# 视觉语言模型：编码器、像素与 grounding

## 目录

想回答的问题 . . . . . 2
一、架构范式与开源基模 . . . . . 2
二、视觉编码器 . . . . . 7
三、分辨率 . . . . . 10
四、像素信息：输入无损吗 . . . . . 11
五、Grounding：坐标表示与监督 . . . . . 16
附录：人与组谱系 . . . . . 21

| 工作 | 时间 | 研究问题 | 核心方法 | 核心结论 |
|---|---|---|---|---|
| **Flamingo** | 2022.04 | 怎样把冻结的视觉模型和语言模型接起来 | 交错图文 + 适配器 | 交错网页语料带来 few-shot 能力 |
| **LLaVA → LLaVA-1.5** | 2023.04 / 10 | 简单投影够不够 | 视觉编码器 + MLP + LLM | 数据比连接器重要 |
| **MM1** | 2024.03 | 哪些设计真正重要 | 系统消融 | 编码器、分辨率、token 数重要，连接器不重要 |
| **Qwen2-VL → 2.5-VL** | 2024.09 / 2025.02 | 原生分辨率与定位 | 2D-RoPE + 动态分辨率 + 绝对坐标 | 坐标系与视觉 token 网格对齐 |
| **SigLIP 2** | 2025.02 | 编码器能否在预训练时就学定位 | sigmoid 对比 + LocCa 等 | 定位和密集特征更好 |
| **GLIP / LocCa** | 2021.12 / 2024.03 | 预训练里怎样放位置监督 | 定位统一成预训练任务 | 编码器特征可读出框 |

## 想回答的问题

整理自 2026-11-26 在 CUA 笔记里的追问（这些内容最初写在 CUA 笔记 里，后拆出成本篇）：

- **A. 输入是否无损**：ViT 以 patch 为单位，patch 内的不同像素靠什么区分？哪些环节会真正丢像素信息？
- **B. 维度与容量**：投影进 LLM 时维度不够、固定数量 query 压缩、VAE 压缩，各丢多少？RAE 怎么讨论这个问题？
- **C. 编码器**：Qwen 用的是什么编码器？CLIP、SigLIP 与 VAE 的区别？encoder-free 和 encoder-based 的第一步是否一样？
- **D. Grounding**：坐标怎么从 token 里读出来？什么样的训练能逼模型关注像素？DeepSeek TVP 解决的是什么？
- **E. 架构与谱系**：VLM 架构范式怎么演进？哪些组在推动？

约定：只讨论理解（不讨论生成）的 VLM；VAE 只作为对照出现。

## 一、架构范式与开源基模

### 1. 2023 年的开源 VLM 基模

| 模型 | 时间 | 和 CUA 相关的能力 | 引用 |
|---|---|---|---|
| Pix2Struct（Google） | 2022.10 | 把遮挡后的网页截图解析成简化 HTML 作为预训练目标；可变分辨率输入 [1] | 576 |
| BLIP-2 | 2023.01 | 冻结视觉编码器和 LLM，中间接一个轻量桥接模块 | 10055 |

| 模型 | 时间 | 和 CUA 相关的能力 | 引用 |
|---|---|---|---|
| LLaVA | 2023.04 | 用纯文本 GPT-4 生成图文指令数据，视觉编码器 + LLM 端到端训练 [2] | 12295 |
| MiniGPT-4 / mPLUG-Owl | 2023.04 | 早期多模态对话模型 | 3485 / 1327 |
| Kosmos-2 | 2023.06 | 用位置 token 输出框：[文字](框) 的 Markdown 链接格式 [3] | 1417 |
| Shikra | 2023.06 | 坐标直接写成自然语言数字，不加新词表 [4] | 1123 |
| Qwen-VL（Alibaba） | 2023.08 | 用「图像-描述-框」三元组对齐，具备定位和文字识别 [5] | 2700 |
| LLaVA-1.5 | 2023.10 | CLIP-ViT-L-336px + MLP 连接器；1.2M 公开数据，8×A100 约 1 天训完 [6] | 6365 |
| Ferret（Apple） | 2023.10 | 离散坐标 + 连续区域特征的混合表示，可输入点、框、任意形状 [7] | 672 |
| CogVLM → CogAgent（智谱） | 2023.11 → 2023.12 | CogAgent 在 CogVLM 上加高分辨率分支到 1120×1120，专门识别网页小元素和小字 [8] | 889 / 988 |

**能力到哪了?**在 SoM 论文的同一子集上（RefCOCOg 指代理解）：GPT-4V 直接输出坐标 25.7，LLaVA-1.5 63.3，GPT-4V + SoM 86.4；Shikra、Ferret 原论文报 82.6、85.8（完整验证集）[9]。也就是说，**专门训过定位数据的开源模型，在自然图像上已经能比较准地框出物体**；但它们输入分辨率低（LLaVA-1.5 是 336px），也没见过 GUI 数据。CogAgent 把分辨率提到 1120×1120，理由正是要看清网页里的小元素 [8]。

**离 Qwen2.5-VL 还有多远？** Qwen-VL（2023.08）已经有框定位和文字识别，但输入只有 448、视觉 token 固定 256 个；Qwen2-VL（2024.09）换成动态分辨率；Qwen2.5-VL（2025.02）原生分辨率 + 框 / 点定位，并把 computer / phone use 作为原生能力。约一年半。

**LLaVA 这条线在做什么？**

| 版本 | 时间 | 重点 |
|---|---|---|
| LLaVA | 2023.04 | 视觉指令微调的起点 |

| 版本 | 时间 | 重点 |
| --- | --- | --- |
| LLaVA-1.5 | 2023.10 | 简单强基线，数据高效 |
| LLaVA-NeXT（博客） | 2024-01-30 | 输入像素提到 4 倍（最高 672×672、336×1344、1344×336），提升 OCR 和推理 [10] |
| LLaVA-OneVision | 2024.08 | 单图、多图、视频统一到一个模型（Higher AnyRes）[11] |

这条线主攻通用多模态理解，不做 GUI agent。和 SoM 的关系在人：Chunyuan Li 是 LLaVA、LLaVA-1.5、SoM、LLaVA-OneVision 四篇的共同作者；Hao Zhang、Feng Li 同时在 SoM 和 LLaVA-OneVision 的作者列表里 [9][11]。同一个圈子在 2023 年下半年一边用 SoM 去 prompt 闭源的 GPT-4V 做定位，一边把开源 LLaVA 做成通用基模。

另一个能对上的人员关系：Mobile-Agent 的作者 Haiyang Xu、Jiabo Ye 也在 mPLUG-Owl 的作者列表里，代码同在 X-PLUG 组织下 [12]。

### 2. VLM 架构范式：精读路线

下面的顺序参考了朋友转来的一份阅读建议（原清单由另一个模型生成），每篇的结论已按原文摘要核对。按「范式转变」而不是严格按年份排：

| # | 工作 | 时间 / 机构 | 核心结论（摘要原意） | 和 CUA 的关系 |
| --- | --- | --- | --- | --- |
| 1 | Flamingo | 2022.04，DeepMind | 桥接预训练的纯视觉模型和纯语言模型，能处理任意交错的图文序列；在交错网页语料上训练是获得 few-shot 能力的关键 [13] | 「冻结两端、中间接适配器」的起点 |
| 2 | BLIP-2 | 2023.01 | 冻结视觉编码器和 LLM，中间用轻量模块（Q-Former）做视觉瓶颈 | Qwen-VL 的 256 个 query 压缩是同类思路 |
| 3 | LLaVA → LLaVA-1.5 | 2023.04 / 2023.10 | 一个 MLP 投影就够；数据比连接器重要 [2][6] | 后来 GUI 模型大多沿用「编码器 + MLP + LLM」 |

| # | 工作 | 时间 / 机构 | 核心结论（摘要原意） | 和 CUA 的关系 |
|---|---|---|---|---|
| 4 | VILA | 2023.12，NVIDIA + MIT（Ji Lin、Song Han） | 预训练时冻结 LLM 只能零样本、没有 in-context 能力，要解冻；交错图文数据优于纯图文对；指令微调时混回纯文本数据既防文本能力退化，又提升 VLM 任务 [14] | 多图推理和 in-context 能力是多步 agent 的前提 |
| 5 | MM1 | 2024.03，Apple | 系统消融：图文对 + 交错 + 纯文本的配比对 few-shot 至关重要；**图像编码器、分辨率、图像 token 数影响大，连接器设计几乎无关紧要** [15] | 直接印证 GUI 的分辨率瓶颈（本章第 4 节） |
| 6 | Molmo / PixMo | 2024.09，AI2 | 不从闭源 VLM 蒸馏，全靠人工采集的新数据：详细描述、自由问答，以及一个 **2D pointing 数据集**；72B 在学术基准和人评上仅次于 GPT-4o[16] | **点击就是 pointing**——像素级指点能力的直接来源之一（问题 E） |
| 7 | Qwen2-VL → 2.5-VL → 3-VL | 2024.09 / 2025.02 / 2025.11，Alibaba | 动态分辨率 → 原生分辨率 + 窗口注意力 → 原生 256K 交错上下文、dense 与 MoE 两种规格 [17][18][19] | 开源 CUA 最常用的骨干（见 CUA 笔记第五章） |

| # | 工作 | 时间 / 机构 | 核心结论（摘要原意） | 和 CUA 的关系 |
|---|---|---|---|---|
| 8 | InternVL3 → 3.5 | 2025.04 / 2025.08，上海 AI Lab 等 | InternVL3：不再「先有文本 LLM 再接视觉」，而是在单一预训练阶段同时学多模态和语言（native multimodal pre-training）[20]；InternVL3.5：Cascade RL，先离线 RL 求稳、再在线 RL 精调；视觉分辨率路由器动态调整视觉 token[21] | 多模态 RL 的配方，可与 CUA 笔记第六章对照 |
| 9 | ERNIE 5.0 | 2026.02，百度（待核） | 所有模态从零开始、在统一的 next-group-of-tokens 目标下训练；超稀疏 MoE，专家路由与模态无关；弹性训练一次得到多个尺寸的子模型 [22] | 「彻底不从文本 LLM 续训」的原生多模态路线 |
| 10 | Molmo2 / LLaVA-OneVision-2 | 2026.01 / 2026.05 | Molmo2：视频和多图上的**点选式 grounding 与像素级跟踪**，强调连闭源模型也缺这个能力 [23]；LLaVA-OV-2：原生分辨率编码器 + 窗口注意力，把压缩视频流当作 token 分配依据 [24] | 视频 / 具身方向；pointing 从图像延伸到视频 |

**五次范式转变**（同一份建议的概括，作为阅读骨架）：跨模态连接（Flamingo）→ 指令微调（LLaVA）→ 配方与数据（MM1、Molmo）→ 原生分辨率视觉（Qwen2/2.5-VL）→ 原生多模态预训练（InternVL3、ERNIE 5.0）。

**MSRA Furu Wei 组**：主页列出的代表作包括 LayoutLM 系列、BEiT 系列、Kosmos / Kosmos-2 等 [25]，本笔记已读到的有 MarkupLM（HTML 结构预训练，本章第 2 节）、Kosmos-1（2023.02，从零训练的多模态 LLM，交错图文语料）[26]、Kosmos-2（用位置 token 输出框）[3]、BEiT（掩码图像建模预训练 ViT）[27]。这一组代表了「读结构 / 文档版面」这条预训练路线。

> **一句话**：VLM 从「冻结两端、中间接适配器」走到「强编码器 + MLP + LLM」，再到原生分辨率和原生多模态预训练；MM1 的消融说明真正重要的是编码器、分辨率和 token 数。

> 我的复述

> 自测
>
> 1. MM1 认为哪些设计重要、哪些不重要?
> 2. VILA 的三条预训练结论是什么?

## 二、视觉编码器

### 1. 编码器的迭代与谱系

| 编码器 | 时间 | 要点 | 谁在用 |
| --- | --- | --- | --- |
| CLIP（OpenAI） | 2021.02 | 图文对比学习 [28] | LLaVA-1.5（ViT-L-336px） |
| OpenCLIP ViT-bigG | — | CLIP 的开源复现与放大 | Qwen-VL |
| EVA-CLIP | 2023.03 | 改进 CLIP 训练，5.0B 的 EVA-02-CLIP-E 在 ImageNet 零样本 82.0[29] | CogVLM / CogAgent（EVA2-CLIP-E） |
| SigLIP（Google） | 2023.03 | 把 softmax 对比损失换成逐对 sigmoid，不需要全局归一化，batch 可放大也可缩小 [30] | LLaVA-OneVision（384px，729 token） |
| InternViT-6B | 2023.12 | 把视觉编码器放大到 6B，再逐步与 LLM 对齐 [31] | InternVL |
| Qwen2.5-VL 自研 ViT | 2025.02 | 从头训练；2D-RoPE + 窗口注意力 + SwiGLU / RMSNorm；按原生分辨率输入 [18] | Qwen2.5-VL |

**一条贯穿的编码器谱系**：**Google Brain Zürich → Google DeepMind**。Lucas Beyer、Xiaohua Zhai、Alexander Kolesnikov 等人在下面几乎每篇里都署名，不是一个人的工作，而是一个组长期推进的路线 [32][33][34][35][36][37][38]：

| 工作 | 时间 | 解决了什么 |
|---|---|---|
| BiT | 2019.12 | 大规模有监督预训练 + 简单迁移配方 |
| ViT | 2020.10 | 纯 Transformer 处理图像块，不再依赖 CNN |
| Scaling ViT | 2021.06 | ViT 的模型 / 数据 / 算力缩放规律 |
| LiT | 2021.11 | 冻结预训练图像模型、只训文本塔，就能做零样本迁移 |
| SigLIP | 2023.03 | sigmoid 逐对损失替代 softmax 对比损失 [30] |
| PaLI-3 | 2023.10 | 对比 SigLIP 与分类预训练的 ViT：SigLIP 在分类上略差，但在**定位和图中文字理解**上明显更好；SigLIP 编码器放大到 2B |
| PaliGemma | 2024.07 | 开源 3B VLM：SigLIP-So400m + Gemma-2B，面向迁移 |
| SigLIP 2 | 2025.02 | 在原目标上加入 captioning 预训练、自蒸馏、掩码预测和在线数据筛选；标题即点明改进了**定位和密集特征** |

这条线回答了一个问题：为什么现代 VLM 普遍收敛成「一个很强的预训练视觉编码器 + 一个 LLM」，而不是复杂的融合模块。PaLI-3 的发现对 CUA 尤其相关——GUI 正好是「定位 + 图中文字」最密集的场景。

### 2. 前沿模型的编码器怎么保住位置

（来源：Purshow Notes 的 Vision Encoder 与 Encoder-Free 两篇 [39][40]；属二手整理，作者据各家公开代码和报告，本笔记未逐项回查官方代码）

(*1*) 一个送进 *LLM* 的视觉 *token* 覆盖多少像素

| 模型 | 每个 **LLM** 视觉 **token** 的空间粒度 | 怎么从补丁合成 | 视觉位置编码 |
|---|---|---|---|
| Kimi K3（MoonViT-V2） | 28×28 | 14 像素补丁，末端空间 2×2 拼到通道维；视频再对最多 4 帧取均值 | 64×64 可学习表插值 + 2D RoPE |
| Qwen3.8-Flash-Next | 32×32 | 16 像素补丁，空间 2×2 拼接 | 48×48 绝对位置表插值 + 2D RoPE |
| DeepSeek-V4.1-Flash | **42×42** | 14 像素补丁，末端 3×3 pixel-unshuffle（$N \to N/9$） | 无绝对位置表，只用 2D RoPE |

| 模型 | 每个 **LLM** 视觉 **token** 的空间粒度 | 怎么从补丁合成 | 视觉位置编码 |
|---|---|---|---|
| GLM-5.3-Flash | 28×28 | 末端用可学习的步长 2 卷积压缩 | 只用 2D axial RoPE |
| MiMo-V2.6-Pro / dots3 | 32×32 / 28×28 | 空间 2×2 拼接 | 只用 2D axial RoPE |
| Thinking Machines Inkling（encoder-free） | **40×40** | HMLP 四级局部折叠，**补丁之间没有 attention** | **视觉模块内没有任何位置编码** |

**pixel shuffle / pixel-unshuffle 是什么**：两者是一对互逆的张量重排，不做任何计算。pixel shuffle 出自超分辨率论文 ESPCN（Shi et al. 2016）[41]：把 $(C\cdot r^2,\ H,\ W)$ 的特征重排成 $(C,\ rH,\ rW)$，即**通道换空间**，用来在低分辨率上算完再放大。pixel-unshuffle（也叫 space-to-depth）反过来：$(C,\ H,\ W)\to(C\cdot r^2,\ H/r,\ W/r)$，**空间换通道**。VLM 里「相邻 2×2 个视觉 token 拼成一个」就是 $r=2$ 的 pixel-unshuffle：token 数变成 1/4，每个 token 的维度变成 4 倍，四个子 token 各占一段通道，谁在左上谁在右下都还在。Seg2Click 用的是反方向的 pixel shuffle：把通道里的信息重新摊回空间，得到比视觉 token 更细的掩码。

前沿模型的一个视觉 token 普遍覆盖 28–42 像素见方。合成方式大多是**把相邻补丁拼到通道维**（pixel-unshuffle），而不是平均池化——这样子位置的信息还留在通道里，LLM 才有可能定位到一个 token 内部。（这一句是本笔记的推断。）

(*2*) 编码器预训练时就训练定位：*SigLIP 2* 的 *LocCa*

SigLIP 2 在 sigmoid 对比损失之外，全程加一个只在训练时使用的 cross-attention 解码器（LocCa），在 ViT 补丁特征上做三个生成任务：整图描述、**区域描述 → 框坐标**、**框坐标 → 区域描述**。区域框的伪标签由开放词表检测器从 alt-text 词组自动生成；解码器在推理时丢掉。也就是说，**视觉编码器在预训练阶段就被要求让补丁特征能读出坐标**——这是「编码器和定位输出协同设计」在编码器一侧最直接的例子。同时它用 NaFlex 保持原始长宽比，按 128 / 256 / 576 / 784 / 1024 几档 token 预算训练，位置编码是可学习的绝对位置表加插值，不用 RoPE。

(*3*) 最后一层不一定适合定位：*Perception Encoder*

Meta 的 PE-G（1.88B）发现最好的视觉特征在中间层：检测和深度在第 39 层、跟踪在第 32 层、OCR 在第 38 层；从第 33 层起出现「全局 token」，后面十几层逐渐做全局语义聚合，**局部空间信息因此丢失**。所以另训一个 PE-spatial：蒸馏第 41 层特征和 SAM 2.1 的掩码对应关系，让最后一层重新具备密集的空间特征。对 CUA 的含义：对比学习得到的编码器，末层特征偏语义、偏全局，直接拿来定位不一定合适。

(*4*) *Encoder-free*：把定位完全交给 *LLM*

MAI-Base-2、NEO、SenseNova-U1、Gemma4-12B、Inkling 等去掉深层 ViT，只用轻量的补丁投影。

**encoder-free 和 encoder-based 的第一步其实一样**：都是切块、展平、线性投影（ViT 的第一层本来就是 patch embedding）。区别只在投影之后：encoder-based 再接几十层 ViT，encoder-free 直接（或经很浅的变换）送进 LLM。所以上面「补丁内位置靠投影矩阵的不同列编码、维度够就无损」的分析，对 encoder-free 同样成立。以 Inkling 的 HMLP 为例（据 Purshow 的整理 [39]），它是**四级 pixel-unshuffle + MLP**，每级的输出维度都不小于它覆盖的原始像素数：

| 级 | 空间折叠 | 覆盖原始像素（RGB） | 输入 → 输出维度 |
|---|---|---|---|
| 1 | 5×5 像素拼到通道 | 5×5×3 = 75 | 75 → 128 |
| 2 | 2×2 拼接 | 10×10×3 = 300 | 512 → 320 |
| 3 | 4×4 拼接 | 40×40×3 = 4,800 | 5,120 → 4,800 |
| 4 | 两个时间位置拼接 | 静态图复制时间维，有效仍是 4,800 | 9,600 → 6,144 |

对静态图，每一级输出维度都 ≥ 覆盖的原始像素数（第 3 级正好相等），维度上留足了无损的空间；对真实视频，第 4 级把两帧的 9,600 个数压到 6,144，才开始真正压缩。每级中间有 RMSNorm 和 GELU，是非线性的，所以这里说的只是「维度够」，不等于严格可逆。Gemma4-12B 等其他 encoder-free 模型的具体结构本笔记没有读到原始资料。这样，定位的分辨率上限由补丁投影决定（Inkling 是 40 像素），「从 token 还原像素」必须完全由 LLM 主干学会。Purshow 主张这条路的理由主要在训练系统：ViT 和 LLM 的计算形状、并行策略不同，视觉负载随样本变化，会拖慢流水线；去掉编码器后，调度和性能建模更接近纯文本 LLM。

> **一句话**：现代 VLM 普遍收敛成「强预训练编码器 + LLM」；编码器一侧有 Google Brain Zürich 的 ViT → SigLIP 谱系，前沿模型每个视觉 token 覆盖 28–42 像素见方，合成方式多为拼接（pixel-unshuffle）。

> 我的复述

> 自测
>
> 1. PaLI-3 比较 SigLIP 与分类预训练的 ViT，发现了什么？
> 2. pixel-unshuffle 和平均池化在保留位置信息上有什么区别？

## 三、分辨率

### 1. 分辨率的几种处理方式

屏幕是 1920×1080 甚至 4K，按钮和字很小。CogAgent 给过一个算账：1120×1120 的图按 14 像素切块是 6,400 个 token[8]。各家的解法：

| 模型 | 时间 | 分辨率处理 |
|---|---|---|
| LLaVA-1.5 | 2023.10 | 固定 336×336[6] |
| Qwen-VL | 2023.08 | 训练第二阶段把输入从 224 提到 448；用 cross-attention 适配器把视觉特征压到固定 256 个，并加 2D 绝对位置编码保住位置信息 [5] |
| Monkey | 2023.11 | 把图切成多块 448×448，每块单独适配，最高 1344×896[42] |

| 模型 | 时间 | 分辨率处理 |
|---|---|---|
| CogAgent | 2023.12 | 双分支：224 的低分辨率大编码器（EVA2-CLIP-E）+ 0.30B 的高分辨率分支（1120×1120），用 cross-attention 融合，在分辨率和算力间折中 [8] |
| LLaVA-NeXT | 2024.01 | AnyRes：最高 672×672、336×1344、1344×336[10] |
| Ferret-UI | 2024.04 | 按屏幕方向切成 2 张子图分别编码（竖屏横切、横屏竖切）[43] |
| LLaVA-OneVision | 2024.08 | Higher AnyRes，SigLIP 384 为单元 [11] |
| Qwen2-VL | 2024.09 | Naive Dynamic Resolution：不同分辨率的图变成不同数量的 token；M-RoPE[17] |
| Qwen2.5-VL | 2025.02 | 原生分辨率：长宽缩放到 28 的倍数直接送入 ViT，用窗口注意力控制算力 [18] |

**脉络**：固定小分辨率 → 切块拼接（Monkey、AnyRes）→ 专门的高分辨率分支（CogAgent）→ 原生动态分辨率（Qwen2-VL / 2.5-VL）。CogAgent 声称是第一个只看截图就在 AitW 和 Mind2Web 上超过「读 HTML 的 LLM」的通用 VLM[8]——分辨率是让纯截图路线可行的前提之一。

**一句话**：分辨率处理从固定小图走到切块拼接、高分辨率分支，再到原生动态分辨率；这是纯截图路线能走通的前提之一。

我的复述

自测

1. CogAgent 为什么用双分支？
2. Qwen2-VL 的 Naive Dynamic Resolution 解决了什么问题？

## 四、像素信息：输入无损吗

本章四组动画：patch 嵌入、VAE 压缩、维度瓶颈、固定 query（网页版可交互）。

### 1. 补丁内的不同像素，模型靠什么区分？

**不是靠位置编码**。位置编码（绝对位置表或 2D RoPE）只告诉模型「这是第几行第几列的补丁」。补丁内部的位置由**补丁嵌入的线性投影**区分：

$$\mathbf{t} = W\,\mathrm{vec}(\text{patch}) + \mathbf{b}, \qquad W \in \mathbb{R}^{D\times 3p^2}$$

补丁里第 $(u, v)$ 个像素的第 $c$ 个通道，乘的是 $W$ 中**专属于它的那一列**。同样一个亮点，落在补丁左上角和右下角，乘的是不同的列，得到的 token 向量就不同——补丁内的位置被编码进了「哪一列被激活」。若 $D \geq 3p^2$ 且 $W$ 满秩，这个映射是单射，**补丁化这一步可以完全无损**（这是线性代数推导，不依赖具体论文）。

【网页版此处有动画】动画：切 *patch* → 展平 → 选中 *W* 的一列 → 位置编码只认 *patch* → 可逆还原 → 拼接 *vs* 平均 → 缩小才真正丢信息。

| 编码器 | 补丁 | 补丁内原始数值 $3p^2$ | 嵌入维度 $D$ |
|---|---|---|---|
| SigLIP2-B | 16×16 | 768 | 768[44] |
| DINOv2-B | 14×14 | 588 | 768[44] |
| DeepSeek-ViT（据 Purshow） | 14×14 | 588 | 1024[39] |

补丁化之后的 ViT 层是非线性的，训练目标又偏语义，信息可能在后面丢掉。有一组直接的反证：RAE（2025.10）把**冻结的** DINOv2-B、SigLIP2-B 编码器接一个 ViT 解码器去重建 ImageNet 图像，重建 FID 分别是 0.49、0.53，好于 Stable Diffusion 的 VAE（0.62）；作者明确反驳了「表征编码器只保留高层语义、丢掉低层细节」的假设 [44]。也就是说，**至少对 256 分辨率的自然图像，这些语义编码器的补丁特征里仍保留着足以重建像素的信息。**

### 2. 哪些环节会损失像素信息?

判据很简单：**某一步之后的数值个数，是否少于它所覆盖的原始像素数值个数。**少了就必然有损——线性映射的秩不超过输出维度；对一般的连续映射，从 $\mathbb{R}^n$ 到 $\mathbb{R}^m$（$m < n$）也不存在连续的单射。不少于只是「有可能无损」，还要看映射本身。按这个判据逐个看：

| 环节 | 是否有损 | 说明 |
|---|---|---|
| 切块 + 线性投影 | 维度够就可以无损 | $D \geq 3p^2$ 且满秩即可逆（见上） |
| **送入前缩小截图** | **有损** | 多个像素平均成一个，不同原图可能变成同一张小图；OSWorld 降采样实验 |
| 相邻 token 拼接（pixel-unshuffle） | 不损 | 只是重排 |
| 相邻 token **取平均** / 时间维取平均 | **有损** | 动画第 6 幕；Kimi K3 对最多 4 帧取均值，视频的帧间细节在这里压掉 [39] |
| **固定数量的 query 压缩** | 大图必然有损 | Qwen-VL 无论多大的图都压成 256 个视觉 token[5]，BLIP-2 的 Q-Former 同理 |
| **投影到 LLM 时的维度瓶颈** | 维度不足就有损 | 见下表 |
| 深层非线性 ViT + 语义训练目标 | 不保证保留 | Perception Encoder 末层丢局部空间信息；RAE 显示冻结编码器仍能高质量重建，但重建 FID 不为 0，不是严格无损 [44] |

| 环节 | 是否有损 | 说明 |
|---|---|---|
| **VAE** | 设计上就有损 | 目标就是压缩：LDM 论文把第一阶段称为「感知压缩」，会去掉高频细节 [45]；Wan2.1-VAE 时间 × 高 × 宽压缩 4×8×8、16 个通道，元素压缩比 48:1（据 Purshow 的 Video VAE 笔记 [46]）；离散化的 VQ tokenizer 还要再加一层量化损失 |

【网页版此处有动画】动画：*ViT* 补丁嵌入变宽、*VAE* 变窄；*VAE* 压掉的是高频细节。

**前沿 VLM 每个视觉 token 的维度 vs 它覆盖的原始像素数**（静态图像；结构数字据 Purshow 的整理 [39]，计算为本笔记所做）：

| 模型 | 每 token 覆盖 | 原始数值 | 送入 LLM 的维度 | 维度上是否够 |
|---|---|---|---|---|
| Kimi K3 | 28×28×3 | 2,352 | 7,168 | 够 |
| GLM-5.3-Flash | 28×28×3 | 2,352 | 4,096 | 够 |
| dots3 | 28×28×3 | 2,352 | 5,120 | 够 |
| Muse Spark | 28×28×3 | 2,352 | 8,192 | 够 |
| MiMo-V2.6-Pro | 32×32×3 | 3,072 | 6,144 | 够 |
| Qwen3.8-Flash-Next | 32×32×3 | 3,072 | **2,560** | 不够（约 0.83 倍） |
| DeepSeek-V4.1-Flash | 42×42×3 | 5,292 | **5,120** | 略不够（约 0.97 倍） |
| Inkling (encoder-free) | 40×40×3 | 4,800 | 6,144 | 够 |

维度够是无损的必要条件，不是充分条件；维度不够则一定有损。真实截图的像素高度相关（大片纯色背景），实际需要的信息量远小于原始数值个数，所以「维度略不够」不一定影响定位。

### 3. 维度瓶颈：RAE 的实验与 Qwen 的维度账

RAE 第 4.1 节专门做了这个实验 [44]：拿一张图，用 DINOv2-B 编码（每个 token $n = 768$ 维），让扩散模型去拟合这一张图。

- 固定深度、改宽度：DiT 宽度 $d$ 从 384 加到 784，$d < 768$ **时生成质量很差、损失降不下去；一旦** $d \geq 768$，**几乎完美复现原图**。
- 固定宽度 $d = 384$、把深度从 12 层翻倍到 24 层：仍然满是伪影，损失也降不到 $d = 768$ 的水平。所以是**宽度**的问题，不是总参数量的问题。
- 定理 1 给了下界：模型内部只有 $d$ 维通道时，单样本情形下平均损失至少是 $\frac{n-d}{n}$。
- 他们的解释：扩散训练往数据上直接加满维高斯噪声，把原本低维的数据流形「扩散」成了满秩分布，模型容量必须跟着完整的数据维度走。这正好和「数据流形本征维度低」的常见看法相反，作者认为矛盾来自扩散的建模方式。

- 工程对策：不加宽整个 DiT，只接一个浅而宽的去噪头（DDT head）；噪声调度按「有效数据维度 = token 数 × 每个 token 的维度」做平移。

【网页版此处有动画】动画：维度不够还原不回来 → 零空间里不同 *patch* 得到同一 *token* → *RAE* 的下界 → 前沿 *VLM* 的维度账。

**搬到 VLM 上要打折扣**：VLM 的视觉投影是判别式的，不往输入里加噪声，「本征维度低」的论点在那里仍然成立——截图的大片纯色背景本来就不需要那么多维。所以 RAE 的结论能直接说明的是「宽度小于 token 维度时，信息有硬性的丢失下界」，至于对定位精度有多大影响，要看具体任务。

**Qwen2.5-VL 的维度账**（论文表 1[18]）：ViT 宽 1,280，补丁 14；一个 14×14 RGB 补丁是 588 个数 → 1,280 维，够。2×2 合并后 $4 \times 1280 = 5120$ 维，再经 MLP 投到 LLM 的宽度：

| 规格 | 每个 **LLM token** 覆盖 | 原始数值 | 合并器输出（= **LLM** 宽度） | 维度上是否够 |
|---|---|---|---|---|
| Qwen2.5-VL-3B | 28×28×3 | 2,352 | **2,048** | **不够**（约 0.87 倍） |
| Qwen2.5-VL-7B | 28×28×3 | 2,352 | 3,584 | 够 |
| Qwen2.5-VL-72B | 28×28×3 | 2,352 | 8,192 | 够 |

**Qwen 的视觉编码器是什么？**不是 VAE，是 CLIP 一类的对比学习编码器：

- **Qwen2-VL**：约 675M 参数的 ViT，**用 DFN 的 ViT 初始化**，把原来固定的位置编码换成 2D RoPE[17]。DFN（Apple，2023.09）是一篇研究「用一个网络过滤网络图文数据」的论文，其产物是在过滤后数据上训练的 CLIP 模型 [47]。
- **Qwen2.5-VL**：**从头训练**的新 ViT（32 层、宽 1,280、SwiGLU + RMSNorm、窗口注意力，只有第 7 / 15 / 23 / 31 层是全局注意力），训练分三段：CLIP 预训练 → 视觉-语言对齐 → 端到端微调 [18]。论文只写了「CLIP pre-training」，**没有说明用的是 softmax 对比损失还是 SigLIP 的 sigmoid 损失**。

**对比学习编码器和 VAE 的区别**

| | **CLIP / SigLIP 类编码器** | **VAE** |
|---|---|---|
| 训练目标 | 图文配对：让匹配的图和文的向量靠近 | 重建像素 + KL 正则 |
| 关心什么 | 语义：图里是什么 | 外观：能否画回原图 |
| 每个 token 的维度 | 高，一般 ≥ 补丁内原始数值（588 → 1,280） | 低，刻意压缩（Wan2.1-VAE：4×8×8×3 = 768 个数 → 16 个通道，48:1） |
| 用在哪 | VLM 的「眼睛」 | 扩散模型的潜空间 |

CLIP 和 SigLIP 的差别只在损失：CLIP 对整个 batch 做 softmax，SigLIP 对每一对图文单独做 sigmoid 二分类（第二章编码器一节）。两者都**没有被要求保留像素**，信息是否保留靠的是维度充足和训练数据；RAE 的重建实验说明它们实际上保留了很多。

#### 4. 固定数量 query 的容量

**算法很简单**：$Q$ 个 query、每个 $d$ 维，总容量是 $Q \cdot d$ 个数，**与输入分辨率无关**；输入的原始数值是 $H \cdot W \cdot 3$。前者小于后者就必然有损，而且图越大越亏：

$$\text{容量比} = \frac{Q \cdot d}{H \cdot W \cdot 3} \propto \frac{1}{HW}$$

| 模型 | 压缩方式 | 容量 | 对照 |
|---|---|---|---|
| BLIP-2[48] | Q-Former，32 个 query × 768 维 | 24,576 | 冻结 ViT-L/14 的特征是 257×1,024 = 263,168；224×224 原图是 150,528 个数。作者明说这是**有意设计的信息瓶颈**，配合预训练目标「迫使 query 只提取和文本最相关的视觉信息」 |
| Qwen-VL[5] | 256 个 query 的 cross-attention 适配器 | $256 \times d$ | 448×448 输入是 602,112 个数，每个 query 要 ≥ 2,352 维才在维度上够；同样 256 个 token 若要装下一张 1920×1080 截图（6,220,800 个数），每个要 ≥ 24,300 维——所以只能先缩到 448 |

【网页版此处有动画】动画：固定 *query* 的容量账 → 抽象器的注意力集中在少数区域 → 一一对应与局部池化保留位置。

**谁在研究这件事**

- **Honeybee（Kakao Brain，2023.12）**[49]：给投影器定了两个要求——token 数量可灵活调节（为了效率），以及**保留局部上下文**（为了空间理解）。它发现 resampler、Q-Former 这类「抽象器」在空间理解任务上比线性投影器难学：抽象过程没有局部性设计，注意力集中在少数区域，丢掉了细节；线性投影是一对一变换，局部信息全在。于是提出 C-Abstractor（卷积）/ D-Abstractor（可变形注意力），既能调 token 数，又保留局部性。
- **DeCo（2024.05）**[50]：沿着语言 token 往回追语义流，发现 Q-Former 这类压缩式投影器把补丁抽象成少数几个「物体 / 属性」概念，LLM 再按指令抽象一次——「双重抽象」，造成视觉语义的累积缺失。改法是**在补丁层面压缩数量**（无参数的 2D 自适应池化），把语义抽象完全交给 LLM；视觉定位任务上提升 7.1%。
- **MM1（2024.03）**：连接器的类型几乎无关紧要，图像 token 数和分辨率影响大（第一章精读路线）。

- **FastV（2024.03）**[51]：推理时剪枝——发现 LLaVA-1.5、Qwen-VL 等模型深层对视觉 token 的注意力极其低效，第 2 层之后剪掉一半视觉 token，LLaVA-1.5-13B 的计算量减少 45%，在通用理解任务上不掉点（论文没有测 GUI 定位）。
- GUI 里的同类做法：ShowUI 按 UI 结构挑视觉 token（减少 33%），InternVL3.5 用分辨率路由器动态调节视觉 token 数。

**结论**：前沿模型基本放弃了固定数量 query，改为「MLP 投影 + 相邻 token 拼接」（Purshow 也提到 Q-Former「目前应该也没有多少人再使用了」）。Honeybee 和 DeCo 的发现对 grounding 尤其要紧：**比信息总量更重要的是空间局部性**——DeCo 用的平均池化在数值上同样有损，但保留了「哪个 token 对应哪块区域」，定位反而更好。

**一句话**：切 patch 加线性投影在维度够时可以无损，patch 内的位置由投影矩阵的不同列编码，位置编码只负责 patch 级位置；真正丢像素信息的是输入前缩小、取平均或固定 query 压缩、投影维度不足，以及以压缩为目标的 VAE。

我的复述

自测

1. patch 内的位置由什么编码？位置编码负责什么？
2. Qwen2.5-VL-3B 为什么在维度上「不够」？
3. 固定数量 query 的容量比为什么随分辨率按 $1/(HW)$ 下降？

## 五、Grounding：坐标表示与监督

### 1. 坐标表示和编码器是一起演化的

CUA 的核心能力是把一句话落到屏幕上的一个像素位置。模型内部只有 token，问题是：**坐标怎么从 token 里「读」出来？** 这件事有两半——输出端（坐标写成什么）和输入端（编码器怎么保留位置）——它们是一起演化的。

| 时间 | 工作 | 坐标怎么输出 | 编码器 / 位置那一侧 |
|---|---|---|---|
| 2021.09 | Pix2Seq（Google，Ting Chen、Geoffrey Hinton）[52] | 把检测当成语言建模：框的角坐标均匀量化成 $[1, n_{\text{bins}}]$ 的离散 token；600×600 的图用 600 个 bin 就零量化误差 | 普通视觉骨干；作者的直觉是「网络若已经知道物体在哪，只需教它读出来」 |
| 2022.02 | OFA（Alibaba）[53] | 框角坐标量化为**位置 token**，和文字、图像码共用一个词表 | — |

| 时间 | 工作 | 坐标怎么输出 | 编码器 / 位置那一侧 |
|---|---|---|---|
| 2023.06 | Kosmos-2（MSRA）[3] | 224×224 的图切成 32×32 个 bin（每个 7×7 像素），新增 1,024 个位置 token；用 Markdown 链接 `[文字](框)` 输出 | 补丁 14×14：一个 bin 正好是半个补丁 |
| 2023.06 | Shikra[4] | 坐标直接写成自然语言数字，**不加新词表** | — |
| 2023.07 | NaViT（Google）[54] | — | **原生分辨率 ViT**：用序列打包处理任意分辨率和长宽比，不再先缩放到固定尺寸 |
| 2023.08 | Qwen-VL[5] | 框坐标归一化到 $[0, 1000)$，写成 `(x1,y1),(x2,y2)` 文本，**不需要额外的位置词表**，用特殊 token 标出框的起止 | 固定 448 输入；适配器的 cross-attention 里加 2D 绝对位置编码，防止压缩时丢位置 |
| 2023.10 | Ferret（Apple）[7] | 离散坐标 + 连续区域特征的混合表示 | 空间感知的视觉采样器 |
| 2024.01 | SeeClick[55] | GUI 上第一次做 grounding 预训练：0–1 的相对坐标，保留两位小数的文本 | 基于 Qwen-VL |
| 2024.09 | Qwen2-VL[17] | 仍是 $[0, 1000)$ 归一化文本坐标 | **ViT 去掉绝对位置编码，换成 2D-RoPE**；动态分辨率；相邻 2×2 个视觉 token 用 MLP 合成一个；LLM 里用 M-RoPE 区分图像的高、宽维度 |
| 2024.09 | Molmo[16] | 点（pointing）作为输出，专门采集 2D pointing 数据 | — |
| 2024.10 | Claude computer use[56] | 训练模型「数像素」 | 未公开 |
| 2025.01 | UI-TARS[57] | 直接预测要操作元素的坐标 | 基于 Qwen2-VL |

| 时间 | 工作 | 坐标怎么输出 | 编码器 / 位置那一侧 |
|---|---|---|---|
| 2025.02 | Qwen2.5-VL[18] | **绝对坐标**：框和点都用输入图像的实际像素尺寸表示。论文明说相对坐标「无法有效表示物体在图中的原始大小和位置」 | 原生分辨率：长宽缩放到 28 的倍数直接送入 ViT；2D-RoPE + 窗口注意力 |
| 2025.06 | GUI-Actor（Microsoft）[58] | **不输出坐标**：用一个 `<ACTOR>` token 对所有视觉补丁 token 做注意力，直接选出动作区域；再用验证器挑最可信的候选 | 论文指出的问题：屏幕坐标是稠密的，而 ViT 特征只有补丁级粒度，两者不匹配 |

**一个视觉 token 对应多少像素?**（以 Qwen2-VL / 2.5-VL 为例）ViT 按 14 像素切补丁，之后相邻 2×2 个补丁合成一个送进 LLM 的 token：

$$\text{每个 LLM 视觉 token 覆盖 } (14 \times 2) \times (14 \times 2) = 28 \times 28 \text{ 像素}$$

所以 Qwen2.5-VL 要求长宽是 28 的倍数——**坐标系和视觉 token 网格是对齐的**。一张 1920×1080 的截图大约变成 $69 \times 39 \approx 2{,}700$ 个视觉 token（按 28 的倍数取整，实际取整规则以实现为准）。而一个按钮可能只有十几像素宽：模型输出的绝对坐标必须精确到**一个 token 内部**的位置，这正是「从 token 里还原像素」最难的地方。

**谁开始做的?** 分三件事说：

1. **坐标当 token 输出**：源头是 Pix2Seq（2021，Google）。OFA（2022）把它带进统一的多模态 seq2seq 模型，其作者 Peng Wang、An Yang、Junyang Lin、Shuai Bai、Jingren Zhou 后来都是 Qwen 的核心作者——**Qwen 系列的 grounding 可以一路追到 OFA**。
2. **编码器侧保留原生分辨率和位置**：NaViT（2023，Google）提出原生分辨率 ViT；Qwen2-VL 把 ViT 的绝对位置编码换成 2D-RoPE，并在 LLM 里用 M-RoPE。
3. **把坐标系和编码器像素网格绑在一起**：Qwen2.5-VL 改用绝对坐标 + 原生分辨率，是目前看到的最明确的「输出和编码器协同设计」。GUI-Actor 则反过来，质疑「用文本写坐标」本身，改成直接在补丁上做注意力选择。

Claude 的「数像素」没有公开架构，只能从效果上看到它解决了同一个问题。

> **一句话**:「输准像素」不是单纯的输出格式问题。坐标从量化 token（Pix2Seq、OFA、Kosmos-2）走到纯文本数字（Shikra、Qwen-VL），编码器从固定分辨率 + 绝对位置编码走到原生分辨率 + 2D-RoPE（NaViT、Qwen2-VL），两者在 Qwen2.5-VL 的「绝对坐标 + 28 像素对齐的 token 网格」上合流；GUI-Actor 则尝试干脆不写坐标。

> 我的复述

**自测**

1. Kosmos-2 的一个位置 bin 是多少像素？它和补丁大小是什么关系？
2. Qwen2.5-VL 为什么要求输入长宽是 28 的倍数？这和它改用绝对坐标有什么关系？
3. GUI-Actor 认为「用文本写坐标」有什么根本问题？它怎么绕开？

### 2. 从视觉 token 读出 token 内部的位置

- **Seg2Click（2026.10）**[59]：直接点名这个瓶颈——一个视觉 token 的覆盖区域可能同时包含目标、相邻控件和背景，却只得到一个分数；对分数做插值能提高输出分辨率，但**不会在 token 内部产生独立的预测**。做法：用一个以指令为条件的 pixel-shuffle 解码器，融合视觉特征和语言模型特征，**在处理后的截图分辨率上**输出「可点击区域」掩码，用标注框监督；最终点击取掩码最大连通区域的质心。Seg2Click-9B：ScreenSpot 92.06、ScreenSpot-v2 94.10、ScreenSpot-Pro 55.22，同底座下优于文本坐标和补丁打分两种做法。
- **Phi-Ground（Microsoft，2025.07）**[60] 从输出端做了系统实验：把坐标切成 1,000 个区间、每个区间一个新 token——大量未训练的新 token 让模型崩溃；给数字 token 加标签平滑（让「5」附近的「4」「6」也得一点分，近似回归损失）、按百位 / 十位 / 个位加权——小数据时有点用，batch 2048、百万级样本后没有效果；最终用「相对坐标 ×1000 的纯文本 + 普通交叉熵」。另一个发现：**先输入文字再输入图像**比反过来好 2–4 个点——因果掩码下，后输入的图像 token 才能「看到」指令，图像特征因此是带着任务去建模的。
- **Kimi-VL（2025.04）**[61]：MoonViT（400M）按原生分辨率输入，沿用 NaViT 的打包方式，ScreenSpot-Pro 34.5；技术报告里没有看到专门研究补丁内定位的实验。

### 3. 怎样逼模型把语言落到像素上：grounding 监督的通用做法

**问题**：输入无损只说明信息在 token 里，不说明模型会用。VQA、图像描述这类目标只考整体语义——答对「图里有几只猫」不需要指出猫在哪，模型没有压力把一句话归因到具体的 token、具体的像素（DeCo 说的「双重抽象」就是这种情况，见第四章第 4 节）。要让模型关注像素，监督里必须有**位置**。按监督放在哪一层，有五类做法：

| 层 | 做法 | 代表工作 |
| --- | --- | --- |
| 视觉编码器预训练 | 让编码器的特征能读出框 | GLIP（Microsoft，2021.12）：把检测和短语定位统一成一个预训练任务，27M 条定位数据，其中 3M 人工标注、24M 是网上图文对用自训练生成的伪框 [62]；LocCa（Google，2024.03）：让描述器从图像直接「读出」框坐标和区域描述，定位类下游任务明显好于普通描述器，整体任务持平 [63]，后来并入 SigLIP 2；PE-spatial 蒸馏 SAM 的掩码 |

| 层 | 做法 | 代表工作 |
| --- | --- | --- |
| VLM 预训练数据 | 文本里夹坐标 | Kosmos-2 的 GrIT：大规模「文字片段—框」对，[文字](框) 格式 [3]；Qwen-VL 的「图像—描述—框」三元组 [5]；Qwen2.5-VL 用绝对坐标的框和点做定位数据 [18]；Molmo 人工采集的 2D pointing 数据 [16]；GUI 上的 SeeClick / OS-Atlas / UGround（见 CUA 笔记） |
| 像素级输出 | 回答里直接带掩码 | GLaMM（2023.11）：生成的回答和对应物体的分割掩码交织在一起 [64]；GUI 上的 Seg2Click |
| 强化学习 | 可由规则判对的空间奖励 | GUI-R1 / GUI-G1 / GUI-G²（见 CUA 笔记第六章）；GroundingPI（2026.09）：4B 的通用定位模型，点和框都量化成共享词表里的坐标，训练是「多模态 + 空间预训练 → SFT → GRPO」；34 个定位基准平均 73.68%，高于 GPT-6 Astra 的 71.54%[65] |
| 推理时 | 把坐标写进思维链 | OpenAI o3（2025-04-16）：图像进入思维链，可以在推理中旋转、放大、裁剪图像 [66]；DeepSeek 的 Thinking with Visual Primitives（见下） |

**DeepSeek「Thinking with Visual Primitives」**

这篇论文 2026 年 4 月底在 GitHub 发布，约一天后被撤下，原地址已失效，官方没有说明原因；**本笔记没有读到原文**，以下来自两篇二手分析 [67][68]，其作者都声明没有复现结果。

- **底座**：DeepSeek-V4-Flash（MoE）+ DeepSeek-ViT；没有放出权重。
- **区分两种失败：感知缺口**（Perception Gap）——东西根本没看到；**指代缺口**（Reference Gap）——看到了，但在长推理里用语言描述它时丢了身份，比如数人时数到后面忘了谁已经数过、「后面那个杯子」在拥挤场景里指代不清。
- **做法**：把框和点当成「最小的思考单元」，在推理过程中和文字交替输出——框是物体的工作记忆，点序列是路径（如迷宫）的工作记忆。中间输出的框和点可以用规则判对错，所以能直接做 RL。
- **局限**（据二手报道）：需要显式的触发词才会启用；二手分析指出论文没有单独消融「视觉基元本身贡献了多少」。

和「输准像素」是什么关系：TVP 解决的是指代缺口，在推理层面——模型已经能定位，要让它在长推理中一直「抓住」那个位置；前面几节讨论的是感知和定位精度，在编码器和输出层面。两者是上下游：TVP 依赖模型先能输出准确的框和点（也就是上表前四类监督的成果）；反过来，它把框和点放进推理链，又给 RL 提供了更多可以判对错的中间空间监督。同样的「视觉基元」思路也出现在机器人动作模型里：AVP（2026.05）让 VLM 先输出视觉基元 token，再由动作专家据此生成动作 [69]。

**一句话**：只考整体语义的目标不会逼模型关注像素；要让它关注，就得在某一层放进位置监督——编码器特征读出框（GLIP、LocCa）、预训练文本里夹坐标（Kosmos-2、Qwen、Molmo）、输出掩码（GLaMM）、可判对错的空间奖励（GUI-G²、GroundingPI），或者在推理链里写坐标（o3、DeepSeek TVP）。TVP 处理的是「看到之后在长推理里别丢」，和定位精度是上下游关系。

我的复述

自测

1. 为什么只用 VQA 和图像描述数据训练，模型没有动力把答案归因到具体像素？
2. GLIP 的 24M 条伪框数据是怎么来的？
3. DeepSeek TVP 区分的「感知缺口」和「指代缺口」分别是什么？它主要解决哪一个？

## 附录：人与组谱系

机构取自论文首页署名；人名用论文英文署名。CUA 相关的组（OSU、CMU、HKU XLang、字节 Seed 等）见 CUA 笔记附录。

| 组 / 机构 | 核心人物（论文署名） | 工作（时间） |
|---|---|---|
| Apple | Haoxuan You、Zhe Gan、Yinfei Yang | Ferret（2023.10） |
| Google Brain Zürich → Google DeepMind | Lucas Beyer、Xiaohua Zhai、Alexander Kolesnikov、Andreas Steiner、Neil Houlsby | BiT（2019.12）→ ViT → Scaling ViT → LiT → SigLIP（2023.03）→ PaLI-3 → PaliGemma（2024.07）→ SigLIP 2（2025.02） |
| DeepMind | Jean-Baptiste Alayrac、Andrew Zisserman、Karen Simonyan | Flamingo（2022.04） |
| NVIDIA + MIT（Song Han） | Ji Lin、Hongxu Yin、Jan Kautz | VILA（2023.12） |
| Apple | Brandon McKinzie、Zhe Gan、Yinfei Yang | MM1（2024.03）；与 Ferret / Ferret-UI 同一圈（Zhe Gan、Yinfei Yang） |
| AI2（Ali Farhadi、Aniruddha Kembhavi、Ranjay Krishna） | Matt Deitke、Christopher Clark | Molmo / PixMo（2024.09）→ Molmo2（2026.01） |

| 组 / 机构 | 核心人物（论文署名） | 工作（时间） |
| --- | --- | --- |
| 上海 AI Lab 等（Jifeng Dai、Yu Qiao） | Zhe Chen、Weiyun Wang、Wenhai Wang | InternVL（2023.12）→ InternVL3（2025.04）→ InternVL3.5（2025.08） |
| MSRA（Furu Wei） | Li Dong、Shaohan Huang、Yiheng Xu | MarkupLM（2021.10）、BEiT（2021.06）、Kosmos-1（2023.02）、Kosmos-2（2023.06） |
| 百度（待核） | Haifeng Wang、Hua Wu | ERNIE 5.0（2026.02） |
| LLaVA-OneVision-2 团队（含 Ziwei Liu） | Xiang An、Yin Xie、Jiankang Deng | LLaVA-OneVision-2（2026.05） |
| Microsoft Research Redmond (Jianfeng Gao) | Chunyuan Li、Jianwei Yang、Hao Zhang、Feng Li | SoM（2023.10）；Chunyuan Li 同时是 LLaVA 共同作者 |
| UW–Madison（Yong Jae Lee） | Haotian Liu（+ Chunyuan Li） | LLaVA（2023.04）→ LLaVA-1.5（2023.10）→ LLaVA-NeXT（2024.01，博客） |
| ByteDance + NTU S-Lab（Ziwei Liu） | Bo Li、Yuanhan Zhang、Chunyuan Li、Feng Li、Hao Zhang | LLaVA-OneVision（2024.08） |
| Alibaba Qwen | Jinze Bai、Shuai Bai、Junyang Lin、Jingren Zhou | OFA（2022.02，Peng Wang、An Yang、Junyang Lin、Shuai Bai、Jingren Zhou 署名）→ Qwen-VL（2023.08）→ Qwen2-VL（2024.09）→ Qwen2.5-VL（2025.02） |
| Alibaba X-PLUG（+ 北京交通大学 Jitao Sang） | Haiyang Xu、Jiabo Ye、Ming Yan、Junyang Wang | mPLUG-Owl（2023.04）→ Mobile-Agent（2024.01） |
| 清华 + 智谱（Jie Tang） | Wenyi Hong、Weihan Wang、Ming Ding、Yuxiao Dong | CogVLM（2023.11）→ CogAgent（2023.12） |
| SenseTime + 北航 | Keqin Chen、Zhao Zhang | Shikra（2023.06） |
| **Salesforce Research / Salesforce AI Research** | Junnan Li、Dongxu Li、Steven Hoi、Caiming Xiong、Silvio Savarese | ALBEF（2021.07）[70] → BLIP（2022.01）[71] → BLIP-2（2023.01，Q-Former）[48] → InstructBLIP（2023.05，与港科大等合作）[72] → xGen-MM / BLIP-3（2024.08）[73] |

| 组 / 机构 | 核心人物（论文署名） | 工作（时间） |
| --- | --- | --- |
| **MSRA（Furu Wei）** | Yiheng Xu、Lei Cui、Li Dong、Shaohan Huang、Tengchao Lv、Yupan Huang | LayoutLM（2019.12，文本 + 版面预训练，一作 Yiheng Xu）[74] → LayoutLMv3（2022.04）[75] → MarkupLM（2021.10，HTML）→ BEiT（2021.06）→ BEiT-3（2022.08，「把图像当外语」）[76] → Kosmos-1（2023.02）→ Kosmos-2（2023.06，位置 token）→ Kosmos-2.5（2023.09，文档读写）[77] |

**两条线的特点与连线**

- **Salesforce 的 BLIP 线**走的是「冻结 / 复用强模型 + 轻量桥接」：ALBEF 先对齐再融合，BLIP 自举清洗网络图文数据，BLIP-2 用 Q-Former 做有意的信息瓶颈（第四章第 4 节），InstructBLIP 加指令微调；到 BLIP-3 已是另一批作者（Le Xue、Ran Xu 等）。**Caiming Xiong、Silvio Savarese 同时署名 OSWorld、Aguvis、Jedi、CUA-Gym 等 HKU XLang 的工作**——Salesforce 是 XLang 在 2023–2025 年的主要合作方（见 CUA 笔记附录）。
- **MSRA Furu Wei 组**走的是「读结构 / 文档版面」：从 LayoutLM 的文本 + 版面，到 MarkupLM 的 HTML，再到 BEiT 系列的掩码图像建模和 Kosmos 系列的多模态 LLM（Kosmos-2 的位置 token 是 grounding 输出格式的早期做法，第五章第 1 节）。**Yiheng Xu** 是 LayoutLM 一作、MarkupLM 二作；HKU XLang 的 Aguvis、Lemur、AgentTrek 一作也叫 Yiheng Xu，是否同一人待核。
- **Chunyuan Li**：LLaVA、LLaVA-1.5、SoM、LLaVA-NeXT、LLaVA-OneVision 都署名，横跨 Microsoft Research 与 ByteDance。
- **Zhe Gan**、**Yinfei Yang**（Apple）：Ferret → Ferret-UI → MM1。

**参考资料**

[1] Lee et al.：Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding

[2] Liu et al.：Visual Instruction Tuning（LLaVA）

[3] Peng et al.：Kosmos-2: Grounding Multimodal Large Language Models to the World

[4] Chen et al.：Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic

[5] Bai et al.：Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond

[6] Liu et al.：Improved Baselines with Visual Instruction Tuning（LLaVA-1.5）

[7] You et al.：Ferret: Refer and Ground Anything Anywhere at Any Granularity

[8] Hong et al.：CogAgent: A Visual Language Model for GUI Agents

[9] Yang et al.：Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V

[10] Liu et al.：LLaVA-NeXT: Improved reasoning, OCR, and world knowledge（2024-01-30）

[11] Li et al.: LLaVA-OneVision: Easy Visual Task Transfer

[12] Wang et al.: Mobile-Agent: Autonomous Multi-Modal Mobile Device Agent with Visual Perception

[13] Alayrac et al.: Flamingo: a Visual Language Model for Few-Shot Learning

[14] Lin et al.: VILA: On Pre-training for Visual Language Models

[15] McKinzie et al.: MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training

[16] Deitke et al.: Molmo and PixMo: Open Weights and Open Data for State-of-the-Art Vision-Language Models

[17] Wang et al.: Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution

[18] Bai et al.: Qwen2.5-VL Technical Report; 博客 Qwen2.5 VL（2025-01-26）

[19] Bai et al.: Qwen3-VL Technical Report

[20] Zhu et al.: InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models

[21] Wang et al.: InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency

[22] Wang et al.: ERNIE 5.0 Technical Report

[23] Clark et al.: Molmo2: Open Weights and Data for Vision-Language Models with Video Understanding and Grounding

[24] An et al.: LLaVA-OneVision-2: Towards Next-Generation Perceptual Intelligence

[25] Furu Wei 个人主页: thegenerality.com

[26] Huang et al.: Language Is Not All You Need: Aligning Perception with Language Models（Kosmos-1）

[27] Bao et al.: BEiT: BERT Pre-Training of Image Transformers

[28] Radford et al.: Learning Transferable Visual Models From Natural Language Supervision（CLIP）

[29] Sun et al.: EVA-CLIP: Improved Training Techniques for CLIP at Scale

[30] Zhai et al.: Sigmoid Loss for Language Image Pre-Training（SigLIP）

[31] Chen et al.: InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks

[32] Kolesnikov et al.: Big Transfer（BiT): General Visual Representation Learning

[33] Dosovitskiy et al.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale（ViT）

[34] Zhai et al.: Scaling Vision Transformers

[35] Zhai et al.: LiT: Zero-Shot Transfer with Locked-image text Tuning

[36] Chen et al.: PaLI-3 Vision Language Models: Smaller, Faster, Stronger

[37] Beyer et al.: PaliGemma: A versatile 3B VLM for transfer

[38] Tschannen et al.：SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features

[39] Purshow Notes：视觉编码器架构与多模态训练对比（2026-11 查看，二手整理）

[40] Purshow Notes：从 Infra 角度看，为什么去掉 Encoder 更好（2026-11 查看）

[41] Shi et al.：Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network（ESPCN，pixel shuffle）

[42] Li et al.：Monkey: Image Resolution and Text Label Are Important Things for Large Multi-modal Models

[43] You et al.：Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs

[44] Zheng et al.：Diffusion Transformers with Representation Autoencoders（RAE，§3）

[45] Rombach et al.：High-Resolution Image Synthesis with Latent Diffusion Models（LDM）

[46] Purshow Notes：视频生成模型 VAE（2026-11 查看，二手整理）

[47] Fang et al.：Data Filtering Networks（DFN）

[48] Li et al.：BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models（§3.1）

[49] Cha et al.：Honeybee: Locality-enhanced Projector for Multimodal LLM

[50] Yao et al.：DeCo: Decoupling Token Compression from Semantic Abstraction in Multimodal Large Language Models

[51] Chen et al.：An Image is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models（FastV）

[52] Chen et al.：Pix2seq: A Language Modeling Framework for Object Detection

[53] Wang et al.：OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework

[54] Dehghani et al.：Patch n' Pack: NaViT, a Vision Transformer for any Aspect Ratio and Resolution

[55] Cheng et al.：SeeClick: Harnessing GUI Grounding for Advanced Visual GUI Agents

[56] Anthropic：Developing a computer use model（2024-10-22）

[57] Qin et al.：UI-TARS: Pioneering Automated GUI Interaction with Native Agents

[58] Wu et al.：GUI-Actor: Coordinate-Free Visual Grounding for GUI Agents

[59] Wang & Huang：Seg2Click: Learning Pixel-Level Action Masks for GUI Visual Grounding

[60] Zhang et al.：Phi-Ground Tech Report: Advancing Perception in GUI Grounding

[61] Kimi Team：Kimi-VL Technical Report

[62] Li et al.：Grounded Language-Image Pre-training（GLIP）

[63] Wan et al.：LocCa: Visual Pretraining with Location-aware Captioners

[64] Rasheed et al.：GLaMM: Pixel Grounding Large Multimodal Model

[65] Yu et al.：GroundingPI: A Grounding Foundation Model towards Physical Intelligence with Visual Primitives

[66] OpenAI：Introducing OpenAI o3 and o4-mini（2025-04-16）

[67] Nagaonkar & Trivedi（VideoDB）：DeepSeek's visual primitives and the missing layer（2026-07-13，二手分析）

[68] Joche Ojeda：The Vision Model That Wasn't There: DeepSeek V4, a Vanished Paper, and a Recipe You Can Use Today（2026-06-24，二手分析）

[69] Guo et al.：Action with Visual Primitives（AVP）

[70] Li et al.：Align before Fuse: Vision and Language Representation Learning with Momentum Distillation（ALBEF）

[71] Li et al.：BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation

[72] Dai et al.：InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning

[73] Xue et al.：xGen-MM (BLIP-3): A Family of Open Large Multimodal Models

[74] Xu et al.：LayoutLM: Pre-training of Text and Layout for Document Image Understanding

[75] Huang et al.：LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking

[76] Wang et al.：Image as a Foreign Language: BEiT Pretraining for All Vision and Vision-Language Tasks（BEiT-3）

[77] Lv et al.：KOSMOS-2.5: A Multimodal Literate Model