

Visualn Grounging模块

三者关系图



好一个最终目的是这个视觉落地，认识类别和对应哪一个，这些都是最终能够将这个视觉落地的体现，这两个联合起来是真的牛逼呀

1 GroundingDINO（接地恐龙 🦖）

名字拆解

- **Grounding** = 落地（语言→图像坐标）
- **DINO** = DETR with Improved deNoising anch**O**r boxes（一种强大的目标检测架构）

这个DIDO和这个YOLO刚好一个是闭集一个是开放集



 **Figure 1-2 精读：整体架构速览****Figure 1：三种检测任务的对比**

任务类型	输入	输出	举例
Closed-Set Detection (封闭集检测)	图片	预定义类别的框	"检测所有 person, bench"
Open-Set Detection (开放集检测)	图片 + 新类别名	新类别的框	"检测 worldcup" (训练时没见过)
REC (指代表达理解)	图片 + 描述语句	唯一一个框	"The left lion" (左边那只狮子)


可以层层递进，越来越具有这个普遍性

场景 2：有 Self-Attention (互相看)

"螺丝" 问大家：谁在描述我？

- └ "红色" 举手：我！我是你的颜色属性！（相关度高 ✓）
- └ "左边" 举手：我！我是你的位置！（相关度高 ✓）
- └ "的" 说：我只是语法词（相关度低 ✕）

结果："螺丝" 的新表示 = 原来的"螺丝" + 一点"红色"的信息 + 一点"左边"的信息

 **关键洞察：**经过 Self-Attention 后，"螺丝"这个词的表示里已经融入了"红色"和"左边"的信息，变成了一个"知道自己是左边的红色螺丝"的表示。

 **数学上是怎么做的？**

自我注意力就是将这个建立文本之间的相互关系

一些概念的辨析

1) RefCOCO / RefCOCO+ / RefCOCOg 是什么？

它们都是“**指代表达数据集**” (Referring Expression Dataset)：

每条数据 = 一张图 + 一句话描述 + 这句话指向的那个目标框。

你可以理解成：在图里做“找茬”，但线索是自然语言。

RefCOCO（允许位置词）

- 特点：描述里可以出现位置关系词，比如 *left/right/front/behind*
- 例子：boy on left（左边的男孩）
- 难度直觉：相对更容易，因为“左边”这类词很强的定位线索。

对工业场景：像“左边那颗螺丝”“靠近夹爪的那个零件”。

RefCOCO+（禁止位置词）

- 特点：描述里不能用位置词（不让你靠“左边/右边”这种捷径）
- 例子：a kid in blue shirt（穿蓝衬衫的小孩）
- 难度直觉：更难，模型必须靠外观属性（颜色、材质、形状、部件）来选对。

对工业场景：像“红色螺丝”“带二维码标签的盒子”，不能说“左边那个”。

RefCOCOg（描述更长、更啰嗦）

- 特点：句子通常更长、信息更丰富，可能包含多个属性和关系
- 例子：the boy wearing a black helmet standing at bat（戴黑头盔、站在球棒位置的男孩）
- 难度直觉：不是“更难或更易”固定的——线索多了有时更好找，但也更考验模型把长句拆成关键约束。

对工业场景：像“靠近夹具、带黑色橡胶圈、旁边有一根气管的那个接头”。

2) val / testA / testB 是什么？

它们都是数据集切分（split），用来公平评估：

- **val（验证集）**：训练时用来调参、看趋势（不参与最终“成绩”）
- **test（测试集）**：最终报告成绩用

RefCOCO 里常见：

- **testA**：偏“人”相关的样本更多（你截图里也写了人物场景）
- **testB**：偏“物体”相关的样本更多

对你更关键的是 testB 的意义：更贴近工业“物体/零件”场景。

3) Zero-shot 是什么？

中文：零样本 / 零训练迁移

意思是：不在这个数据集上训练（比如不在 RefCOCO 上训练），直接拿模型去做 RefCOCO 测试。

直觉：

- 像你从没刷过“指代题”，只刷过“检测题”，现在直接去做“根据一句话找那个物体”的题——一般会吃亏。
- 这能测模型的“泛化能力”：到底是不是“见过才会”。

4) Fine-tune 是什么？

中文：微调

在目标数据集（比如 RefCOCO）上**继续训练一小段**，让模型适应这种任务的“题型”。

直觉：

- 像做题前先刷一套同类型真题，模型就学会：“这类句子里，颜色/位置/关系词分别怎么用来选目标”。

5) SOTA 是什么？

中文：当前最好水平（State Of The Art）

就是“在这个榜单/任务上目前最强的结果”。

6) 这些概念和你“模糊指令消歧”怎么挂钩？

你要做的“消歧”通常分两步：

1. **先 grounding 出候选**（可能不止一个）：这更像 RefCOCO/RefCOCO+ 的能力
2. **再通过追问/多模态信息选对那个**（你的研究重点）

所以你截图里“zero-shot 一般、fine-tune 提升巨大”的含义很现实：

只靠通用检测训练，模型对“细粒度指代”（那个/哪里/靠近谁）会不够稳；有指代数据或类似监督就会显著变强。

Image-to-Text Cross-Attention (图像去"问"文本)

图像中每个区域都在问："我是什么？文本里有描述我吗？"

红色螺丝区域 (Query) → 去文本里找 (Key/Value)

- └─ "红色" (Key): 相似度高! ✓
- └─ "螺丝" (Key): 相似度高! ✓
- └─ 结果: 这个区域融合了"红色"和"螺丝"的语义信息

蓝色螺丝区域 (Query) → 去文本里找 (Key/Value)

- └─ "红色" (Key): 相似度低 ×
- └─ "螺丝" (Key): 相似度高! ✓
- └─ 结果: 只融合了"螺丝"的信息, 但不是"红色的"

扳手区域 (Query) → 去文本里找 (Key/Value)

- └─ "红色" (Key): 相似度低 ×
- └─ "螺丝" (Key): 相似度低 ×
- └─ 结果: 几乎不融合任何信息

互相交叉融合, 是一种看着这个prompt的逐渐的融合, 那么在此基础上我们使用这个cross-attention就变得很合理了

有意思, 这个cross其实是这个跨模态的意识

好一个编码器和解码器encoder和decoder

3) 你可能会问: 为什么用 Max, 而不是平均?

很关键!

- **Max 的意义: OR (或) 逻辑**

只要像任意一个关键词就行 (red 或 screw) 。

- 如果用 mean (平均), 反而会把很多"只匹配其中一个关键词"的位置压低分数, 容易漏掉。

这很符合 grounding 的需求: **宁可多选一点候选 (召回高), 后面再精筛。**

这样确实可以保留更多的信息, 而且计算量和正确率的综合性最好, 每一个图像最像文本信息的图像位置的全部标出, 真是精彩

3.2 语言引导的查询选择 (Language-Guided Query Selection)

问题：图像有上万个位置 (token)，不可能都送进解码器，怎么办？

解决方案：用文本来"筛选"最相关的 900 个位置

核心公式：

$$I_{N_q} = \text{Top}_{N_q}(\text{Max}^{(-1)}(X_I X_T^\top))$$

逐步拆解：

1. $X_I \in \mathbb{R}^{N_I \times d}$ ：图像特征， N_I 个位置，每个位置 d 维
2. $X_T \in \mathbb{R}^{N_T \times d}$ ：文本特征， N_T 个词，每个词 d 维
3. $X_I X_T^\top$ ：计算每个图像位置和每个文本词的相似度（点积）
4. $\text{Max}^{(-1)}$ ：对每个图像位置，取它和所有文本词的最大相似度
5. Top_{N_q} ：选出相似度最高的 $N_q = 900$ 个位置

由语言引导的这个查询选择

你现在只要牢牢记住这句话就够了

语言引导的 query 选择 = 用文本给每个图像位置打分，挑出最可能相关的 900 个位置，减少计算，同时尽量不漏掉目标。

3. ? 部件关系消歧 (待验证)

论文未直接测试的场景：

- "那个连接着红色零件的螺丝"
- "拧在孔里的那颗" (vs 放在旁边的)

对工业场景的启示：

- 这类"部件间关系"的描述在工业场景很常见
- RefCOCO 数据集主要是日常场景，缺少这类工业装配关系
- 这可能是你课题的一个切入点：构建工业场景的指代消歧数据集

确实，如果部件和部件连接在一起，它又很难理解了

问题 1：Open-set Detection 和 REC 的区别？

对比维度	Open-set Detection（开放集检测）	REC（指代表达理解）
输入文本	类别名："screw. bolt. wrench."	描述语句："左边那个红色螺丝"
输出数量	所有符合类别的框	唯一一个框
核心挑战	识别没见过的新类别	从多个同类物体中消歧
举例	输入"螺丝" → 输出 3 个螺丝的框	输入"左边那个红色螺丝" → 只输出 1 个框

一句话区分：

Open-set Detection 问的是 "图里有哪些螺丝？"（找所有）
REC 问的是 "你说的是哪个螺丝？"（找唯一）

从数量上来看，两者还有区别，我觉得这个理解不精确，这个指代表达理解，应该是这个更加根据文本精确的定位召回，落地阶段grounding

Day1 你应达到的直觉

- “检测 vs 指代落地”差别要说得清：
 - 检测：回答“有哪些东西”
 - 指代落地：回答“你说的是哪一个”

检测和指代落地之间的关系，检测是找到这个全部，指代落地是找到具体是哪一个