> 三个基准测试（Benchmark）：
>
> - **SlideVQA**：**逻辑推理**专项考试，考察在结构化文档（如PPT）中的深度理解和推理能力。
> - **ViDoSeek**：**信息检索与定位**专项考试，考察在海量文档中“大海捞针”并找到唯一答案的能力。
> - **MMLongBench**：**视觉感知与长文本理解**综合考试，考察在复杂的长文档中处理多种视觉元素的能力。

Benchmark	核心目标	测试的关键能力	对VRAG-RL的意义
SlideVQA	理解结构化的PPT幻灯片	布局理解、**多跳逻辑推理**	检验智能体的**推理能力**
ViDoSeek	在海量文档中检索并回答	**高效准确的检索**、定位信息	检验智能体的**RAG检索效率**
MMLongBench	理解长篇幅、多模态文档	长上下文、**多样的视觉感知**（图表、表格等）	检验其`<region>`**视觉感知动作**的有效性

二、方法论

作者采用了一套基于强化学习的综合方法来解决上述问题，该方法被称为**VRAG-RL**。

其详细步骤和设计如下：

1. 整体框架：迭代推理与强化学习

论文将VLM与环境的交互建模为一个迭代推理和工具调用的过程，如**图1(a)** 所示。模型（Policy Model）根据历史交互轨迹 H_{t-1} 生成“思考”和“动作” A_t，与外部环境（包括搜索引擎和视觉感知功能）交互后获得“观察” O_t。

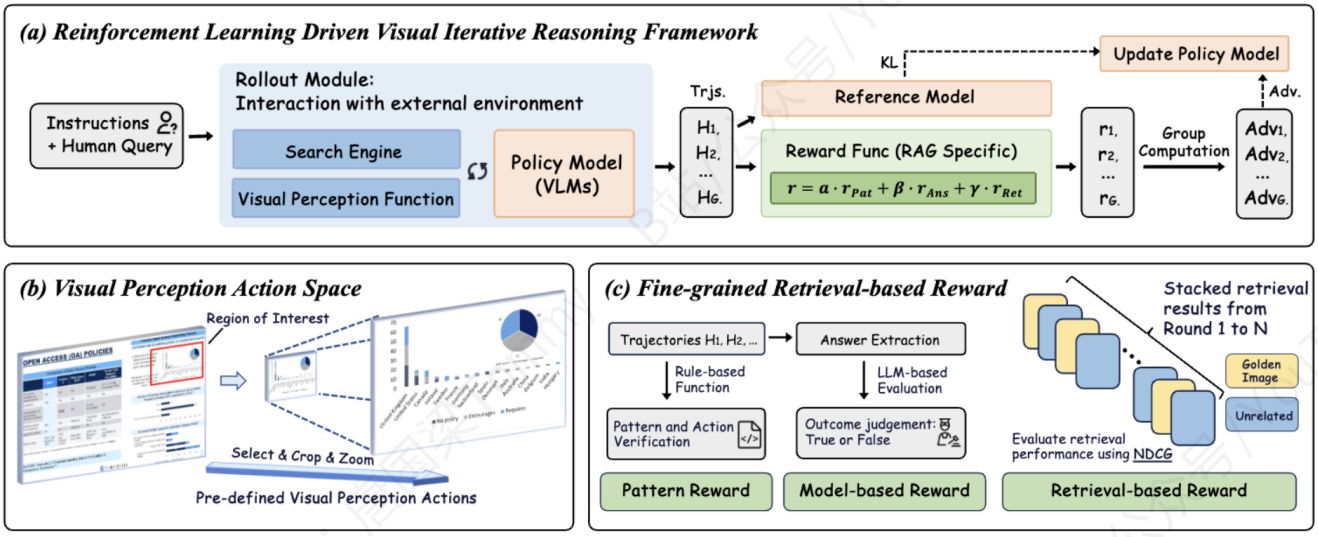
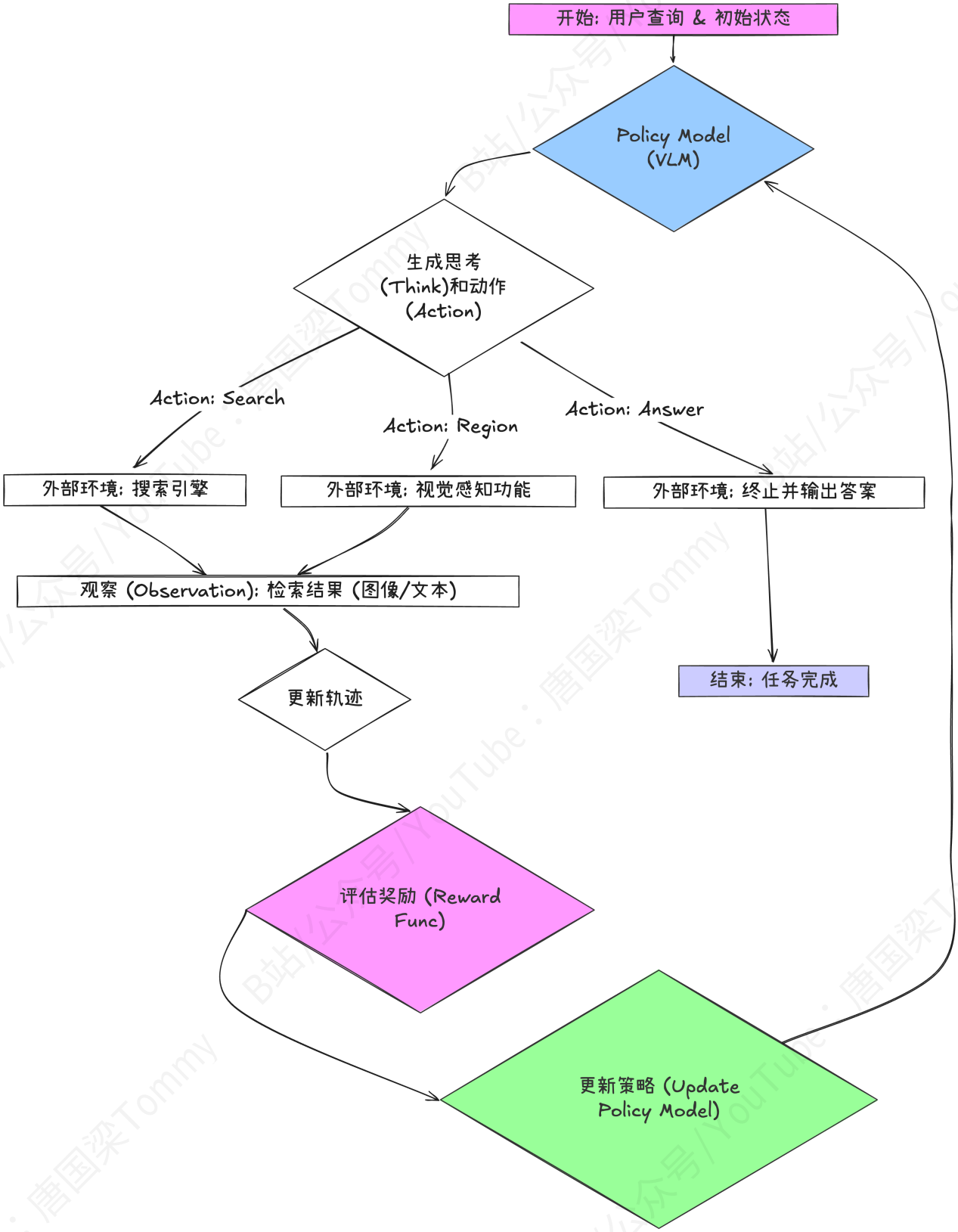


图 1：**强化学习框架总体结构**。(a) 展示了模型与外部环境之间的交互过程，以及 GRPO 算法的实现。(b) 展示了所提出的视觉感知动作空间，使模型能够以粗到细的方式提取信息。(c) 是为 RAG 特别设计的奖励机制，它在整个采样过程中综合考虑了结果和检索性能。



知识补充：

> GRPO(**分组相对策略优化**) 是一种强化学习算法，属于**策略梯度（Policy Gradient）**方法的一种。

它的核心目标是直接优化模型（即"策略"）的行为，使其能够产生更高奖励的输出。

它的核心思想是：**我们不需要知道一个回答的绝对分数有多高，只需要知道在一组候选回答中，哪个回答比其他回答更好。**

GRPO 将这个"分组比较"的思想流程化，具体步骤如下（以 VRAG-RL 为例）：

1. **分组采样**
   对于同一个输入问题，让当前的 VLM 模型（策略 π_θ）独立地生成 **G** 个不同的回答轨迹（在论文中，作者设置 G=5）。一个"轨迹"包含了模型完整的多轮交互，比如所有的搜索查询、视觉感知动作和最终答案。这样我们就得到了一个包含G个不同解决方案的"组"。
2. **评估与奖励**
   使用论文中定义的那个综合奖励函数 ($r_p = \alpha \cdot r_{Ret} + \beta \cdot r_{Ans} + \gamma \cdot r_{Pat}$)，为组内的**每一个**轨迹计算一个奖励分数。
3. **计算基线和优势**
   这是 GRPO 的精髓所在。
   - **基线 (Baseline)**：它的基线不是一个需要学习的复杂函数，而就是这 **G** 个轨迹奖励分数的**平均值** ($\bar{R}$)。这个基线是动态的、无偏的，并且非常容易计算。
   - **优势 (Advantage)**：对于组内的每一个轨迹 i，它的优势 A_i 就是它的奖励 R_i 与基线（平均奖励 $\bar{R}$）的差值：$A_i = R_i - \bar{R}$。
     - 如果 $A_i > 0$，说明这个轨迹比本次采样的平均水平要好。
     - 如果 $A_i < 0$，说明这个轨迹比平均水平要差。
4. **策略更新 (Policy Update)**
   算法根据计算出的 **优势** 来更新模型的参数。更新的原则是：
   - 对于有**正优势**的轨迹，**提高**模型生成这个轨迹的概率。
   - 对于有**负优势**的轨迹，**降低**模型生成这个轨迹的概率。

   这样，通过一轮又一轮的 **分组采样-评估-比较-更新** ，模型就会逐渐学会生成那些能够获得相对更高奖励的回答。

---

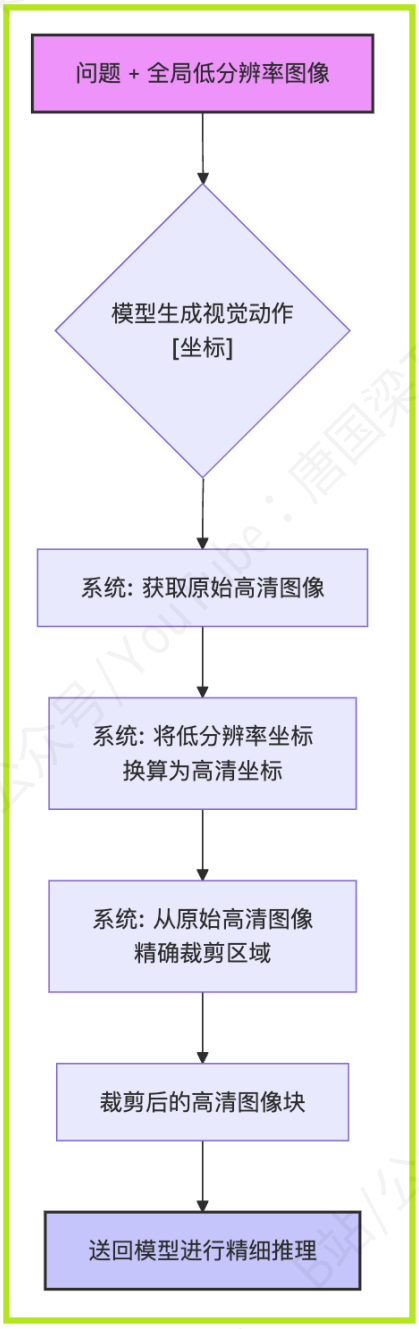
2. 视觉感知动作集成

为了让模型能够从粗到细地分析图像，作者定义了一个新的视觉感知动作空间。

2.1 实现流程

1. **全局侦察**：
   模型首先接收一张低分辨率的全局图像，并分析判断出包含关键信息的区域（例如一个表格或图表），但因模糊而无法读取细节。

2. **发出指令**：
   模型生成一个特定的视觉动作指令，格式为 `<region> [坐标] </region>`，像用手指一样，精确地告诉系统它想“放大”查看的区域。
3. **后台裁剪**：
   系统截获此指令，并将模型给出的低分辨率坐标，换算到**原始高清图像**上，然后从高清原图上精确地裁剪下该区域。
4. **聚焦推理**：
   这个新裁剪出的、清晰的高清图像块，作为全新的视觉信息被送回给模型。模型基于这个“特写镜头”完成最终的细节读取和推理，从而得出答案。



2.2 公式解析

- 模型可以生成特定的视觉感知Token `<region> [x_{min}, y_{min}, x_{max}, y_{max}] </region>`。
- 通过一个基于规则的函数提取出这个动作和坐标。如公式所示，系统会根据指定的边界框，从原始高分辨率图像 I_{raw} 中裁剪出感兴趣的区域 $\hat{R}$：

$$\hat{R} = \text{Crop}(I_{raw}, [x_{min} \times \frac{w_{raw}}{w_{encoder}}, y_{min} \times \frac{h_{raw}}{h_{encoder}}, \dots]) \tag{1}$$

> 作用是：**将模型在低分辨率图上给出的坐标，换算成高清原图上的坐标，然后进行裁剪。**
>
> 我们来拆解公式中的每一个部分：
>
> - $\hat{R}$ **(R-hat)**：
> - 最终裁剪出来的、清晰的图像区域。这就是模型将要收到的高清特写。
> - $\text{Crop}(\dots)$：
> - 一个函数，表示 **裁剪** 这个动作。
> - I_{raw}：
> - 原始的、高分辨率的图像文件。比如一张 3000x4000 像素的扫描文档。
> - $[x_{min}, y_{min}, \dots]$：
> - 模型在其看到的**低分辨率图像**上指定的坐标。
> - $w_{encoder}, h_{encoder}$：
> - 模型看到的低分辨率图像的宽度（width）和高度（height）。这是VLM的视觉编码器（vision encoder）能处理的固定尺寸，比如 448x448 像素。
> - w_{raw}, h_{raw}：
> - 高清原图 I_{raw} 的宽度和高度。例如，3000x4000 像素。
> - $\frac{w_{raw}}{w_{encoder}}$ **和** $\frac{h_{raw}}{h_{encoder}}$：
> - 这是整个公式的**核心——缩放比例因子**。
> - **解释**：它计算了原图尺寸是模型输入尺寸的多少倍。例如，如果原图宽度 $w_{raw} = 3000$ 像素，模型输入宽度 $w_{encoder} = 448$ 像素，那么宽度缩放比例就是 $3000/448 \approx 6.7$。
> - **作用**：这个比例因子用来将低分辨率图上的坐标“投射”回高分辨率原图上。

- 这个被裁剪出的高分辨率区域 $\hat{R}$ 会被重新编码并作为新的观察 O_t 输入到模型中。如**图2**所示，这种“裁剪并重新输入”的策略，有效提高了模型对信息密集区域的感知分辨率。

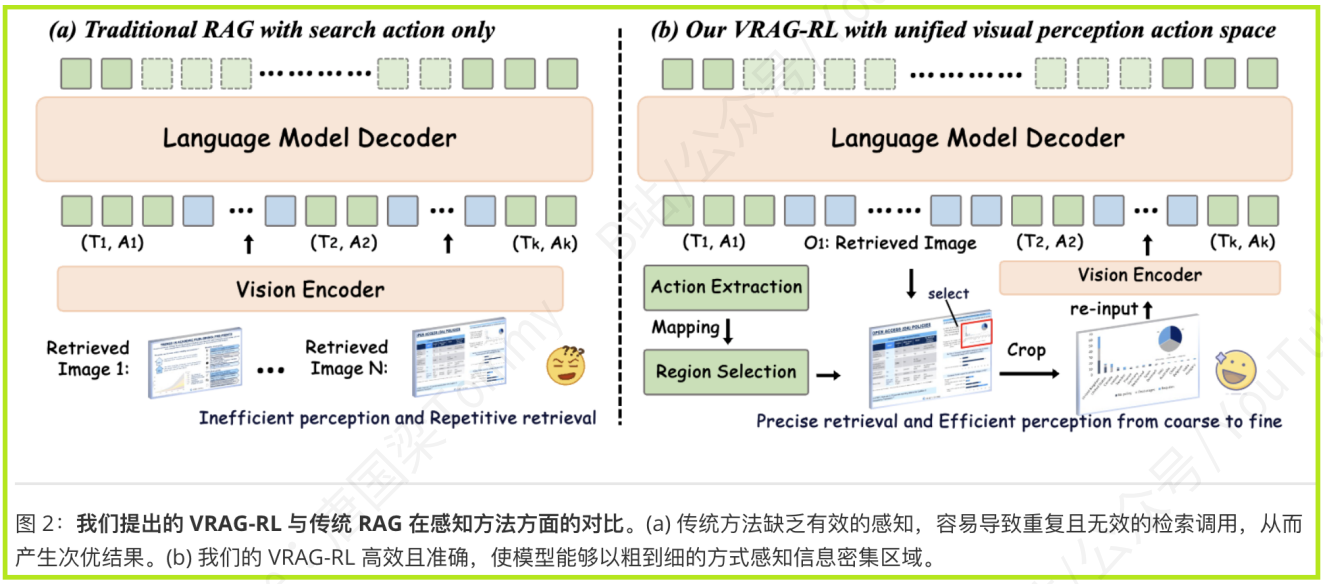


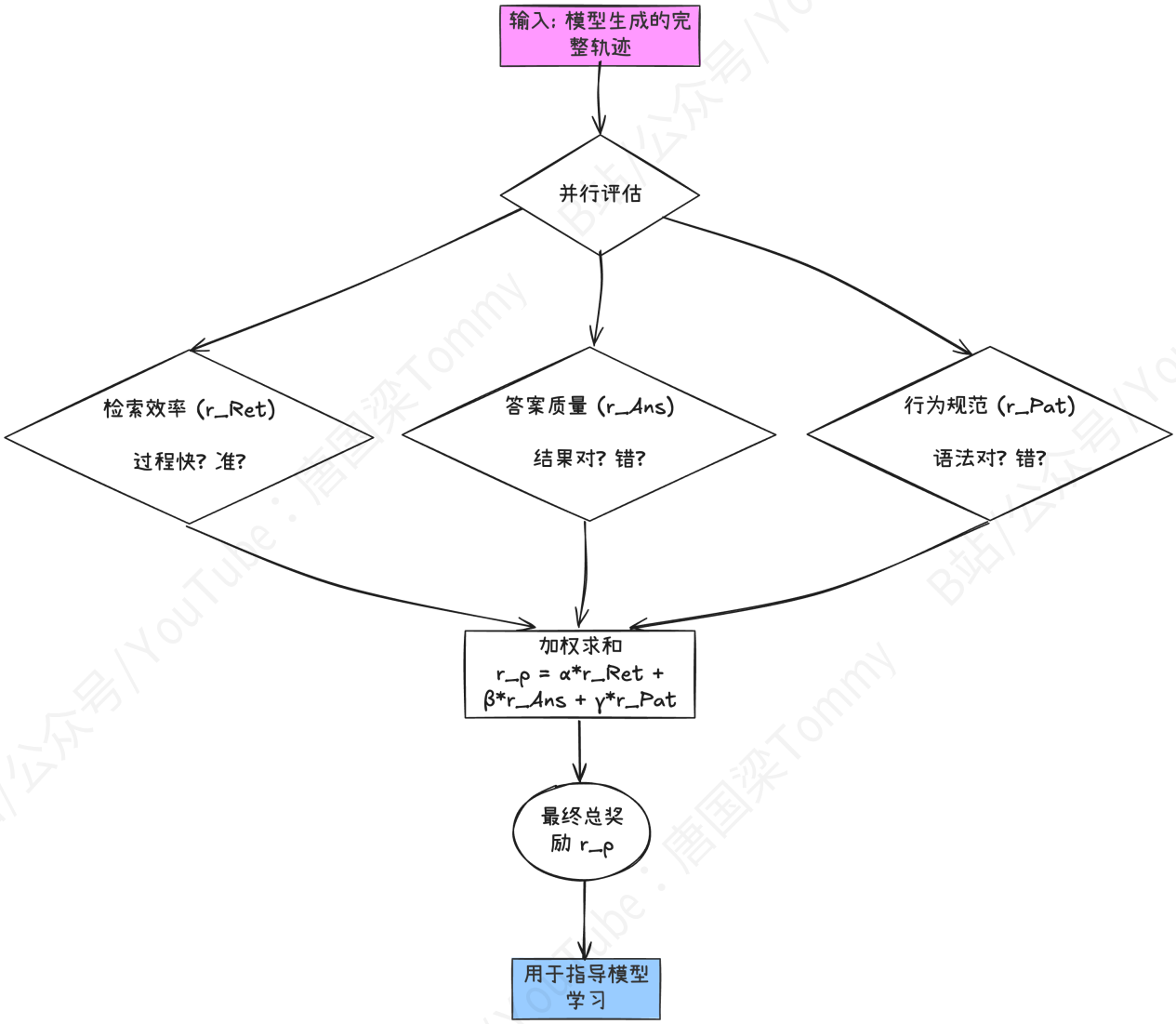
图 2：**我们提出的 VRAG-RL 与传统 RAG 在感知方法方面的对比。**(a) 传统方法缺乏有效的感知，容易导致重复且无效的检索调用，从而产生次优结果。(b) 我们的 VRAG-RL 高效且准确，使模型能够以粗到细的方式感知信息密集区域。

3. 细粒度的奖励函数

为了给RL训练提供更精确的指导，作者设计了一个由三部分组成的综合奖励函数，如公式所示：

$$r_p = \alpha \cdot r_{Ret} + \beta \cdot r_{Ans} + \gamma \cdot r_{Pat} \tag{2}$$

这里的 α, β, γ 是权重，用来调节我们对不同方面的重视程度。



3.1 r_{Ret} (检索效率奖励)

- **目的**：这个奖励专门用来评估模型**检索信息的好坏**。它鼓励模型**尽早、准确地**找到所有相关的“黄金”文档，同时避免检索不相关的“垃圾”文档。
- **为什么重要？**：论文在**图3**中证明了一个关键点：检索到的信息并非越多越好。过多的无关信息会形成上下文噪音，干扰模型的最终推理，导致性能下降。因此，一个高效的检索过程至关重要。

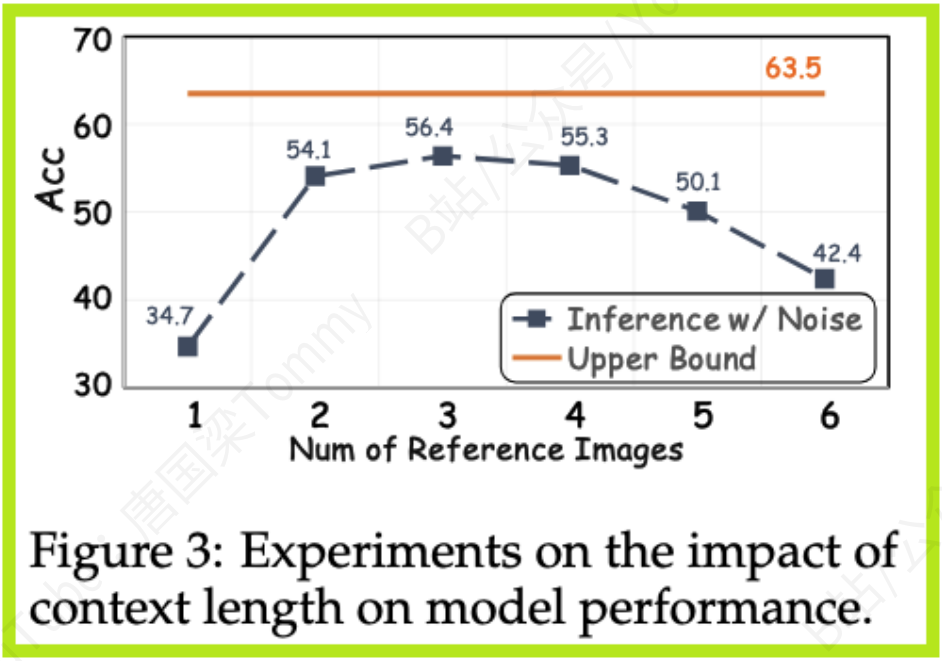


Figure 3: Experiments on the impact of context length on model performance.

- **如何计算?**：它受信息检索领域的经典指标 **NDCG (Normalized Discounted Cumulative Gain)** 的启发。
  - **Discounted (折损)**：意思是，**越早**找到相关文档，得分越高。第一轮就找到，比第五轮才找到的奖励要大得多。
  - **Cumulative Gain (累计增益)**：每找到一个相关文档，总分就增加。
  - **Normalized (归一化)**：将模型实际得到的 **累计折损增益（DCG）**，与一个"理想模型"在最佳情况下（即第一轮就找到所有相关文档）能得到的最高分（IDCG）进行比较。

$$r_{Ret} = \frac{\mathrm{DCG}(\text{模型的检索路径})}{\mathrm{IDCG}(\text{理想的检索路径})} \tag{3}$$

  - 这个比值在0到1之间，直观地反映了模型的检索效率有多接近完美。

一句话总结 r_{Ret}：它奖励那些 **快、准、狠** 的检索行为，惩罚那些 **拖沓、盲目** 的检索行为。

3.2 r_{Ans} (基于模型的成果奖励)

- **目的**：评估模型生成的**最终答案是否正确**。这是整个任务最核心的目标。
- **为什么重要?**：这是模型行为的最终落脚点，直接关联任务的成功与否。
- **如何计算?**：传统方法常用 **精确匹配（EM）**，即模型的答案必须和标准答案一字不差，这对于复杂的开放性问题来说过于死板。
  - VRAG-RL采用了一种更智能的方式：**让另一个强大的VLM（论文中使用Qwen2.5-7B-Instruct）来充当裁判**。
  - 这个裁判模型会同时读取**原始问题、标准答案和模型生成的答案**，然后像人类专家一样，从语义上判断模型生成的答案是否正确，并给出一个分数（比如1代表正确，0代表错误）。

- 这种方法更加灵活和鲁棒，能理解同义词、不同表述方式等，更接近真实世界的评判标准。

一句话总结 r_{Ans}：它用一个 **AI专家** 来评判最终答案的质量，确保模型学习的目标是生成语义上正确的答案。

3.3 r_{Pat} (模式一致性奖励)

- **目的**：鼓励模型遵循预定义的**交互模式和语法**。
- **为什么重要?** ：在强化学习的初期（冷启动阶段），模型可能会生成一些完全不合语法的无效动作，比如缺少 `</search>` 标签，或者在思考不完整时就强行输出答案。这会导致整个交互流程崩溃。
- **如何计算?** ：这是一个非常简单的、基于规则的奖励。系统会检查模型生成的轨迹，看它是否正确地使用了 `<search>`，`<region>`，`<think>` 等标签。如果遵循了规范，就给予少量奖励；如果没有，则不给奖励或给予惩罚。
- **补充说明**：这个奖励在训练后期通常就不那么重要了，因为模型已经学会了基本的交互语法。所以论文提到，在训练后期会把它的权重 γ 设为0或一个很小的值。

一句话总结 r_{Pat}：它像一个教官，确保模型在学会高级技能之前，先掌握了最基本的行动规范。

---

4. 基于多专家采样的轨迹数据扩充

4.1 为什么需要"多专家采样"?

在训练VRAG-RL模型时，尤其是在正式的强化学习（RL）阶段之前，通常需要一个**监督微调（Supervised Fine-Tuning, SFT）**阶段。

在这个阶段，模型学习的是模仿“专家”给出的标准答案。我们需要向模型展示大量**专家是如何一步步解决问题的范例**。

这些范例就是所谓的“轨迹数据”（Trajectory Data），一个轨迹包含了从问题到最终答案的完整思考和行动链，例如：

`问题 -> <think>思考1...</think> <search>搜索查询1</search> -> 观察1(图片) -> <think>思考2...</think> <region>区域坐标</region> -> 观察2(裁剪后的图片) -> <think>思考3...</think> <answer>最终答案</answer>`

这里的挑战在于：

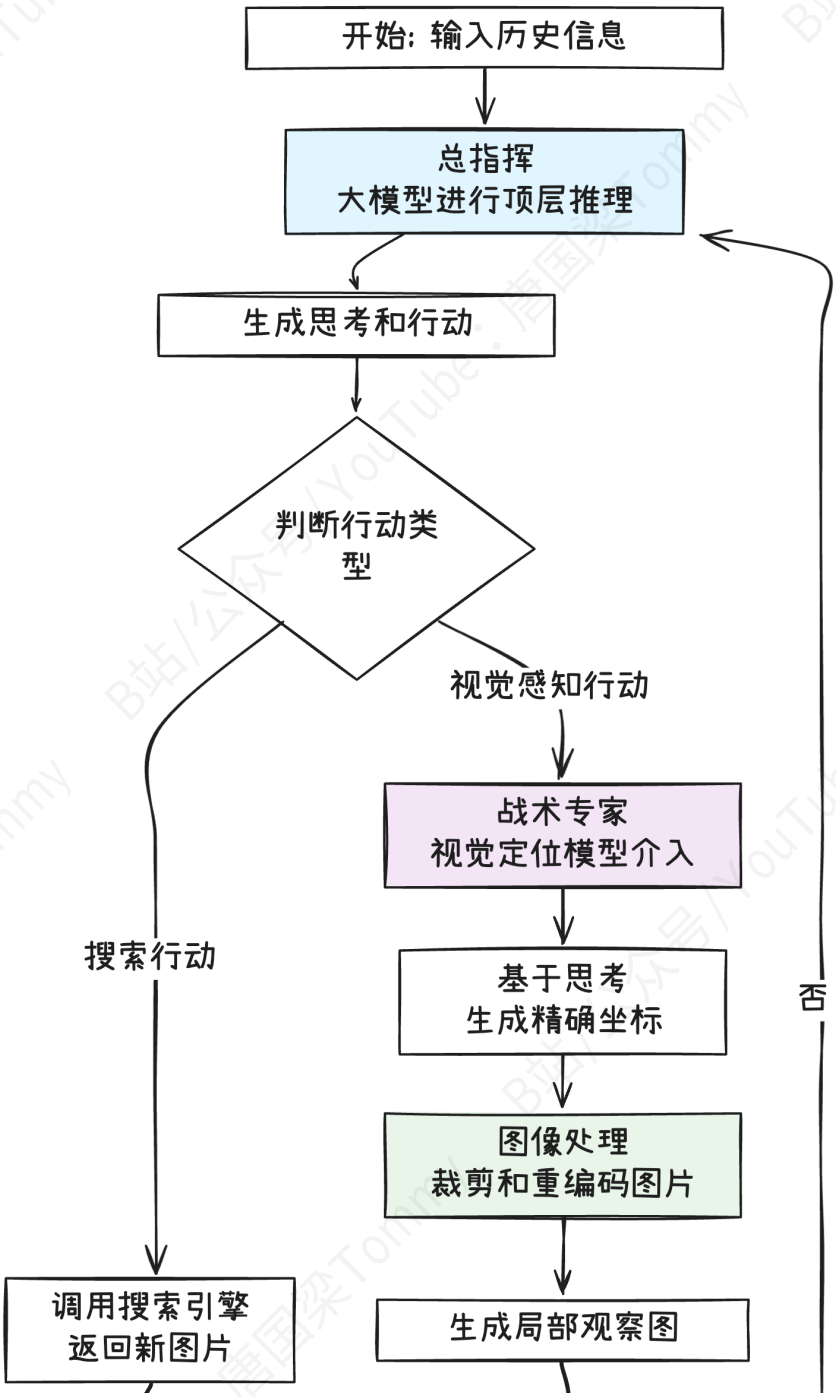
- **人力成本高**：让人类专家来标注成千上万条这样复杂的轨迹，既耗时又昂贵。
- **单一AI专家的局限性**：如果只用一个最强的AI模型（比如GPT-4V）来自动生成这些轨迹，可能会遇到问题。这个最强的模型可能在**宏观战略**（比如决定下一步是该搜索还是该回答）上很强，但在某些**微观操作**（比如在图片上精确地框出表格的位置）上未必是最佳选择，或者说成本很高。

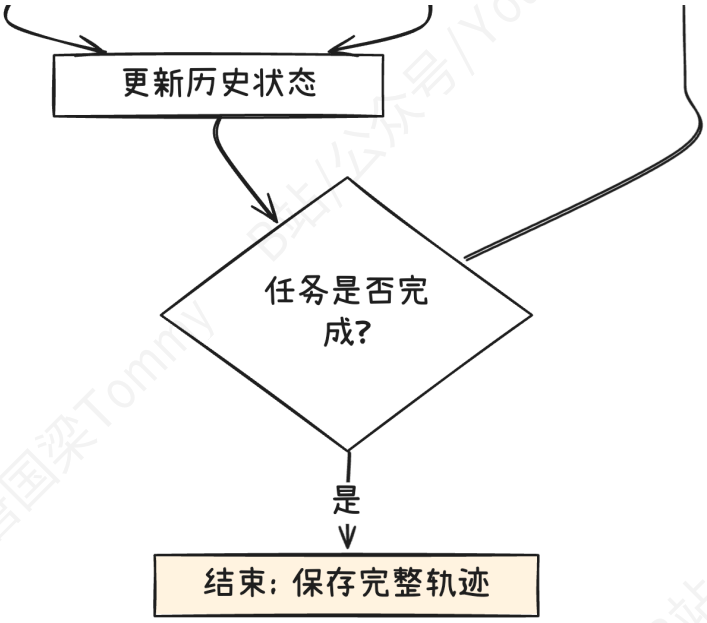
因此，论文提出了 **多专家采样** 的策略来自动化地生成这些高质量的轨迹数据。

4.2 “多专家采样”的核心流程

这个策略的精髓在于 **分工合作** ，让不同的 **AI专家** 在它们各自擅长的环节上发挥作用。

整个流程可以分解为以下几步：





第1步："总指挥" 制定宏观计划

- **专家角色**：一个规模巨大、推理能力超强的旗舰级VLM。论文中提到的是 **Qwen-VL-max-latest**。
- **任务**：这个总指挥负责制定解决问题的**高层战略**。它来决定整个推理的流程，比如：
  - 第一步应该进行一次宽泛的搜索。
  - 根据搜索结果，下一步应该进行一次更精确的搜索。
  - 现在信息足够了，应该对某张图片进行区域裁剪。
  - 最后，基于所有信息生成答案。
- 它会生成一个包含思考（`<think>`）和动作意图（`<search>`或`<region>`）的**初步轨迹**。但是，当它生成`<region>`动作时，它提供的坐标**可能并不完美**。

第2步："战术专家"执行精细操作

- **专家角色**：一个在特定任务上经过优化的、更专注的模型。论文中提到的是 **Qwen2.5VL-72B**，这个模型在**视觉定位（Grounding）**任务上非常强大。
- **任务**：每当总指挥在轨迹中生成了一个`<region>`动作时，战术专家就会介入。它的任务非常明确：
  - 读取总指挥的思考过程（即`<think>`标签里的内容），理解它到底想看图上的哪个部分。
  - 利用自己强大的视觉定位能力，在图像上生成一个**更精确、更完美的边界框（Bounding Box）坐标**。

第3步：数据整合与生成最终轨迹

- 系统会将总指挥生成的宏观计划，与战术专家提供的精确坐标进行**替换和整合**。
- 也就是说，初步轨迹中那个可能不太准的`<region>`坐标，会被替换成由定位专家给出的高质量坐标。

- 这样，一条由多个专家协作完成的、既有清晰战略逻辑又有精确战术执行的**高质量训练轨迹**就诞生了。

三、总结

作者得出的主要结论是，他们提出的**VRAG-RL框架能够有效地使视觉语言模型（VLM）与搜索引擎进行交互，从而显著增强其在处理富视觉信息时的推理和检索能力**。

这些结论是基于充分且有力的证据的：

- **压倒性的实验结果**：在多个具有挑战性的基准测试上，VRAG-RL相比所有基线模型都取得了巨大性能提升（如**表1**所示），这是最直接的证据。

Table 1: **Main Results.** The best performance are marked in bold. SlideVQA and ViDoSeek mainly focus on reasoning type, while MMLongBench focuses on the visual type of reference content. OCR-based (👁‍🗨) RAG and purely visual (👁) RAG are evaluated with the same prompt and setting.

Method	SlideVQA		ViDoSeek		MMLongBench					Overall
	Single-hop	Multi-hop	Extraction	Logic	Text	Table	Chart	Figure	Layout	
Qwen2.5-VL-3B-Instruct										
👁‍🗨 Vanilla RAG	15.1	12.1	8.8	14.3	3.9	5.1	1.7	3.1	2.5	11.2
👁‍🗨 ReAct	11.8	9.9	5.3	7.4	6.5	3.7	3.9	5.2	2.5	8.4
👁‍🗨 Search-R1	17.5	13.8	13.3	20.7	3.4	3.2	4.5	4.1	6.8	14.1
👁 Vanilla RAG	19.4	12.2	10.1	17.3	2.2	4.1	5.2	4.7	4.3	13.2
👁 ReAct	15.7	10.9	6.7	14.2	2.7	3.6	3.4	3.1	5.1	10.9
👁 Search-R1-VL	26.3	20.1	20.1	29.8	8.5	7.8	7.9	9.3	7.6	21.3
👁 **VRAG-RL**	**65.3**	**38.6**	**63.1**	**73.8**	**22.7**	**16.1**	**21.9**	**21.4**	**19.5**	**53.5**
Qwen2.5-VL-7B-Instruct										
👁‍🗨 Vanilla RAG	26.1	10.6	24.7	30.9	8.5	5.4	11.7	4.4	3.3	20.9
👁‍🗨 ReAct	21.2	13.3	14.3	21.3	5.9	5.1	7.3	5.5	1.7	15.8
👁‍🗨 Search-R1	28.4	19.7	20.8	30.6	9.9	6.0	7.9	10.1	5.9	22.2
👁 Vanilla RAG	29.1	17.4	26.4	41.3	13.1	14.7	15.9	4.3	7.6	24.2
👁 ReAct	34.8	20.4	27.5	42.1	10.1	12.4	10.2	6.2	7.1	26.9
👁 Search-R1-VL	48.3	42.3	40.5	50.3	19.9	13.4	12.9	11.4	10.2	37.4
👁 **VRAG-RL**	**69.3**	**43.1**	**60.6**	**74.8**	**26.1**	**26.3**	**24.8**	**25.9**	**21.2**	**57.1**

- **严谨的消融研究**：通过逐一移除核心创新点（视觉感知动作、RAG特定奖励）并观察性能下降（如**表2**所示），论文令人信服地证明了其成功的关键归因于这些特定的方法设计，而非其他偶然因素。

Table 2: Ablation study on three benchmarks.

Reward		Action Space		Accuracy
Vanilla	RAG-Specific	Search	Visual-Perception	
✓		✓		47.2
✓		✓	✓	49.3
	✓	✓		54.9
	✓	✓	✓	57.1

- **多维度分析**：论文不仅比较了最终的答案准确率，还从检索效率、对不同视觉元素的处理能力以及推理效率（**表3**）等多个角度分析了模型的行为，为结论提供了更全面的支撑。

Table 3: Average Finish Rate (%) and Average Invalid Action Rate (%).

Method	Invalid Action Rate ↓	Finish Rate ↑
SFT	9.4	84.2
+ RL	**5.1**	**97.1**

作者对他们的结果非常有信心，他们主张其方法通过**将精细的视觉感知能力与与任务紧密对齐的强化学习奖励机制相结合**，为解决复杂的多模态RAG问题提供了一个强大而通用的解决方案。
