Report-桌面视觉智能体任务

> 本项目及代码已同步至 本人 GitHub 仓库

阶段 1

部署环境

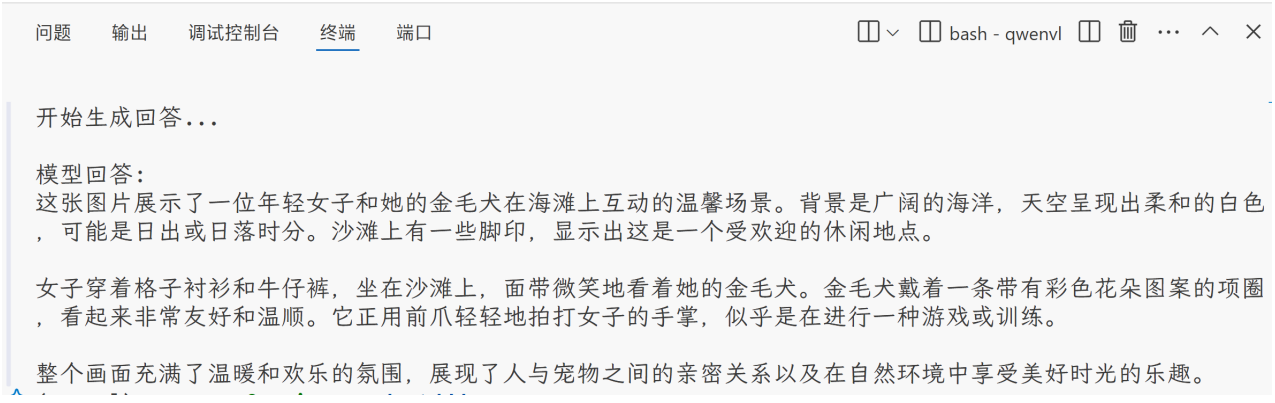
硬件配置：NVIDIA GeForce RTX 3060 Laptop (6GB VRAM)
OS：Ubuntu 20.04.6 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WSL2 x86_64)
软件包版本：见 Repo `requirements.txt`
选用模型：Qwen2.5-VL-3B-Instruct-unsloth-bnb-4bit

环境部署与模型加载

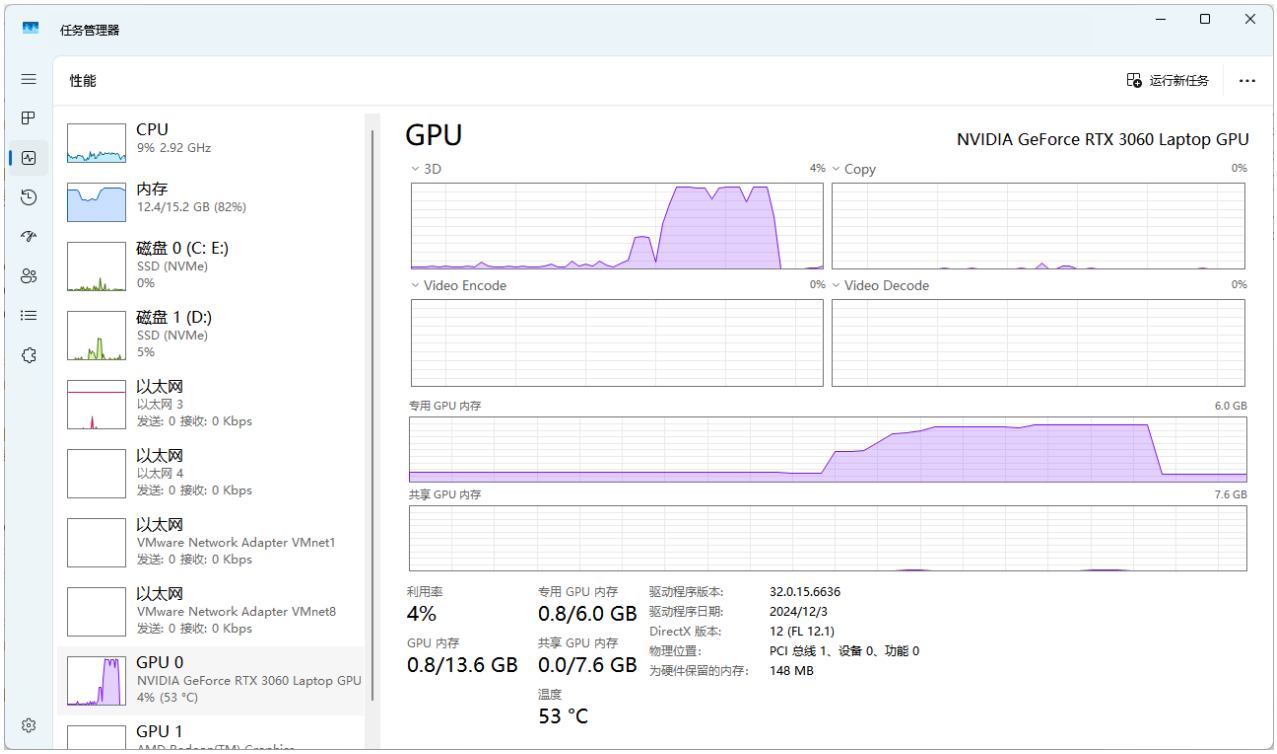
- 由于显存有限，参考官方文档后，决定使用 4bit BitsAndBytes 量化模型
  1. 创建、配置虚拟环境 conda
  2. 鉴于网络下载问题，使用 ModelScope SDK 下载模型，下载 flash_attn 预编译版本地安装（本地编译巨慢）
  3. 补充下载 qwen_vl_utils，transformers 等库

> *如想要启用 flash-attn-2 节约内存、加速计算？*
>
> 可以根据 cuda 版本选择合适的预编译 wheel 安装 ~~节约生命~~
>
> 但本人由于使用 Ubuntu 20.04，较新的预编译 flash-attn 使用先进的 GlibC_2.32（Ubuntu 22.04 支持），见 repo issue
> 因此不得不采取旧版 flash_attn（2.7.4.post1），这又导致必须使用旧版torch（2.6.0）
> 经典一个环境配一天

- 输出情况



- 占用情况



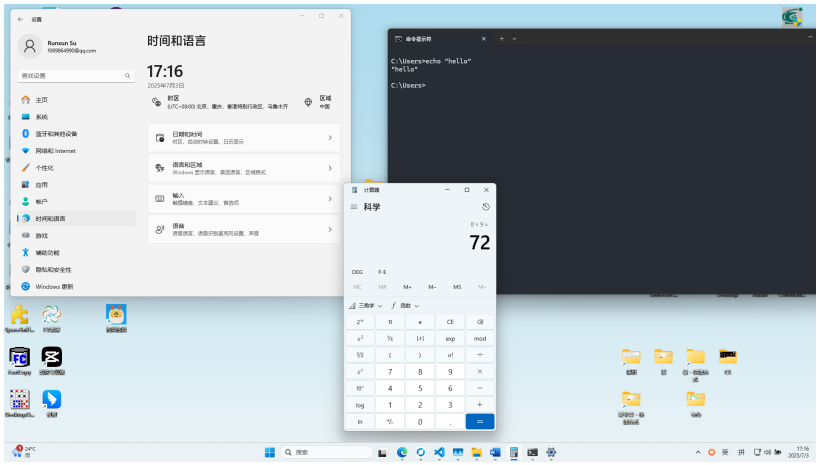
（模型加载、推理时的 GPU 内存占用；推理时恰好占用低于 6 GB）

> *为什么不使用 Qwen 官方提供的使用人数最多的 AWQ 量化模型？*
>
> 在测试部署环境时，出现了 `error: probability tensor contains either inf, nan or element < 0`
> 尝试后发现图片尺寸过大，需要 resize ，遂采用 BNB 量化模型
> （可能是显存不足，但在官方 repo 也存在待解决issue）

基础能力测试

1 测试素材准备

- 准备若干截图：桌面、文件管理器、网站登陆页面



（示例：桌面的截图，包含系统设置、计算器、命令行和桌面图标）

2 代码编写

重写示例文档，封装为调用函数 `get_vlm_response()` ，对不同任务测试

3 系统测试与分析

- 图像描述 Captioning
  所有的图片共用问题

```
*Prompt*

简单描述这张截图的内容。
```

- 视觉问答 VQA
  对于不同的截图，提出不同的问题（模拟实际工作流）

```
*Prompt*

desktop_clean.png: 这张截图中的桌面上有多少应用程序？它们分别是什么？

login_page.png: 这张截图中的用户可以通过什么方式登录？

file_explorer.png: 这张截图中的当前位于的路径是什么？
```

- 结果

```
==================================================
      开始执行Qwen-VL多模态任务测试
==================================================

--- 任务1: 图像描述 (Image Captioning) ---
The following generation flags are not valid and may be ignored: ['temperature']. Set `TRANSFORMERS_VERBOSITY=info` for mo
re details.
图片: data/desktop.png
问题: 简单描述这张截图的内容。
模型回答:
这张截图显示了一个Windows操作系统的界面，具体来说是一个设置窗口。左侧的菜单栏中包含了“主页”、“系统”、“蓝牙和其他设备”、“网
络和 Internet”、“个性化”、“应用”、“帐户”、“时间和语言”、“游戏”、“辅助功能”、“隐私和安全性”以及“Windows 更新”等选项。右侧是
当前时间（17:16）和日期（2025年7月3日），还有区域和语言设置、日期和时间设置、语言和区域设置等选项。在底部，有一个任务栏，
显示了多个应用程序图标，如浏览器、文件夹、记事本等。此外，还有一个命令提示符窗口，显示了一些文本输出。
```

```
--- 任务2: 视觉问答 (Visual Question Answering) ---
The following generation flags are not valid and may be ignored: ['temperature']. Set `TRANSFORMERS_VERBOSITY=info` for mor
e details.
图片: data/desktop.png
问题: 这张截图中有哪些应用程序窗口？
模型回答:
在这张截图中，有以下应用程序窗口：

1. Windows 设置窗口。
2. 命令提示符窗口。
3. 计算器窗口。
4. 文件资源管理器窗口（显示了多个文件夹和快捷方式）。

这些窗口分别位于不同的位置，展示了Windows操作系统的不同功能。

==================================================
      任务测试完成
--------------------------------------------------
```

（示例：桌面截图的对应任务输出，分别为 Captioning、VQA）

图片	任务类型	模型输出	评价	分析
仅含应用图标的桌面	Captioning	这是一张电脑桌面的截图，显示了一个湖边的风景图作为背景。 桌面上有三个图标：一个是 Microsoft Edge 浏览器的图标，另一个是 Microsoft Office 的图标，第三个是一个带有企鹅图案的应用程序图标。 屏幕右下角有一个任务栏，显示了时间、日期和一些系统图标。	成功	1 识别出桌面 2 描述了壁纸 3 识别出Icon图标
	VQA	这张截图中的桌面上有三个应用程序。它们分别是： 1. Excel 2. Microsoft Edge 3. 一个带有企鹅图案的图标，可能是某个特定的应用或文件夹 这些应用程序常用于办公、浏览网页和管理文件等。	部分成功	1. 未能识别出 QQ
含大量图标的文件管理器	Captioning	这是一张 Windows 操作系统的文件资源管理器截图，显示了 D 盘中的多个文件夹和应用程序。 文件夹包括 Adobe、American Truck Simulator、Blender 等，还有一些个人文件夹如 My eBooks 等。此外，还有虚拟机、Visual Studio 2022等开发相关文件夹。	成功	1 识别出多个区域的信息 2 识别出多个文件夹图标下方的名称
	VQA	当前所在的路径是：D:\文档	成功	1 识别出位于上方路径栏的文字
网站登陆页面	Captioning	这是一张登录界面的截图，用户需要通过手机号码和验证码进行登录。 页面顶部有硅基流动的标志和欢迎语。中间部分有一个输入框用于输入手机号码，并有一个获取验证码的按钮。下方还有一个邀请码输入框，但仅支持手机注册。底部有注册/登录按钮，以及一个勾选框表示 30 天内保持登录。 此外，还有其他登录方式的选项，如邮箱登录、微信登录等。最后，用户需要同意用户协议和隐私政策才能完成登录。	成功	1 识别出页面前端的垂直排列、表单元素 2 识别出可视度低的文字 3 识别出社交媒体 Icon 并能联系上下文
	VQA	这张截图中，用户可以通过手机号码和验证码、邀请码（仅支持手机注册）、邮箱登录以及通过微信、微博或Google等第三方平台进行登录。此外，用户还需要同意用户协议和隐私政策才能完成登录。	部分成功	1 识别出主要的登陆方式 2 错将 Github 识别为微博 3 指出细节：需要同意协议

4 分析与总结

1. OCR 能力：能够准确读出图片中的文字（按钮、窗口标题、文件名）
2. 空间关系：能够辨识“左上角”、“... 下方”这类概念
3. 幻觉现象：推测由于训练数据，会错误识别部分无文字的纯图像 Logo
4. 细粒度辨识：能分辨淡灰色字体、复选框
5. 通用知识：能辨识桌面、常见社交媒体 Icon

- 目前的能力测试仅体现模型定性的识别能力，接下来需要定量的 Grounding 能力分析，以更好地实现桌面智能

阶段 2

视觉定位 Grounding

> Qwen 官方 Repo 的 cookbook 中给出工具函数，直接改造使用

模块化

将之前的模型载入函数、新加入的 grounding 工具模块化

解析元素边界框

使用 one-shot 提示大模型遵循输出格式

```
*Prompt*


User instruction: "{instruction}"
Please provide a JSON list containing the bounding box for the requested element. The
format should be:
[
  {{"bbox_2d": [y1, x1, y2, x2], "label": "your_label"}}
]
```

```
--- 模型输入文本 ---
 <|im_start|>system
You are a helpful assistant that can accurately locate objects in an image based on user instructions and prov
de their coordinates in a JSON format.<|im_end|>
<|im_start|>user

User instruction: "定位登录按钮"
Please provide a JSON list containing the bounding box for the requested element. The format should be:
[
  {"bbox_2d": [y1, x1, y2, x2], "label": "your_label"}
]
The coordinates must be normalized between 0 and 1000.
<|vision_start|><|image_pad|><|vision_end|><|im_end|>
<|im_start|>assistant
```

(print 调试信息，可见特殊标识及拼接的输入提示词)

````
--- 模型原始输出 ---
 ```json
[
  {"bbox_2d": [293, 451, 637, 486], "label": "登录按钮"}
]
```
````

(大模型输出，按照 json 格式输出的 `bbox` 及 `label`)

可视化绘制

1. 提取 json 内容的函数 `parse_json_from_string()`

2. 绘制边界框及标签的函数 `plot_bounding_boxes()`

（待 Grounding 的原始图像输入）

（根据模型解析的边界框，调用绘制函数可视化；左为定位用户名输入框、右为定位登录按钮）

推理函数

接收输入、构建消息、应用模板、生成内容、调用解析函数、调用绘制函数

挑战性任务分析：能与不能

Grounding 共设计四项任务

1. 定位登录按钮（成功）
2. 定位用户名输入框（成功）
3. 定位关闭按钮（失败）
4. 定位多个元素（部分成功）

案例一：消歧定位——利用空间关系解决目标模糊性

现象：对于不同的 Prompt，`"定位关闭按钮"` 失败，但 `"定位右上角的关闭按钮"` 成功。

（左图错定位至注册/登录按钮；右图准确定位至右上角的 × 按钮）

分析：

1. 当单纯的物体描述不足以唯一确定一个目标时，**加入空间位置描述是极其有效的**（如“右上角”、“... 的左边”、“... 下面”）。
2. 这证明了模型具备一定的**空间理解**能力，并且能利用这种能力来解决视觉歧义。
3. 这是 VLM 作为桌面智能体的一个关键能力。

案例二：多目标定位——探索模型的指令复杂性处理上限

现象：定位两个目标时成功，当目标多于三个时失败

```
![[folders.png|500]] ![[folder_3.png|500]]
*（左图正确定位至 Linux、Program 文件夹；右图数量正确，但 Grounding 错误）*
```

分析：

1. 3B 模型的“认知负荷”或“复杂性上限”有限，当目标数量从2增加到3时，模型的性能下降。
2. 在设计任务流/Agent 时，应倾向于**简单、原子化的指令**。

总结

通过以上实验，我们发现 Qwen2.5-VL-3B 模型展现了**良好的基础定位精度**。
然而，其性能高度依赖于 Prompt 的质量。成功的关键在于**提供足够无歧义的信息**，例如利用空间关系词汇。
同时，我们也观察到模型在处理**复杂、多目标**指令时存在明显的**能力瓶颈**。
这些发现为下一阶段设计鲁棒的自动化流程提供了宝贵的经验：我们的智能体应采用**短小、精炼、单步**的指令与模型进行交互，而非试图让模型一次性理解和规划复杂的任务链。

阶段 3

任务设计

选择“使用 Windows 计算器计算 123 + 456”

(Workflow 执行完成后的截图示例)

Workflow

Observe → Think → Act

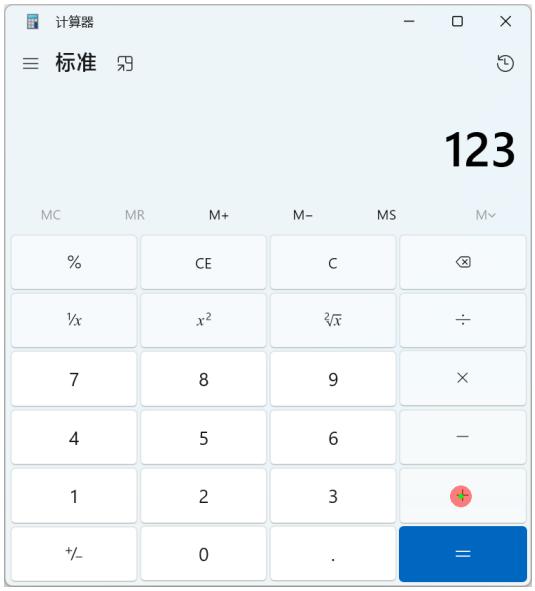
1. Observe: 当前图像中计算器 UI 呈现的算式
2. Think: 对比任务目标，确定下一步行动（人工划分任务，模拟 Agent 思考结果）
3. Act: 根据思考结果，调用 Grounding 确认坐标，输出 Click 指令，输出模拟可视化结果

```
--- 步骤 1/8 ---
👀 观察: data/calc_01_initial.png
🤔 思考: 我的下一步指令是 '定位按钮 '1''。正在定位...
--- 模型输入文本 ---
 <|im_start|>system
You are a helpful assistant. Locate the object in the image based on the instruction and provide its bounding box in JSON format.<|im_end|>
<|im_start|>user
Instruction: "定位按钮 '1'". Provide the JSON for the bounding box: [{"bbox_2d": [x1, y1, x2, y2], "label": "element"}]<|vision_start|><|image_p
ad|><|vision_end|><|im_end|>
<|im_start|>assistant

The following generation flags are not valid and may be ignored: ['temperature']. Set `TRANSFORMERS_VERBOSITY=info` for more details.

--- 模型原始输出 ---
 ```json
[
        {"bbox_2d": [84, 697, 173, 745], "label": "按钮 '1'"}
]
```
✅ 行动: 生成指令 CLICK(x=128, y=721)
🖼️ 可视化结果已保存至: output/calculator_task/step_01_action_on_calc_01_initial.png
```

(定义的 Workflow 执行时的输出)



（当前状态为 123 输入完毕，模型执行下一步：Click +，模拟可视化的结果；红圈为预计点击的位置）

思考

当前的 Workflow

1. 仅在相对稳定的环境下运行
   实际应用中，桌面状态动态变化剧烈
2. 任务序列由人工构造
   应该实现 Agent 自主规划“下一步该做什么”
3. 未与环境产生真实互动
   集成相关库，将 CLICK 等指令转换为真实操作

云侧大语言模型（参数较多，根据用户输入的目标，完成任务规划）
→ 端侧 Qwen-2.5-vl（参数较少，具有多模态功能，根据简单原子的任务，生成操作指令序列）
→ 云侧大语言模型（根据端侧更新的界面，下发新的任务）　（本地注意力机制？）
→ ...