ 后训练篇

监督微调，强化对齐

基础篇

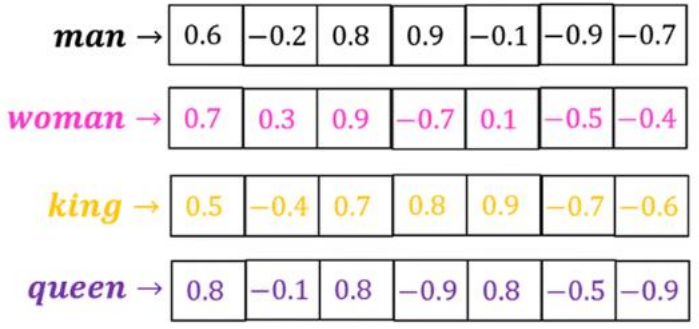


- 输入嵌入层
- Encoder堆叠层（多头自注意力、前馈神经网络）
- Decoder堆叠层（掩码多头注意力、多头交叉注意力、前馈神经网络）
- 输出层（线性层、softmax激活函数）



one-hot编码（独热编码）

有效位只有一位



word embedding

离散特征映射到低维、稠密连续向量空间

位置编码（Positional Encoding）

➢ 作用

是一种将序列中每个元素的**绝对或相对位置信息编码**为向量的技术，其核心目的是**弥补自注意力机制的顺序感知缺陷**。

➢ 例子

句子 A： Dog bit Man （狗咬了人）　　句子 B： Man bit Dog （人咬了狗）

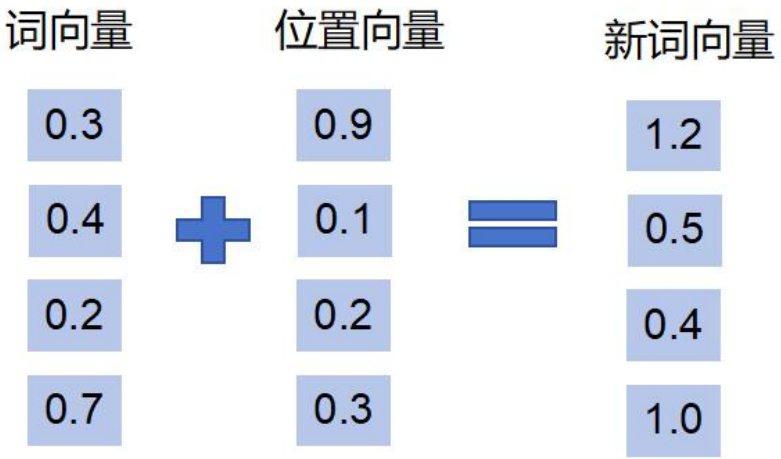
➢ 核心思想

为序列中每一个位置pos分配一个唯一的向量p_i,表示"这是第i个位置"。

$$PE_{(pos,2i)} = sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}})$$

$$PE_{(pos,2i+1)} = cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}})$$

- **pos**：token在序列中的绝对位置（整数，如0，1，2，3，...,L-1）
- **i**：维度组索引，用于生成成对的（sin、cos）值
- $\mathbf{d_{model}}$:模型的嵌入维度（如512、768、4096）



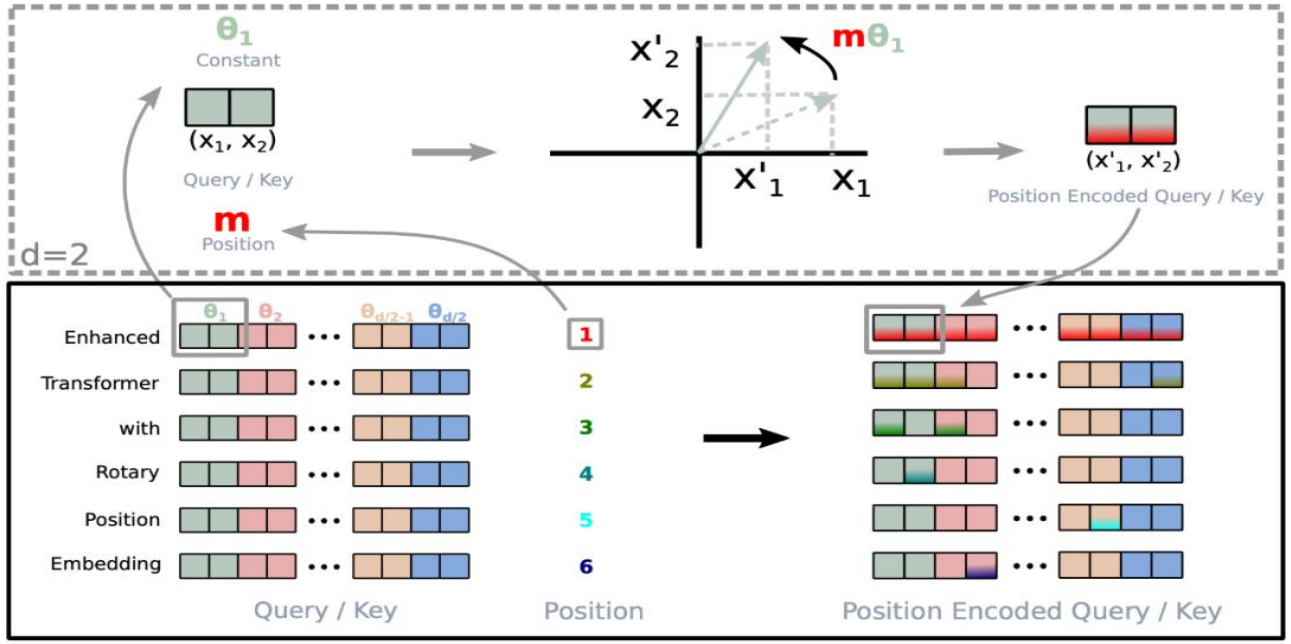
长度外推性差；语义和位置信息混在一起

参考文献：① https://developer.aliyun.com/article/1144579　　② https://blog.csdn.net/v_JULY_v/article/details/134085503

基础篇

旋转位置编码（ROPE)

将位置信息编码为一种旋转变换，并作用于 Query 和 Key 向量上。



RoPE 流程：

1. 输入词 → Embedding → Q, K, V 向量
2. 根据位置 m，生成旋转矩阵 R_m
3. 旋转：Q' = R_m × Q, K' = R_n × K （只在计算时用！）
4. 计算 Attention: Score = Q' · K'^T
5. V 向量不旋转，直接使用

$$[QK^T]_{m,n} = q_m \cdot k_n = q_m^T k_n$$

即第 m 个词的 Query 向量与第 n 个词的 Key 向量的点积。

$$\begin{aligned}\text{Score}(q_m, k_n) &= \langle f(q_m, m), f(k_n, n)\rangle \\ &= \langle R_m q_m, R_n k_n\rangle \\ &= (R_m q_m)^T (R_n k_n) \\ &= q_m^T R_m^T R_n k_n\end{aligned}$$

$$R_m^T R_n = R_m^{-1} R_n = R_{-m} R_n = R_{n-m}$$

$$\text{Score}(q_m, k_n) = q_m^T R_{n-m} k_n$$

结论：内积结果只依赖R_{n-m}，即只与相对位置**n-m有关**，与绝对位置m或n**无关**

$$\boldsymbol{R}^d_{\Theta,m} = \underbrace{\begin{pmatrix} \cos m\theta_0 & -\sin m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_0 & \cos m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_1 & -\sin m\theta_1 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_1 & \cos m\theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2-1} & -\sin m\theta_{d/2-1} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2-1} & \cos m\theta_{d/2-1} \end{pmatrix}}_{W_m}$$

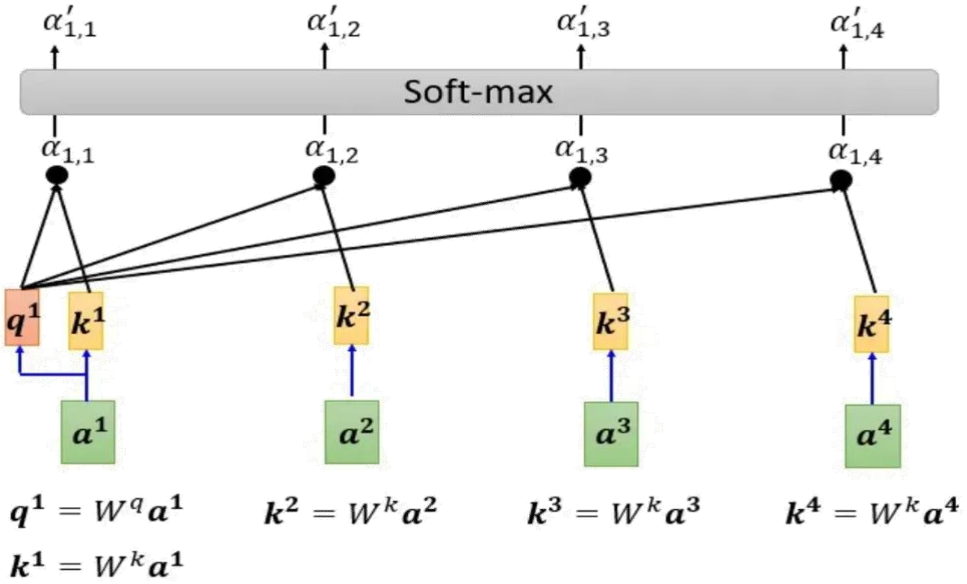
$$\Theta = \left\{\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, \ldots, d/2]\right\}$$

参考文献：① https://spaces.ac.cn/archives/8265 ② https://zhuanlan.zhihu.com/p/2023493768003724514

（多头）自注意力机制

Self-attention

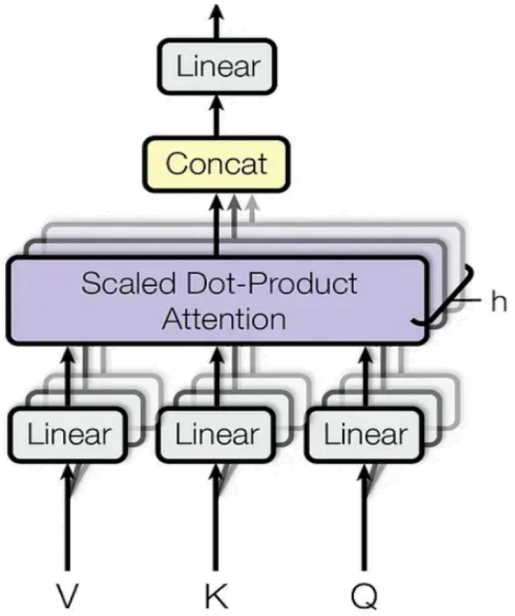
$$\alpha'_{1,i} = exp(\alpha_{1,i}) / \sum_j exp(\alpha_{1,j})$$



$q^1 = W^q a^1$ $k^2 = W^k a^2$ $k^3 = W^k a^3$ $k^4 = W^k a^4$

$k^1 = W^k a^1$

$$Attention(Q_h, K_h, V_h) = softmax(\frac{Q_h K_H^T}{\sqrt{d_k}}) V_h$$



突破单头注意力的表达瓶颈

- 设计动机

将**输入映射到多个子空间并行计算注意力**，使模型能同时关注序列中不同位置的多维度特征。

仅 Q、K、V 的线性层参数量就达 **3*hidden_size²**，且推理时需缓存所有头的K,V，导致**KV缓存占用过高**

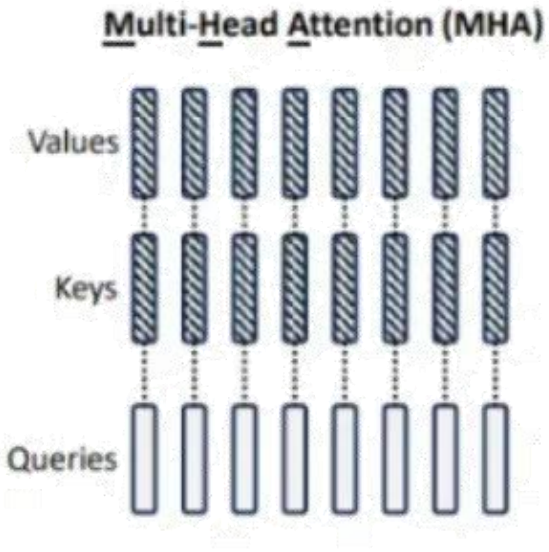
hidden_size：模型中每个 token 的向量表示的维度

基础篇

多头注意力机制改进

多头注意力（MHA）

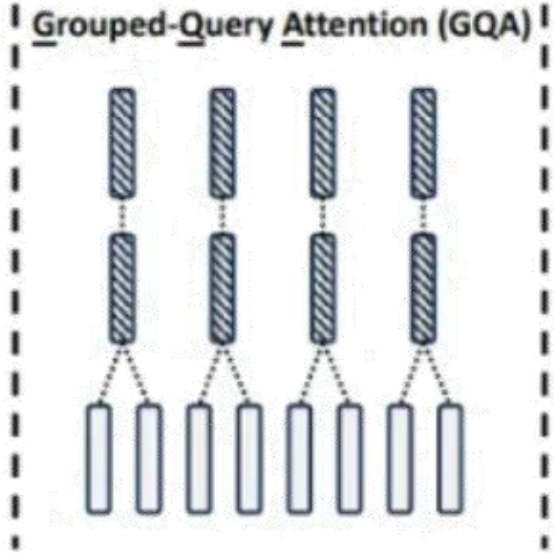
突破单头注意力的表达瓶颈



参数规模：$hidden\_size^2$

分组查询注意力（GQA）

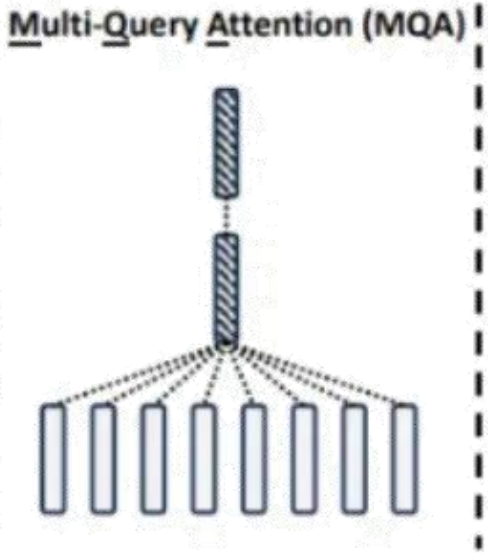
平衡表达与效率的中间方案



性能依赖分组大小 group_size 的选择，过小则接近 MHA（效率低），过大则接近 MQA（表达弱）

多查询注意力（MQA）

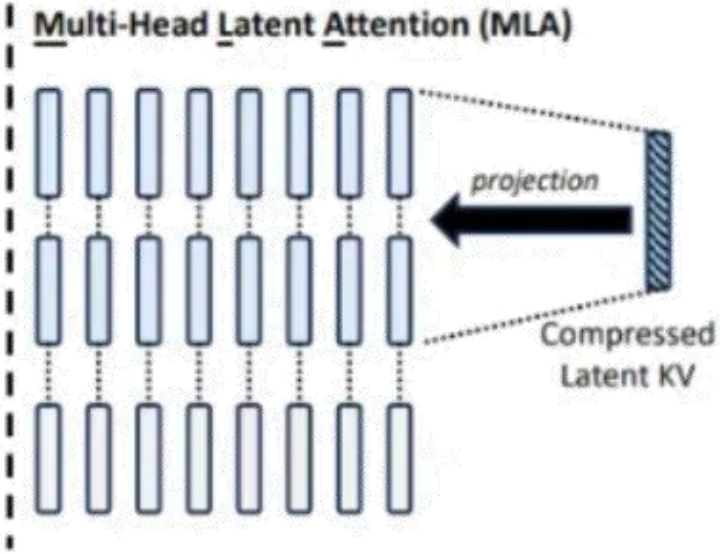
破解 KV 缓存的内存瓶颈



参数规模：
2*hidden_size*head_dim

多头潜在注意力（MLA）

低秩分解与位置编码的协同优化



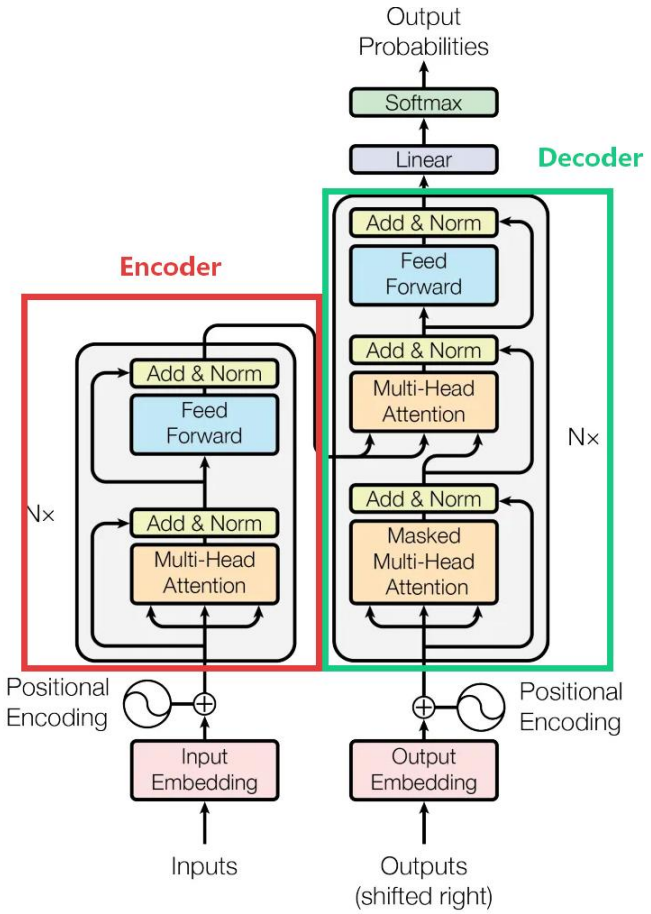
1、低秩联合压缩
2、权重吸收
3、解耦ROPE

head_dim：每个注意力头（Head）内部用于计算的向量维度

参考文献：① https://www.cnblogs.com/rossiXYZ/p/18827618#

基础篇

Encoder和Decoder



模块	输入	输出	核心任务
Encoder	输入序列（如句子、图像 patch）	上下文感知的表征向量（Context Vectors）	**理解输入：**提取语义信息、建模序列内部依赖关系
Decoder	已生成的部分输出（如已翻译的句子）	下一个 token 的概率分布	**生成输出：**自回归地预测序列中的下一个词或token

➢ **注意力机制不同（核心）**

- Encoder 的注意力： 是 **Self-Attention（自注意力）**。
  无 Mask：每个 token 可以看到序列中所有其他 token（双向），目的是充分理解上下文。

- Decoder 的注意力： 分为两层。
  第一层（Masked Self-Attention）： **有 Mask（掩码）**。每个 token 只能看到它左边（已生成）的 token，防止偷看未来，保证自回归生成的合理性。

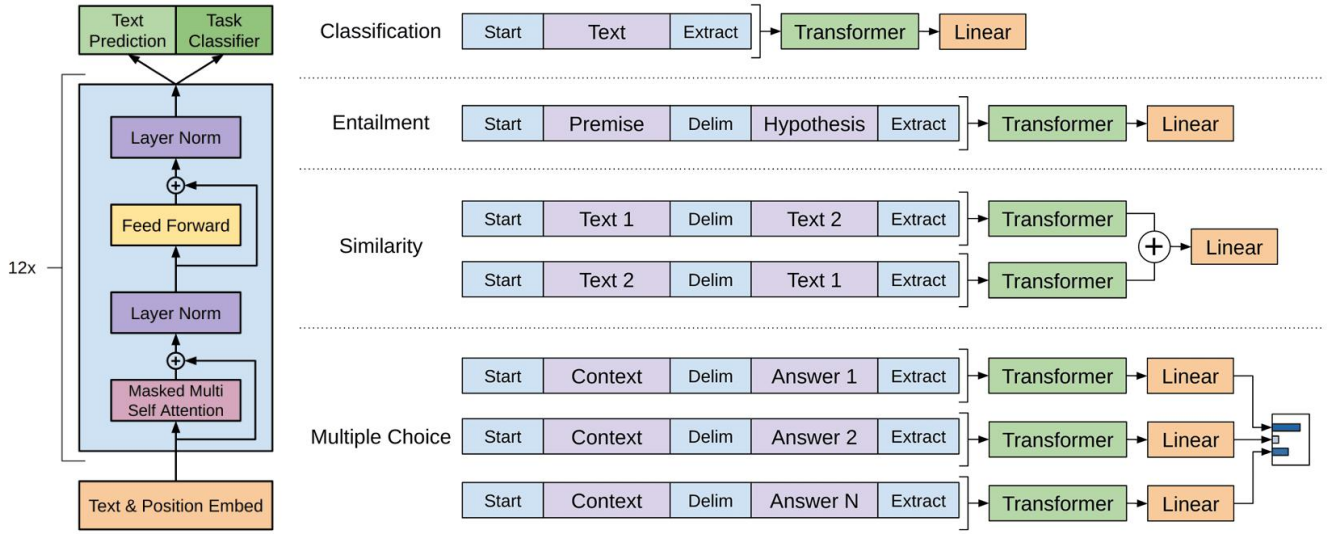
  第二层（Cross-Attention）： **接收 Encoder 的输出。**Query 来自 Decoder 自身，Key 和 Value 来自 Encoder。这让 Decoder 在生成时能“回头看”源序列的信息（比如翻译时对齐原文）。

参考文献：① https://zhuanlan.zhihu.com/p/1967860454354301874 ② https://zhuanlan.zhihu.com/p/1977120174495798604

模型篇

GPT-1训练范式：预训练+Fine-Tuning（2018.6）

不通用：不同损失函数在各个任务上表现差异大，训练数据集并没有包含所有任务。
不统一：预训练模型迁移到下游任务的方法不统一，不同的子任务需调整模型结构。

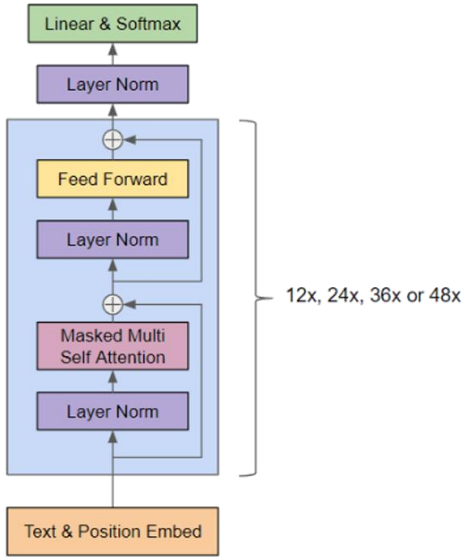


二阶段法
①**无监督预训练**：海量无规则纯文本，让模型自学
②**有监督微调**：特定下游任务的标注数据，交叉熵监督微调

参数量：1.17亿　　预训练数据量：5GB

GPT-2训练范式：预训练+零样本（2019.2）

反超BERT，构建更大的数据集和模型，在多任务场景更好



"I'm not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I'm not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: "**Mentez mentez, il en restera toujours quelque chose**," which translates as, "**Lie lie and something will always remain.**"

"I hate the word '**perfume**,'" Burr says. 'It's somewhat better in French: '**parfum**.'

If listened carefully at 29:55, a conversation can be heard between two guys in French: "**-Comment on fait pour aller de l'autre coté? -Quel autre coté?**", which means "**- How do you get to the other side? - What side?**".

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

"**Brevet Sans Garantie Du Gouvernement**", translated to English: "**Patented without government warranty**".

亮点
①构建WebText，百万级别的文本数据集
②零样本多任务学习

参数量：15亿　　预训练数据量：40GB

参考文献：　① https://gwern.net/doc/www/s3-us-west-2.amazonaws.com/d73fdc5ffa8627bce44dcda2fc012da638ffb158.pdf
② https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf

模型篇

GPT-3训练范式：预训练+少样本（2020.5）

解决GPT-2零样本有效性不足

➢ 更大的数据集

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

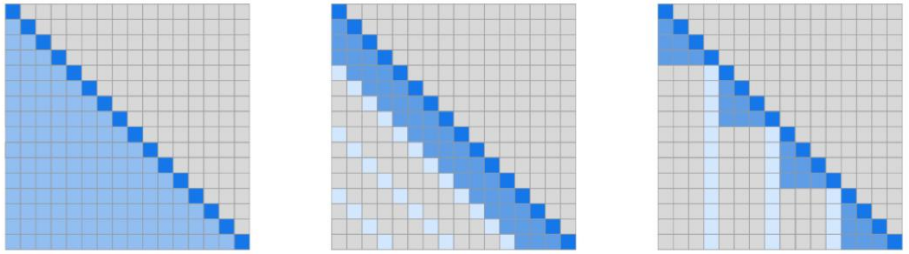
①**自动过滤**：从海量文档筛选高质量文本；Pareto分布根据分数筛选。
②**模糊去重**：使用 Spark MinHashLSH（配置哈希函数）计算文档相似度。
参数量：1705亿　　预训练数据量：45TB

➢ 更大的模型规模

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

➢ 模型架构

稠密和稀疏交替



➢ 上下文学习

```
Poor English input:  I eated the purple berries.
Good English output:  I ate the purple berries.
Poor English input:  Thank you for picking me as your designer.  I'd appreciate it.
Good English output:  Thank you for choosing me as your designer.  I appreciate it.
Poor English input:  The mentioned changes have done.  or I did the alteration that you
requested.  or I changed things you wanted and did the modifications.
Good English output:  The requested changes have been made.  or I made the alteration that you
requested.  or I changed things you wanted and made the modifications.
Poor English input:  I'd be more than happy to work with you in another project.
Good English output:  I'd be more than happy to work with you on another project.

Poor English input:  Please provide me with a short brief of the design you're looking for and
that'd be nice if you could share some examples or project you did before.
Good English output:  Please provide me with a brief description of the design you're
looking for and that would be nice if you could share some examples or projects you have
done before.
```

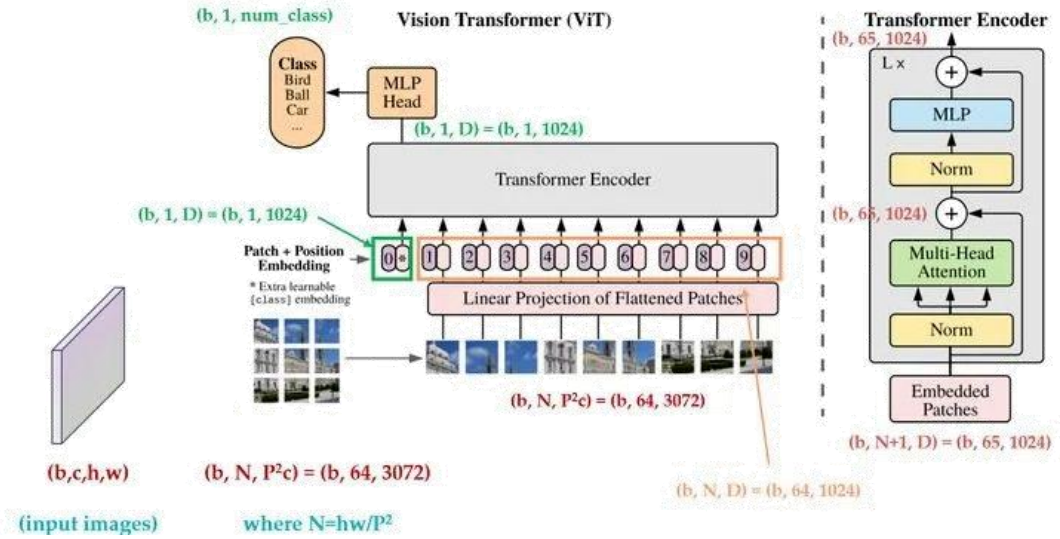
范式	定义	Prompt 结构示例	本质
Zero-shot	不提供任何任务示例，仅用自然语言描述任务	`Translate to French: Hello →`	**指令驱动**：靠预训练先验直接泛化
One-shot	提供**1 个**输入 - 输出示例作为参考	`English: Hi → French: Salut\nEnglish: Bye →`	**模式模仿**：单示例激活任务格式认知
Few-shot	提供 **K 个（通常 2~10）** 示例作为上下文	`Ex1: A→B\nEx2: C→D\nEx3: E→`	**上下文学习（In-Context Learning）**：示例构建任务流形

💡 **关键澄清**：这三种范式**都不涉及梯度更新**（即不微调模型参数），区别仅在于**推理时输入的构造方式**。

参考文献：① https://arxiv.org/pdf/2005.14165 ② https://arxiv.org/pdf/1904.10509（sparse transformer）　③ https://zhuanlan.zhihu.com/p/18104675680

模型篇

ViT（Vision Transformer）2021



1. 输入图像 (input images) 的 shape = (b = b, c = 3, h = 256, w = 256)。
2. 输入图像 (input images) 被切分 (Split / Divide) 并展平 (Flatten) 为：batch size 仍为 b，通道数 c = **3**、尺寸 P = **32**、个数 N = (256×256) / (32×32) = **64** 的图像块 (Patch)，每个图像块 (Patch) 均有 P^2c = 32×32×3 = **3072** 个像素。
3. 图像块 (Patch) 馈入线性投影层 (Linear Projection)，得到个数/长度 (length) 为 N = **64**、像素数/大小/维度 (dimension) 为 D = (32×32×1) = **1024** 的图像块嵌入 (Patch Embedding)。
4. 每个图像块嵌入 (Patch Embedding) 按元素加 (Element-wise Summary) 入位置向量/嵌入后，尺寸仍为 N×D = **64×1024**。
5. 具有位置嵌入的图像块嵌入 (Patch Embedding) 再于长度 (length) 维度 拼接 (Concat) 一个用于预测分类结果的 **1×1024** 可学习嵌入/向量，构成大小为 **65×1024** 完整嵌入 (长度 (length) N+1 = 64+1 = **65**)。
6. 完整嵌入输入编码器经过一系列前向处理后，得到尺寸仍为 N×D = **65×1024** 的输出。

切图成块 → 投影成向量 → 加位置+CLS → 自注意力全局交互 → 提取CLS向量 → 全连接分类

➢ 效果展示

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

参考文献：① https://arxiv.org/pdf/2010.11929 ② https://www.zhihu.com/search?type=content&q=VIT

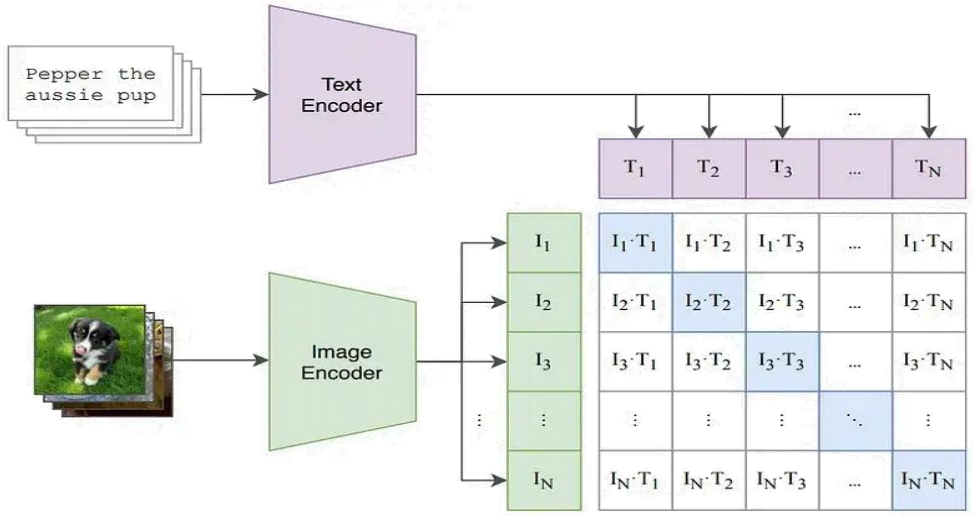
模型篇

CLIP 2021

➢ 动机

- **(image, text) 配对学习**：能学视觉表征，但数据稀缺
- **弱监督大数据**：数据量大（18K 类），但类别固定，无法 zero-shot 泛化到新概念

基于**ConVIRT**进行改进，**四亿条图文对**



信息噪声对比估计损失（Info NCE loss）

$$\hat{y}_+ = softmax(z+) = \frac{exp(z+)}{\sum_{i=0}^{k} exp(z_i)} \Rightarrow -log\frac{exp(z+)}{\sum_{i=0}^{k} exp(z_i)}$$

交叉熵损失函数 $L(\hat{y}) = -\sum_{i \in K} y_i log(\hat{y}_i)$

对于一组样本 x_i，假设每个样本 x_i 有一个正样本对 x_j（通常为同一语义的不同视图，如图片的不同增强版本），以及 K 个负样本 $x_k{}_{k=1}^{K}$，InfoNCE Loss 的公式为：

$$\mathcal{L}_{InfoNCE} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(sim(x_i, x_j^+)/\tau)}{\exp(sim(x_i, x_j^+)/\tau) + \sum_{k=1}^{K}\exp(sim(x_i, x_k^-)/\tau)}$$

其中：

2. N 是样本数量，$sim(\cdot)$ 是相似度函数（如余弦相似度或点积）。
3. τ 是温度参数（Temperature），用于调节概率分布的平滑程度：τ 越小，相似度差异被放大，分类边界更严格；τ 越大，分布越平滑。

```python
# ========== 准备图像 ==========
image = preprocess(Image.open("dog.jpg")).unsqueeze(0).to(device)

# ========== 定义分类标签 ==========
classes = [
    "a photo of a dog",
    "a photo of a cat",
    "a photo of a bird",
    "a photo of a car",
    "a photo of a person"
]
```

```
分类结果:
a photo of a dog: 0.8523
a photo of a cat: 0.0821
a photo of a bird: 0.0312
a photo of a car: 0.0201
a photo of a person: 0.0143
```

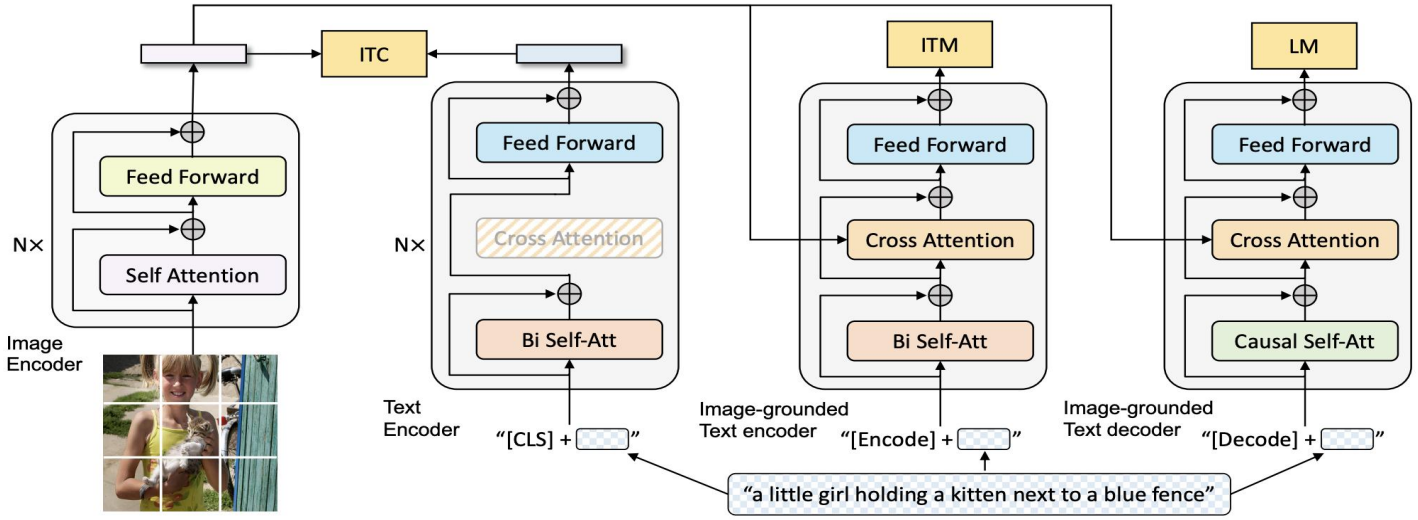
```python
def contrastive_loss(logits_per_text, logits_per_image):
    batch_size = logits_per_text.shape[0]
    labels = torch.arange(batch_size, device=logits_per_text.device)
    loss_text = nn.CrossEntropyLoss()(logits_per_text, labels)
    loss_image = nn.CrossEntropyLoss()(logits_per_image, labels)
    return (loss_text + loss_image) / 2
```

参考文献：① https://zhuanlan.zhihu.com/p/506544456 ② https://arxiv.org/pdf/2103.00020

模型篇

BLIP 2022.1

- **理解类模型**（如CLIP、ALBEF）采用纯encoder架构，擅长图文匹配、视觉问答（VQA）等理解任务，但难以直接用于文本生成
- **生成类模型**（如Oscar）采用encoder-decoder架构，适合图像描述生成，但在理解任务上表现受限



- **image encoder:** 图像编码器，用于提取视觉特征，初始化为在ImageNet上预训练的ViT，该模型需要与其他3个模型联合训练，没有单独的训练目标。
- **text encoder:** 文本编码器，用于提取文本特征，初始化为Bert。该模型与image encoder联合训练的目标为ITC（Image-Text Contrastive Loss），ITC将图像和文本映射到相同的特征空间（与CLIP类似）。
- **image-grounded text encoder:** 图像文本编码器，用于判断文本和图像是否匹配。该模型的输入为[Encode] +文本，另外每个block层都以交叉注意力的形式关注image encoder对应层的视觉特征。训练目标为ITM(Image-Text Matching Loss)，即对当前文本和图像是否匹配进行二分类。
- **image-grounded text decoder:** 图像文本解码器，用于生成图像的caption。交叉注意力机制的引入与image-grounded text encoder类似，训练目标为LM 损失(Language Modeling Loss)。

- **ITC 损失作用：**将整张图和整句话映射到同一特征空间。计算余弦相似度，拉近匹配对，推远不匹配对。
- **ITM 损失作用：**二分类任务（匹配/不匹配）。通过交叉注意力进行词-区域级对齐，判断图文是否真正对应。
- **LM 损失作用：**自回归语言建模（预测下一个 token）。迫使模型在生成每个词时，必须参考图像视觉线索。

参考文献：①https://www.cnblogs.com/JuggyZhan/p/18863169 ② https://arxiv.org/pdf/2201.12086

模型篇

BLIP 2022.1

➢ 数据集自推荐（Dataset Bootstrapping ）

网络数据虽量大但充斥**图文不匹配**的噪声，**高质量标注稀缺**，导致仅靠网络数据训练仅为次优方案

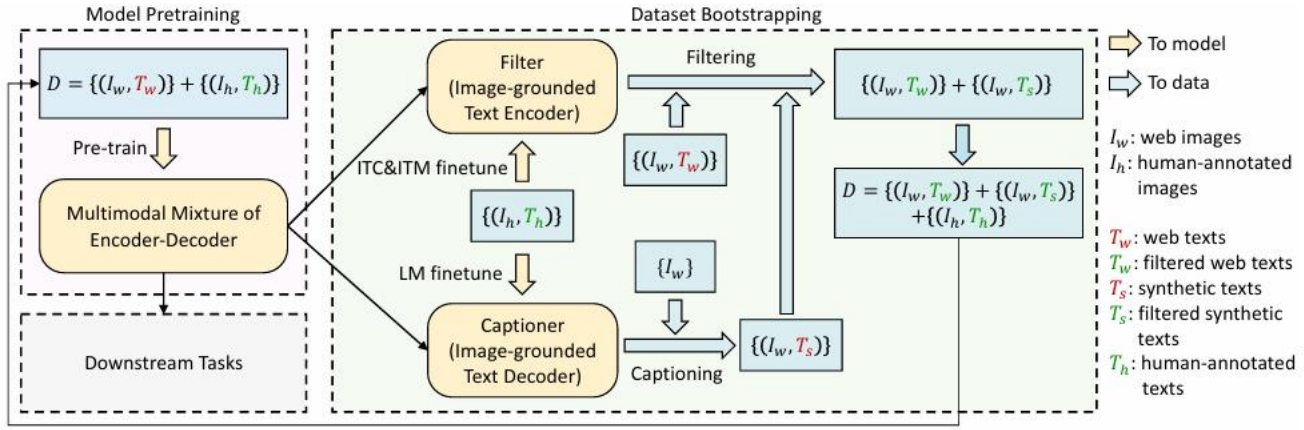


Figure 3. Learning framework of BLIP. We introduce a captioner to produce synthetic captions for web images, and a filter to remove noisy image-text pairs. The captioner and filter are initialized from the same pre-trained model and finetuned individually on a small-scale human-annotated dataset. The bootstrapped dataset is used to pre-train a new model.

字幕器 Captioner：给一张网络图片，生成字幕。它是一个视觉文本解码器，在 COCO 数据集上使用 LM 目标函数微调。给定网络图片 I_w ，Captioner 生成字幕 T_s 。

过滤器 Filter：过滤掉噪声图文对。它是一个视觉文本编码器，看文本是否与图像匹配，在 COCO 数据集上使用 ITC 和 ITM 目标函数微调。Filter 删除原始 Web 文本 T_w 和合成文本 T_s 中的嘈杂文本，如果 ITM 头将其预测为与图像不匹配，则认为文本有噪声。

➢ 效果展示

Figure 4. Examples of the web text T_w and the synthetic text T_s. Green texts are accepted by the filter, whereas red texts are rejected.

参考文献：① https://www.cnblogs.com/JuggyZhan/p/18863169 ② https://arxiv.org/pdf/2201.12086

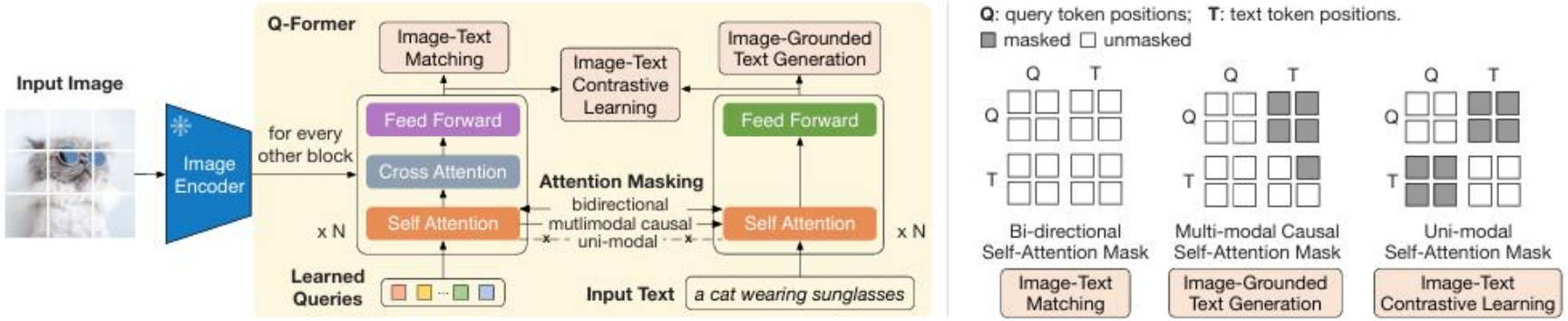
模型篇

BLIP-2 2023.1

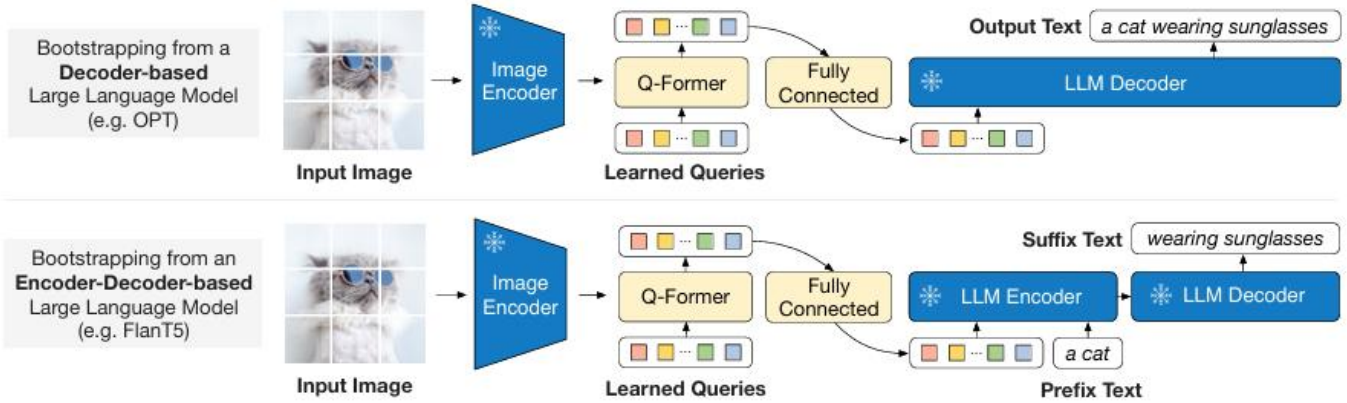
➢ 背景

ChatGPT 证明了 LLM 的通用性，Flamingo 以 LLM 为核心引入交叉注意力弥合图文鸿沟；

表征学习（Representation Learning）：学习Q-Former以提取与文本相关的图像特征；



生成学习（Generative learning）：将Q-Former提取的特征对齐到文本模态



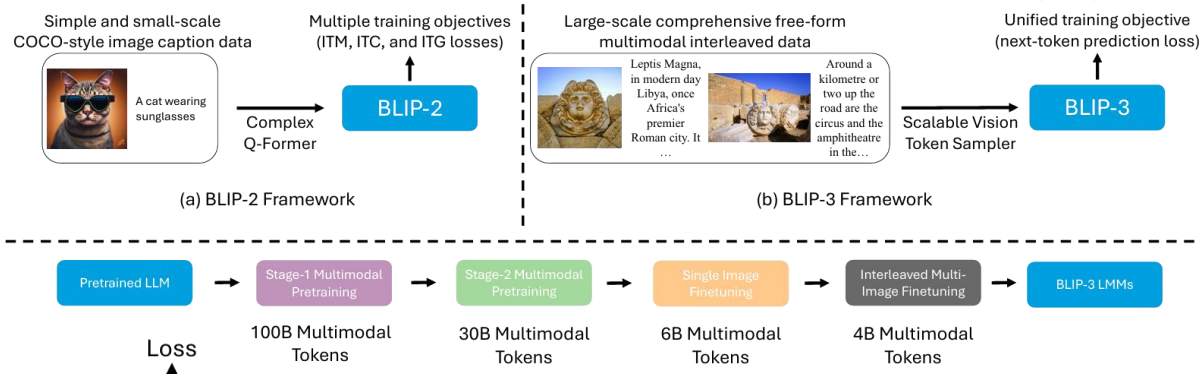
参考文献：① https://arxiv.org/pdf/2301.12597

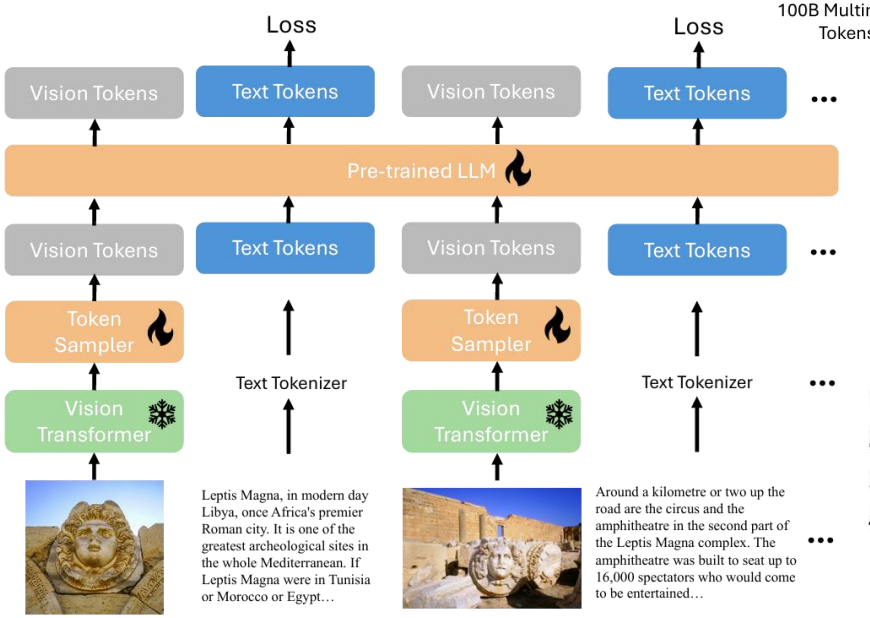
模型篇

BLIP-3 2024.8

数据规模、质量和多样性不足，难以达到现代 LMM 的竞争力
- 架构复杂（Q-Former + 三损失函数），难以大规模扩展
- 仅支持单图像输入，无法处理更自然的图文交错格式

- **数据驱动：** 整合 MINT-1T、BLIP3-KALE等高质量数据集
- **架构简化：** 用可扩展的 Vision Token Sampler 替代 Q-Former
- **目标统一：** 仅保留自回归文本损失，移除 ITC/ITM/ITG 多任务
- **格式升级：** 原生支持图文交错的多图像输入





- vision transformer：SigLIP（不训练）
- token sampler（训练）
- LLM：phi3-mini（训练）

1、token采样器

1. **输入**：ViT 编码后的图像 Patch 特征序列（如 N 个 Token）
2. **可学习 Query**：初始化一组固定数量的查询向量（论文中每 Patch 用 **128 个 Query**）
3. **交叉注意力**：Query 作为 Query，图像 Token 作为 Key/Value，进行 Cross-Attention
4. **输出**：128 个高度浓缩的视觉表示向量，直接送入 LLM

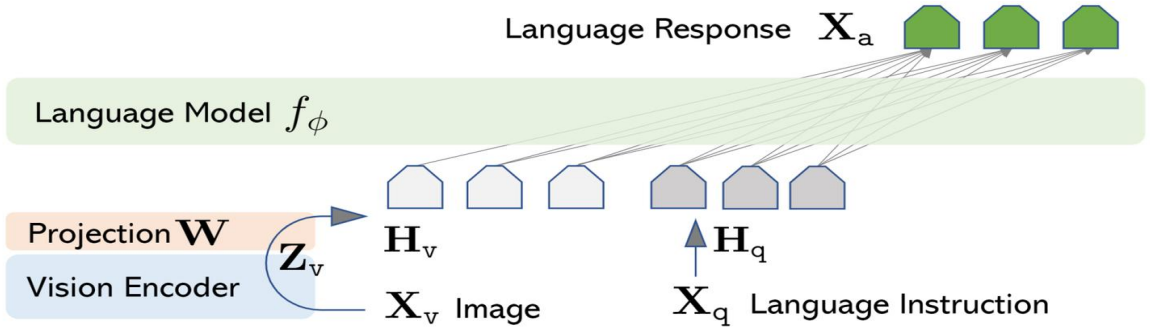
2、Any-Resolution Vision Token Sampling（任意分辨率视觉 Token 采样）

1. **动态切分**：
   - 限制图像长边最大为 **1536 像素**
   - 将图像切分为多个 **384×384** 的 Patch（最多 **12 个**）
   - 保留一个缩小的全局原图 Patch 提供上下文
2. **独立编码**：每个 Patch 单独输入冻结的 ViT，得到独立特征
3. **Patch-wise 独立降采样**：
   - **关键设计**：不拼接所有 Patch 再统一降采样，而是**对每个 Patch 单独执行 Perceiver Resampler**
   - 将各 Patch 的 128 个输出向量拼接，送入 LLM

参考文献：① https://arxiv.org/pdf/2408.08872 ② https://zhuanlan.zhihu.com/p/28392731664

模型篇

LLaVA 2023.4



文本编码器： Vicuna　　视觉编码器：CLIP预训练的ViT-/14

➢ 构造数据

- **Conversation**：对话数据，共 58K 个样本
- **Detailed description**：对图像丰富而全面的描述，共 23K 个样
- **Complex reasoning**：复杂推理数据，数据的回复通常需要遵循严格的逻辑逐步推理，共 77K 个样本

Context type 1: Captions
A group of people standing outside of a black vehicle with various luggage.
Luggage surrounds a vehicle in an underground parking area
People try to fit all of their luggage in an SUV.
The sport utility vehicle is parked in the public garage, being packed for a trip
Some people with luggage near a van that is transporting it.

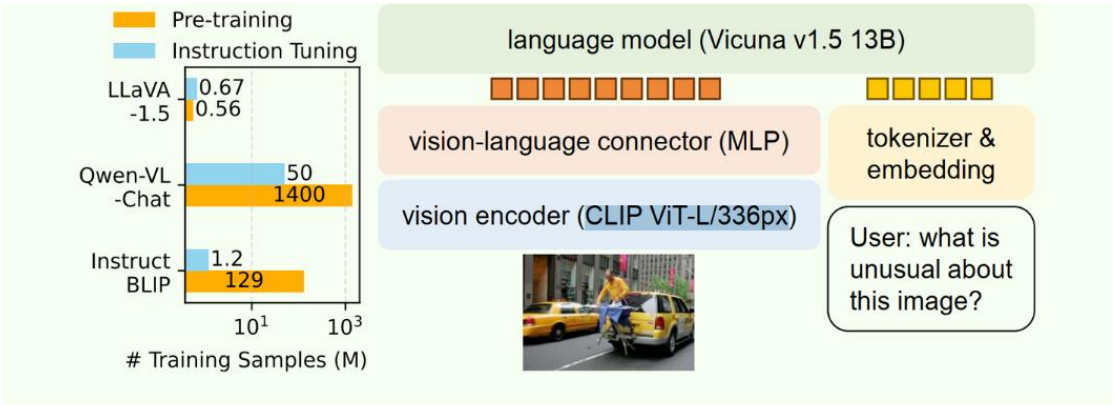
Context type 2: Boxes
person: [0.681, 0.242, 0.774, 0.694], backpack: [0.384, 0.696, 0.485, 0.914], suitcase: ...<omitted>

Response type 1: conversation
Question: What type of vehicle is featured in the image?
Answer: The image features a black sport utility vehicle (SUV) ...<omitted>

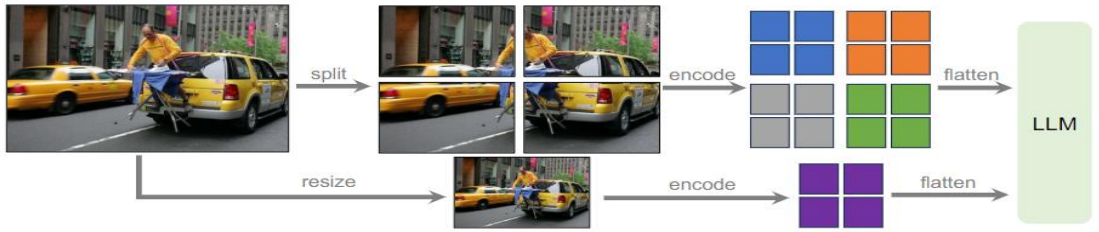
Response type 2: detailed description
The image is an underground parking area with a black sport utility vehicle (SUV) parked. There are three people in the scene, with one person standing closer to the left side of the vehicle, another person in the middle, and the third person on the right side. They are all working together to pack their luggage into the SUV for a trip. ...<omitted>

Response type 3: complex reasoning
Question: What challenges do these people face?
Answer: In the image, a group of people is standing outside a black SUV in a parking area, surrounded by various pieces of luggage, including suitcases and backpacks. They are facing the challenge of fitting all their luggage into the black SUV. There are multiple suitcases and backpacks to be packed, which suggests that the group has a significant amount of belongings ...<omitted>

LLaVA 1.5　2023.10



LLaVA 1.5-HD

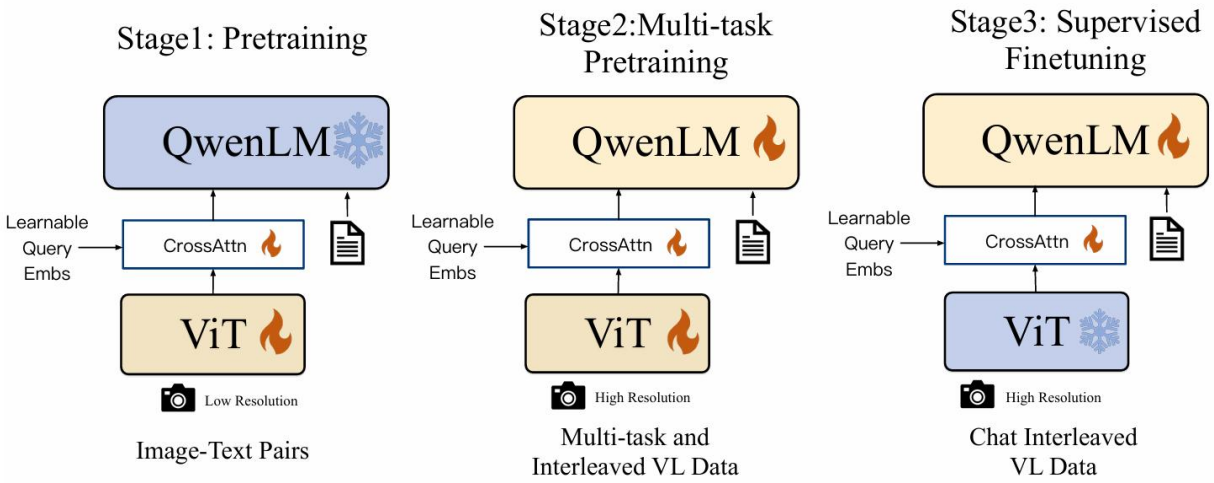


- 将高分辨率的图像分割成块，块的大小取决于视觉编码器能够处理的大小（例如 CLIP-ViT-L/14 可以处理的分辨率为 224*224）。视觉编码器单独处理每一块。
- 同时，将高分辨率的图像 resize 成视觉编码器能够处理的大小并使用视觉编码器进行编码
- 将上面两步的结果拼接在一起作为视觉特征

参考文献：① https://arxiv.org/pdf/2304.08485　② https://arxiv.org/pdf/2310.03744　③ https://zhuanlan.zhihu.com/p/696402890

模型篇

Qwen-VL 2023.8



➢ Stage1 预训练

目标：利用大规模、弱标注的**图像-文本对**数据训练模型，优化视觉编码器和视觉-语言适配器，同时冻结大型语言模型。（分辨率为224*224）

Language	Dataset	Original	Cleaned	Remaining%
English	LAION-en	2B	280M	14%
	LAION-COCO	600M	300M	50%
	DataComp	1.4B	300M	21%
	Coyo	700M	200M	28%
	CC12M	12M	8M	66%
	CC3M	3M	3M	100%
	SBU	1M	0.8M	80%
	COCO Caption	0.6M	0.6M	100%
Chinese	LAION-zh	108M	105M	97%
	In-house Data	220M	220M	100%
	Total	5B	1.4B	28%

➢ Stage2 多任务预训练

目标：引入高质量、细粒度标注数据，提升模型的多任务能力，解锁并训练整体模型。（分辨率为448*448）

Image Captioning
<img>cc3m/01581435.jpg</img>Generate the caption in English: the beautiful flowers for design.<eos>
Vision Question Answering
<img>VG_100K_2/1.jpg</img> Does the bandage have a different color than the wrist band? Answer: No, both the bandage and the wrist band are white.<eos>
OCR VQA
<img>ocr_vqa/1.jpg</img> What is the title of this book? Answer: Asi Se Dice!, Volume 2: Work-book And Audio Activities (Glencoe Spanish) (Spanish Edition)<eos>
Caption with Grounding
<img>coyo700m/1.jpg</img>Generate the caption in English with grounding: Beautiful shot of <ref>bees</ref><box>(661,612),(833,812)</box><box>(120,555),(265,770) </box> gathering nectars from <ref>an apricot flower</ref><box>(224,13),(399,313) </box><eos>
Referring Grounding
<img>VG_100K_2/3.jpg</img><ref>the ear on a giraffe</ref><box>(176,106),(232,160) </box><eos>
Grounded Captioning
<img>VG_100K_2/4.jpg</img><ref>This</ref><box>(360,542),(476,705)</box> is Yellow cross country ski racing gloves<eos>
OCR
<img>synthdog/1.jpg</img>OCR with grounding: <ref>It is managed</ref> <quad> (568,121), (625,131), (624,182), (567,172)</quad>...<eos>

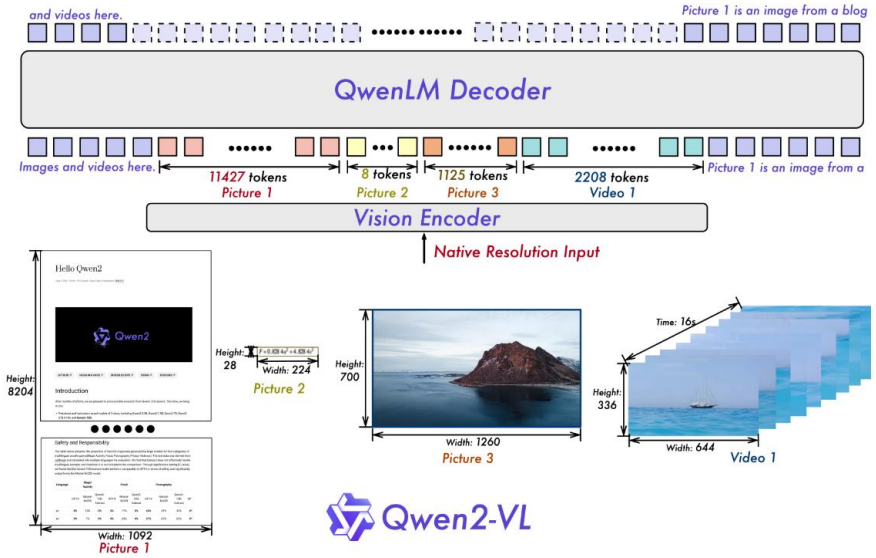
➢ Stage3 监督微调

目标：通过指令微调提升 Qwen-VL 的指令遵循和对话能力，生成 Qwen-VL-Chat。

参考文献：① https://arxiv.org/pdf/2308.12966

模型篇

Qwen2-VL 2024.9 奠基代

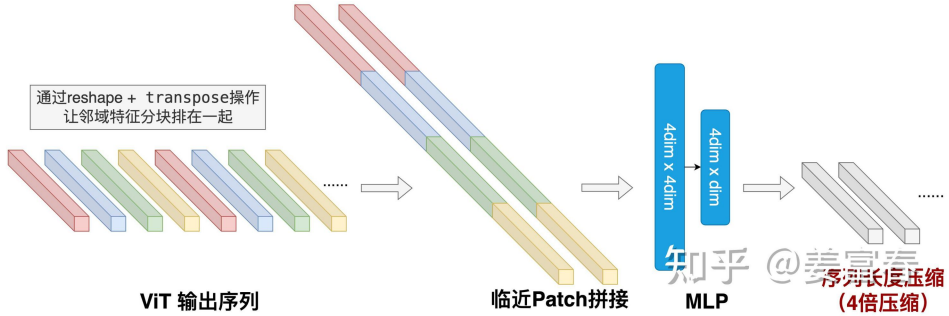


➢ 1、原生动态分辨率（Naive Dynamic Resolution）

生成变长Patch序列，边缘零填充保证张量对齐

➢ 2、ViT引入2D-RoPE 位置编码

➢ 3、输入投影层：压缩Vision token + MLP Adapter



➢ 4、多模态旋转位置嵌入（M-RoPE）

➢ 5、统一的图像和视频理解框架

视频处理：2fps采样保留时序信息，动态调整单帧分辨率并约束总Token≤16K，平衡长视频细节与计算效率。

图像处理：单图复制为2帧序列，统一视觉输入范式，复用视频时序编码路径，简化架构设计与训练流程。

参考文献：① https://arxiv.org/pdf/2409.12191 ② https://zhuanlan.zhihu.com/p/25267823390

模型篇

Qwen2-VL

➢ 对话数据

The Dataset Format Example of ChatML

<|im_start|>user
<|vision_start|>Picture1.jpg<|vision_end|><|vision_start|>Picture2.jpg<|vision_end|>What do the two pictures have in common?<|im_end|>
<|im_start|>assistant
Both pictures are of SpongeBob SquarePants. <|im_end|>
<|im_start|>user
What is happening in the video?<|vision_start|>video.mp4<|vision_end|><|im_end|>
<|im_start|>assistant
The protagonist in the video is frying an egg.<|im_end|>

➢ 定位

Referring Grounding

<|vision_start|>Picture1.jpg<|vision_end|>
<|object_ref_start|>the eyes on a giraffe<|object_ref_end|><|box_start|>(176,106),(232,160)<|box_end|>

➢ Agent格式

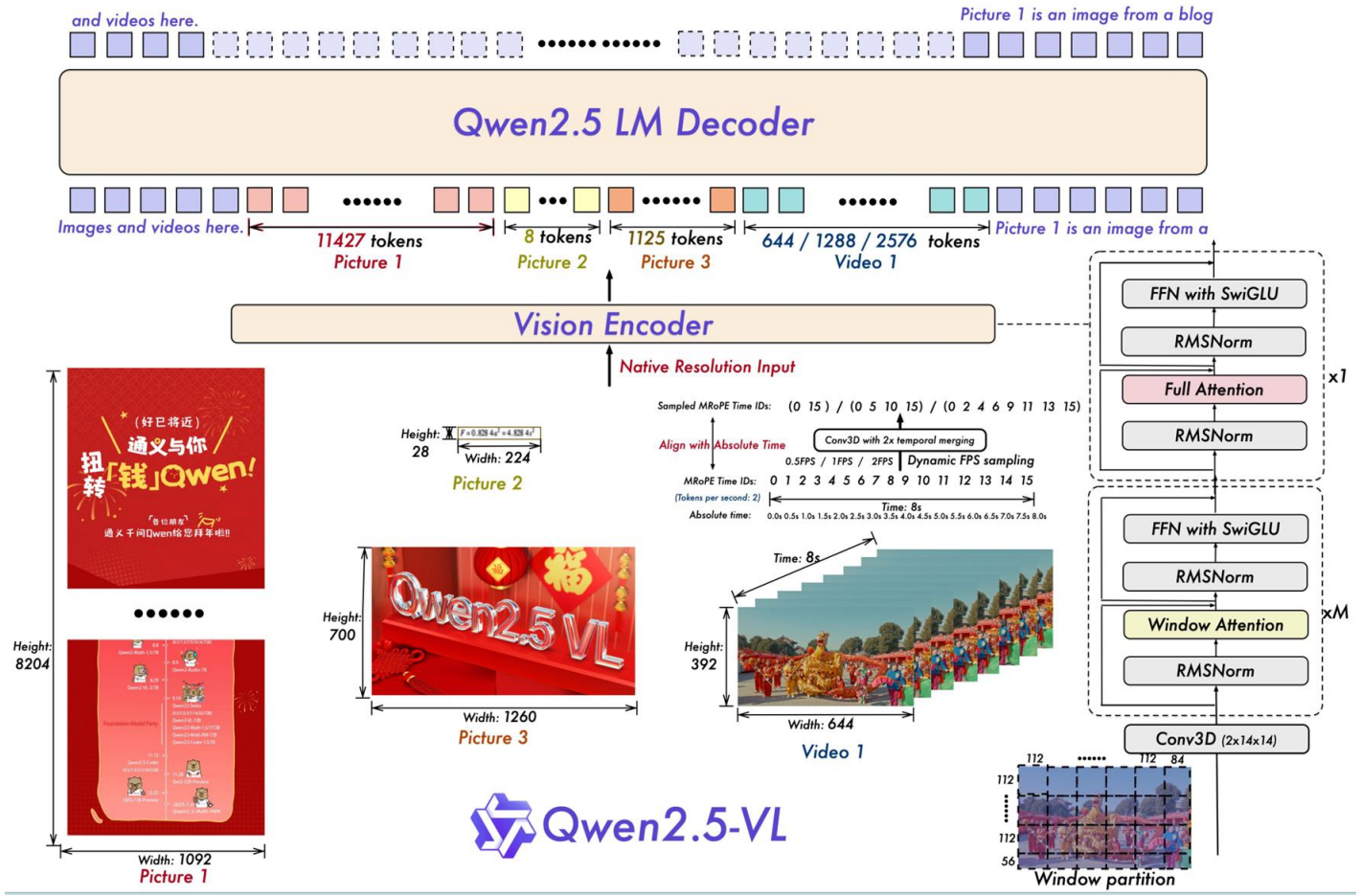
Visual Agent

<|im_start|>system
You are a helpful assistant.
Actions
You have the following actions.
Tap
Tap: A gentle tap that commands, chooses, or navigates through a smartphone's user interface.
Parameters: [{"name": "point", "description": "The specific spot of interest on the monitor, denoted by the coordinates (x, y) where x and y range from 0 to 1000.", "required": True}]
Home
Home: Go to phone's home screen. Parameters: []
Other Actions ...
Continuously take action until the task is completed.
FUNCTION: The action to take, should be one of {Actions}.
ARGS: The input of the action.
RESULT: Action results.
RETURN: Reply based on action results. <|im_end|>
<|im_start|>user
Find a pizza restaurant nearby in Map. <|vision_start|>Screenshot_1.jpg<|vision_end|> <|im_end|>
<|im_start|>assistant
Several e-mails are displaying on the phone screen. To open Map, I need go back to the home screen and find the corresponding app icon.
FUNCTION: Home *ARGS*: {}
RESULT: <|vision_start|>Screenshot_2.jpg<|vision_end|>
RETURN: I return to the home screen. Next, I need to find the icon of Map and tap on it.
FUNCTION: Tap *ARGS*: {"point": (348,291)}
RESULT: <|vision_start|>Screenshot_3.jpg<|vision_end|>
RETURN: [Thinking for the next action.]
[Other subsequent actions.] .....
I have found the pizza restaurant nearby in Map. <|im_end|>

参考文献：① https://arxiv.org/pdf/2409.12191 ② https://zhuanlan.zhihu.com/p/25267823390

模型篇

Qwen2.5-VL 2025.2 效率优化代



- **滑动窗口注意力**

主体层采用固定窗口的局部注意力计算，每4层周期性地注入一次全局注意力。

- **动态分辨率**

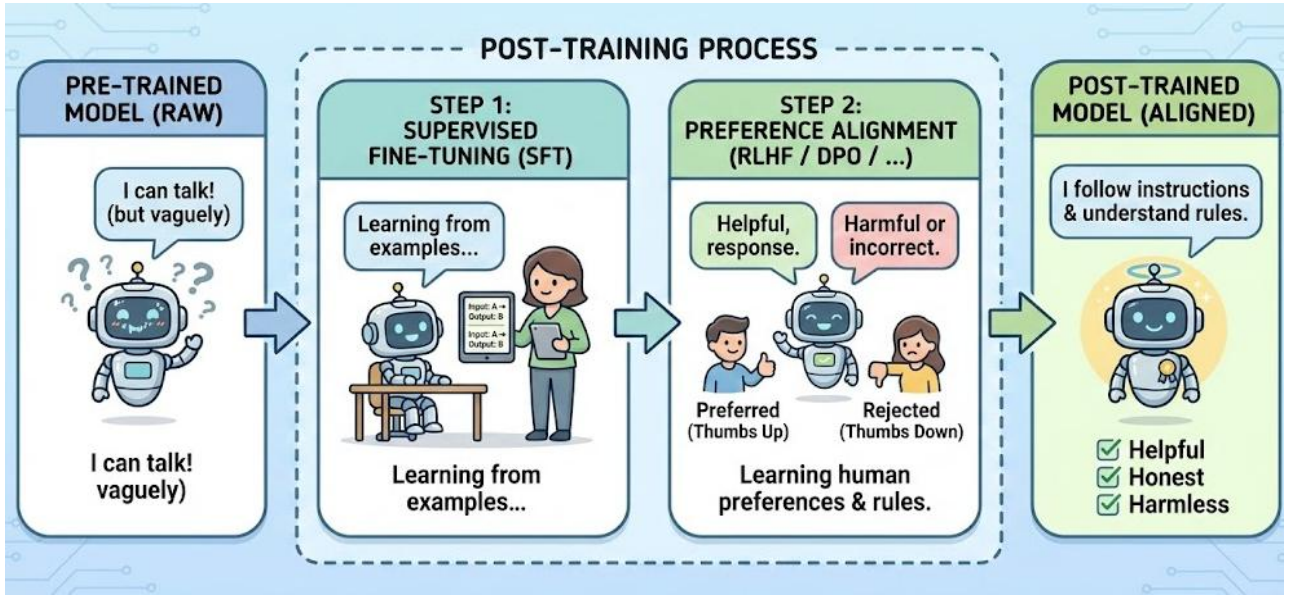
训练期对同一视频随机施加不同采样率提取帧序列，推理期兼容任意帧率输入。

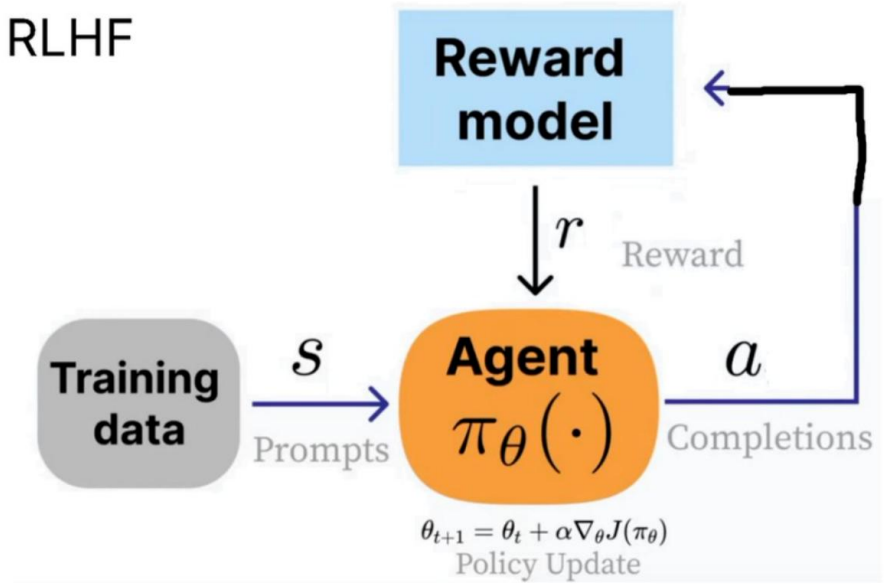
- **绝对时间编码对齐 M-RoPE 时间维**

摒弃传统“帧序号”位置编码，将每帧真实物理时间戳直接映射至M-RoPE时序(T)维度计算旋转矩阵。

参考文献：① https://arxiv.org/pdf/2502.13923

后训练篇





1. SFT (Supervised Fine-Tuning / 监督微调)

- **核心逻辑**：**“照我说的做，死记硬背。”**
- **怎么玩**：人类或强模型写好高质量的“问题-答案”对，强迫模型逐字逐句地模仿（交叉熵损失）。
- **优点**：简单、直接、稳定。只要数据质量极高，模型很快就能学会特定格式（比如 JSON 输出、礼貌用语）。
- **致命伤**：**分布偏移与上限锁死。**模型只是在“模仿”表面规律，并不知道“为什么”好。一旦遇到没见过的问题，容易胡言乱语（幻觉）；而且上限被人类标注者/数据集的水平死死卡住。

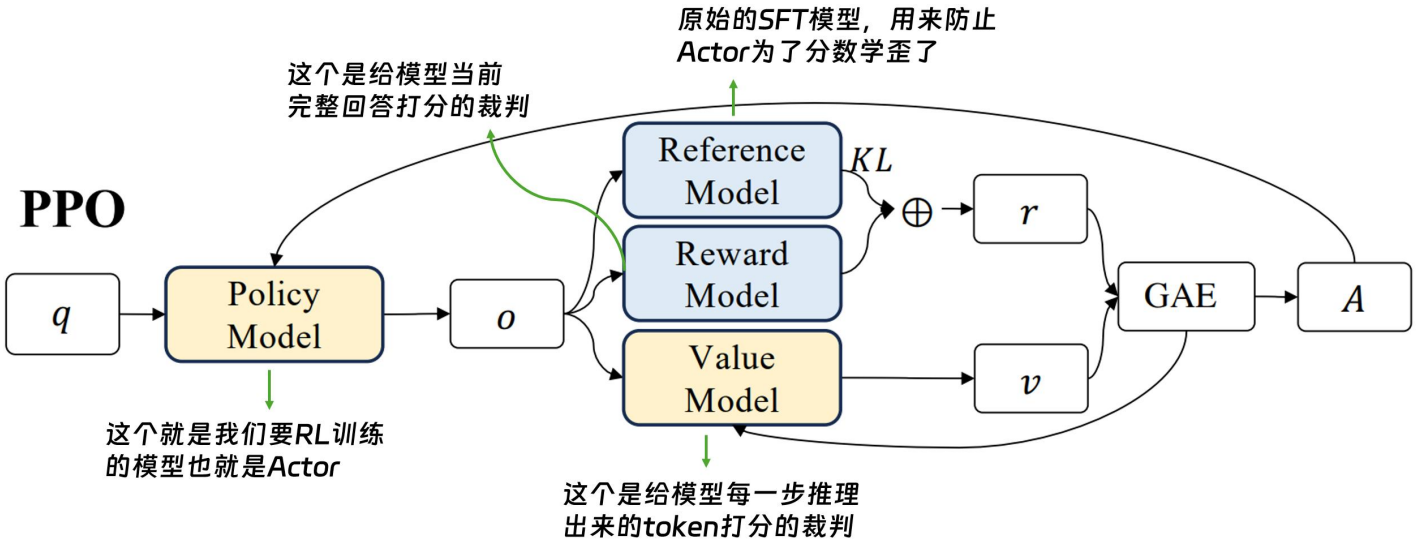
强化学习基本概念

强化学习是一种通过与环境交互、根据奖励信号自主学习最优策略的机器学习范式。核心要素包括：

- **智能体（Agent）**：做出决策的主体（如语言模型）
- **环境（Environment）**：智能体交互的外部系统
- **状态（State）**：环境在某一时刻的观测信息
- **动作（Action）**：智能体在某个状态下可执行的行为
- **奖励（Reward）**：环境对动作的即时反馈信号，指导智能体优化长期收益

与监督学习不同，强化学习不依赖"标准答案"，而是通过**"试错+奖励"**机制学习策略，特别适合序列决策、延迟反馈的任务。

后训练篇



$$L^{CLIP}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$$

I. Actor（需更新参数）：

- 角色：主角。这就是我们正在训练、希望它变符合我们预期的那个模型。
- 任务：负责根据问题生成回答。

II. Reference Model 参考模型（冻结参数）：

- 角色：SFT 后的原始模型副本。
- 任务：它不参与打分，而是作为约束。它时刻提醒 Actor：“你别为了拿高分就乱说话，要保持原本的语言能力，不要偏离太远（计算 KL 散度）。”

III. Reward Model 奖励模型（冻结参数）：

- 一般用什么：通常直接拿 SFT 训练好的模型（比如 Llama-3-8B-Instruct 或 Qwen3-14B-Instruct）来改。
- 角色：代表人类偏好的判卷人。
- 任务：它是个哑巴，平时不说话，只有等 Actor 把整句话写完了，它才给出一个最终分数（比如 8.0 分）。

IV. Critic 评判模型（更新参数）：

- 一般用什么：和 Reward Model 一样，通常也是由 Actor 同款的 Transformer 改出来的（比如 Actor 是 7B，Critic 通常也是 7B）。
- 维度的大不同：Critic 的独特之处在于它输出的形状，它的最终输出是[Batch_Size, Seq_Len, 1]，也就是 Actor 的每一个token都会有一个当下分。

➢ 缺点

1、工程复杂度极高：需要同时加载四个模型到显存中。

2、训练极不稳定：PPO 对超参数极其敏感，Reward Model 的打分容易出现 Scaling 问题，且采样的随机性导致梯度方差大。

3、效率低：需要在线生成数据（Rollout），生成长文本非常耗时。

后训练篇

DPO

在 RLHF 中，我们训练策略 π_θ 的目标是最大化奖励，同时通过 KL 散度约束模型不要偏离参考模型 π_{ref}（即 SFT 模型）太远：

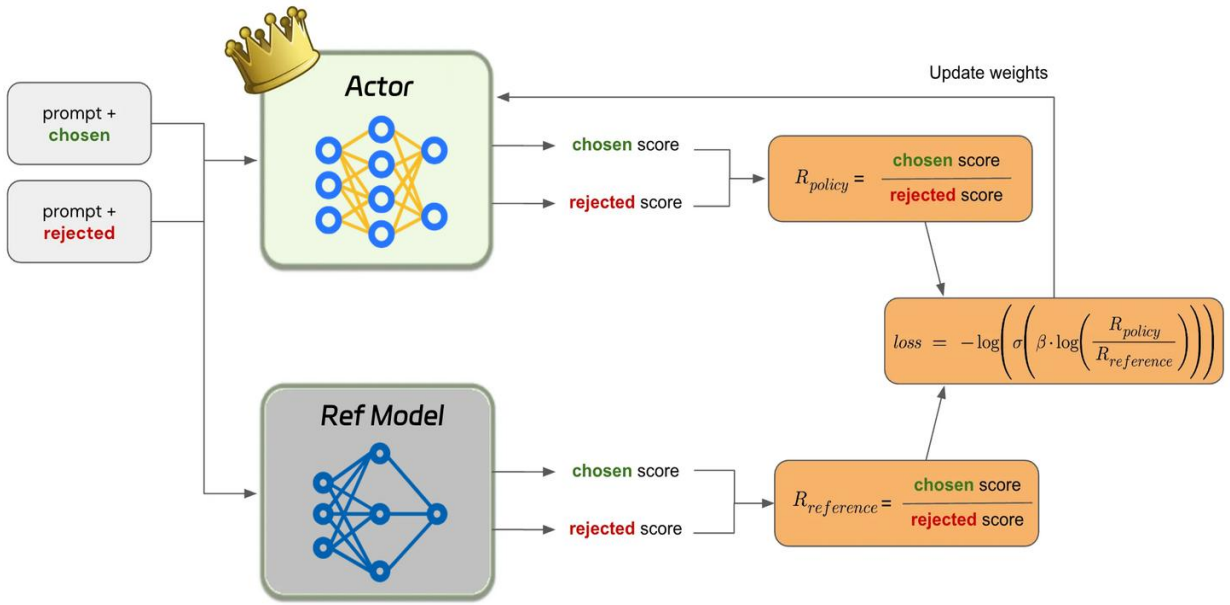
$$\max_{\pi_\theta} \mathbb{E}_{q\sim D, o\sim\pi_\theta(o|q)}[r(o,q)] - \beta\mathbb{D}_{KL}[\pi_\theta(\cdot|q)||\pi_{ref}(\cdot|q)]$$

这是一个典型的受约束的优化问题。数学上可以证明，对于任意给定的奖励函数 $r(o,q)$，上述目标函数的**最优解（Optimal Policy）** π^* 有一个显式解：

$$\pi^*(o|q) = \frac{1}{Z(q)}\pi_{ref}(o|q)\exp\left(\frac{1}{\beta}r(o,q)\right)$$

其中 $Z(q)$ 是归一化因子，仅与输入 q 有关。

这个公式告诉我们：**最优的策略分布，其实就是参考策略分布经过"奖励加权"后的结果。**

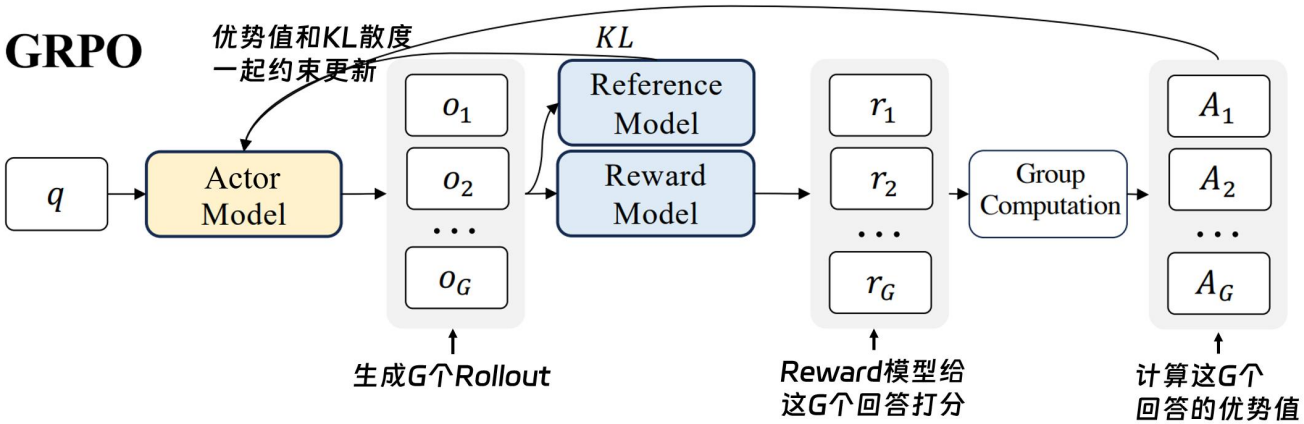


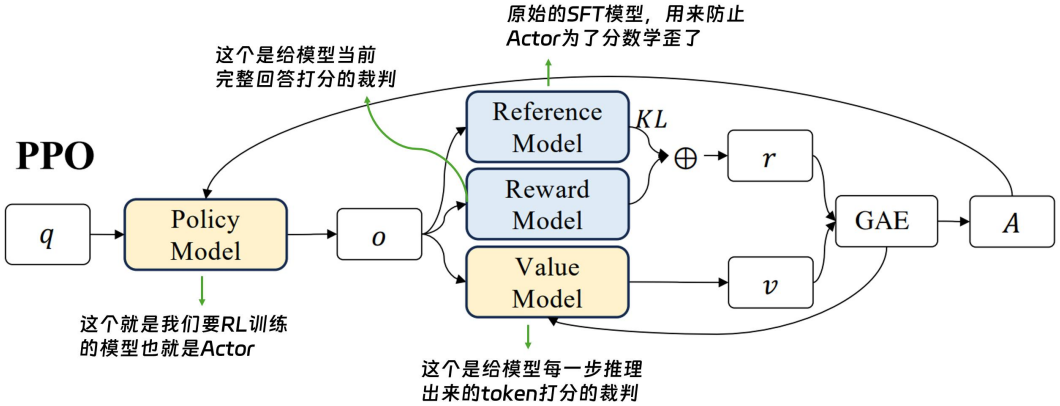
PPO 考验的是你的显卡财力，而 DPO 考验的是你的数据清洗能力

后训练篇

GRPO

核心思想：Baseline 不一定非要通过神经网络去预测，可以通过统计学的方法直接估算出来。





$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{q\sim D,\{o_i\}_{i=1}^{G}\sim\pi_\theta(q)} \left[\frac{1}{G}\sum_{i=1}^{G} \left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)} A_i, \text{clip}(\dots) A_i \right) - \beta D_{KL}(\pi_\theta||\pi_{ref}) \right) \right]$$

$$L^{CLIP}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right]$$

后训练篇

特性维度	PPO (Proximal Policy Optimization)	DPO (Direct Preference Optimization)	GRPO (Group Relative Policy Optimization)
核心思想	在可信区域内小步更新策略，以最大化奖励模型给出的分数。	直接将成对的偏好数据转化为一个分类损失，绕过奖励建模。	通过组内回答的相对好坏来估计优势，从而指导策略更新。
方法论	On-policy 强化学习	Off-policy 偏好学习 (类似于分类)	On-policy 强化学习
所需模型	4个：策略模型、价值模型 (Critic)、奖励模型 (RM)、SFT 参考模型。	2个：策略模型、SFT 参考模型。	3个 (通常)：策略模型、奖励函数/模型、SFT 参考模型。**无价值模型**。
数据需求	需要奖励模型能给出的绝对分数。	需要成对的偏好数据 。	需要奖励函数/模型能给出的分数 (可以是相对的)。
计算成本	**最高**。需要维护和训练多个大型模型，且有复杂的采样循环。	**最低**。流程类似监督微调，非常简洁高效。	**中等**。比 PPO 成本低（省去了价值模型），但比 DPO 复杂（仍有 RL 循环）。
主要优势	**灵活、鲁棒**。适用于各种复杂的奖励函数，是久经考验的工业标准。	**简洁、高效**。训练稳定，实现简单，大大降低了 RLHF 的门槛。	**高效、灵活**。显著降低了 PPO 的成本，且能灵活利用可验证奖励。
主要挑战	**复杂、昂贵**。实现和调试难度大，对资源消耗极高。	**依赖数据质量**。对偏好数据的一致性和质量要求高。	**组采样开销**。需要为每个指令生成多个样本，会增加推理开销。
适用场景	需要精细控制奖励函数、追求极限性能的通用场景。	数据集是成对偏好形式，希望快速、低成本地进行模型对齐的场景。	计算资源受限，或任务存在客观、可程序化验证的奖励标准的场景 (如代码、数学)。