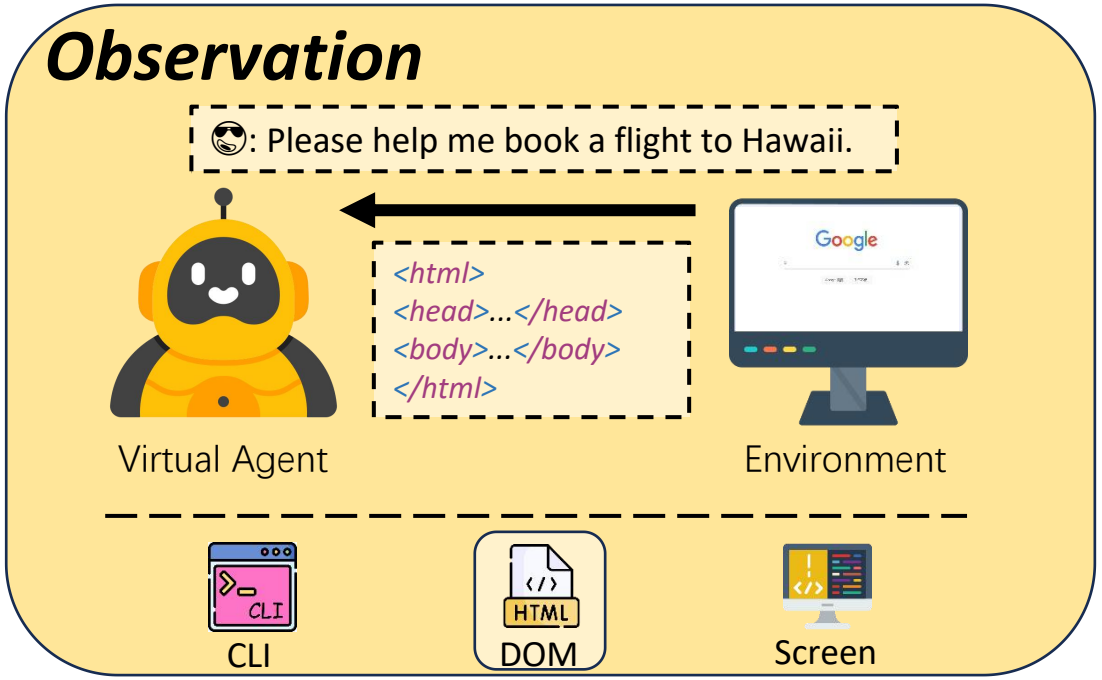
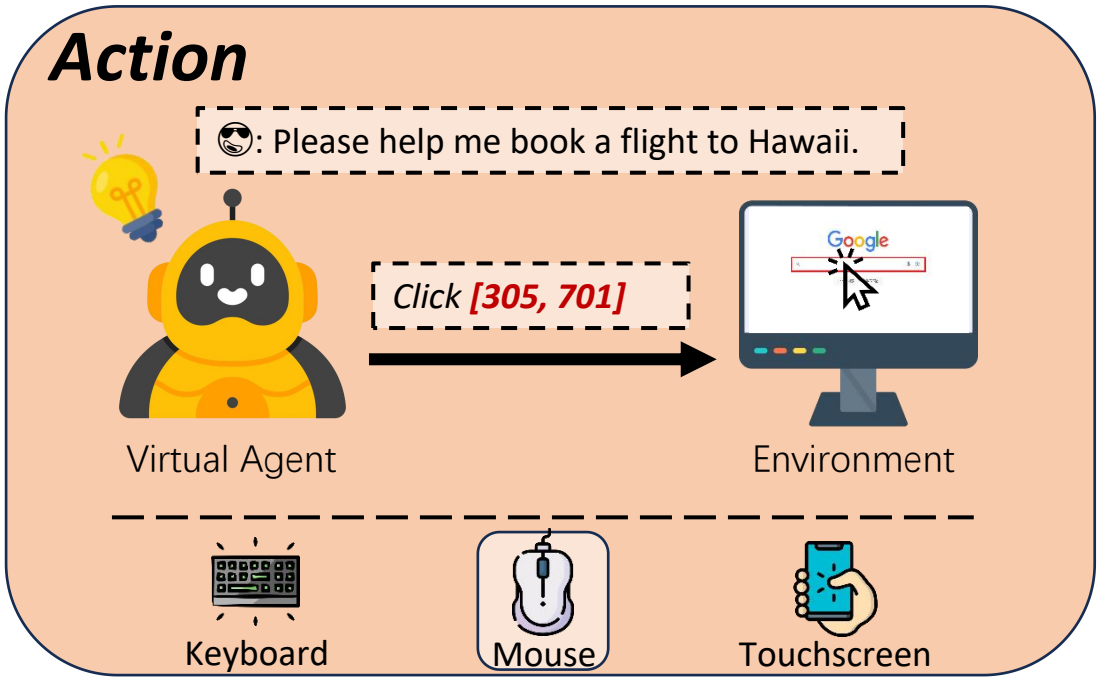
Compositional Arena

Overview

Virtual Agent：通过观察环境，决策下一步行动，迭代此过程直至完成任务的虚拟智能助手。

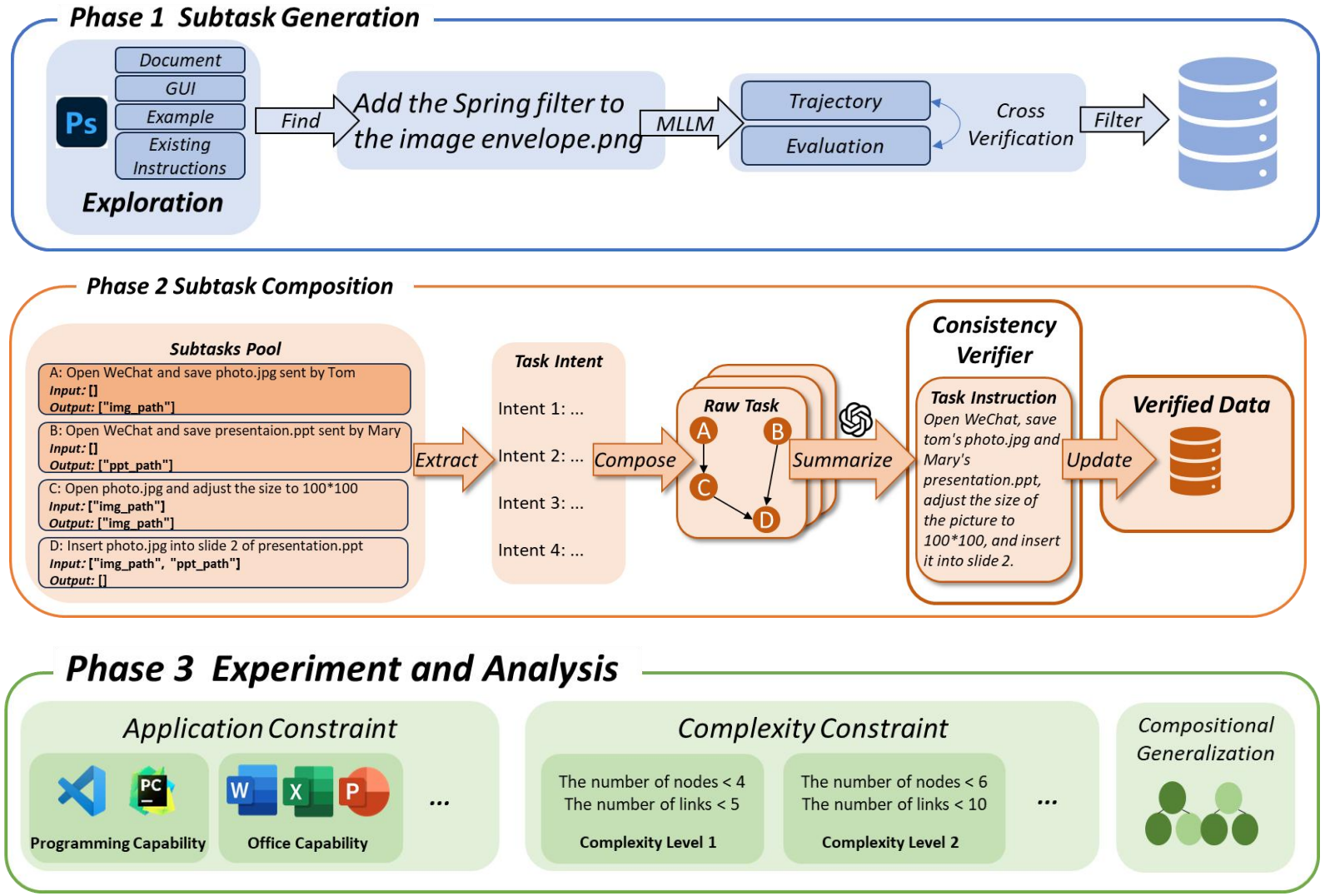
- 观察（Observation）：屏幕截图、结构化文本（HTML或XML文档）……
- 行动（Action）：鼠标点击、键盘输入……
- obs1 -> act1 -> obs2 -> act2 -> …





Compositional Arena
Overview

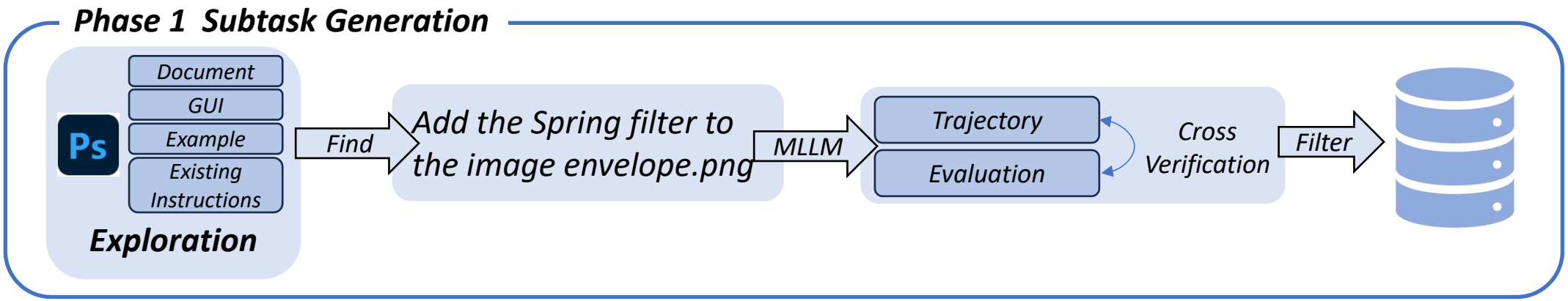
- **阶段1**：MLLM探索应用环境以生成子任务指令，根据该指令合成初步的执行轨迹和评估函数，之后通过交叉验证与人工筛选，过滤出准确的子任务执行轨迹和评估函数。
- **阶段2**：通过任务意图、一致性校验来保证合成任务的质量，将子任务组合为有向无环图形式的高质量数据。
- **阶段3**：从图的角度对任务进行复杂度和所需能力的划分，构建多维度测试集。对现有Agent开展实验与分析，测试它们在不同难度下的性能表现以及它们各项能力的强弱。



3. 研究方法

3.1 生成子任务

- MLLM在交互式环境（Windows 11）中不断探索，自行生成子任务的指令。
- 由目前SoTA的Agent生成子任务的执行轨迹。
- 由代码生成能力最强的MLLM生成子任务的评估函数。
- 最后通过交叉验证与人工清洗，过滤出准确的子任务数据 (**子任务指令、执行轨迹、评估函数**)。

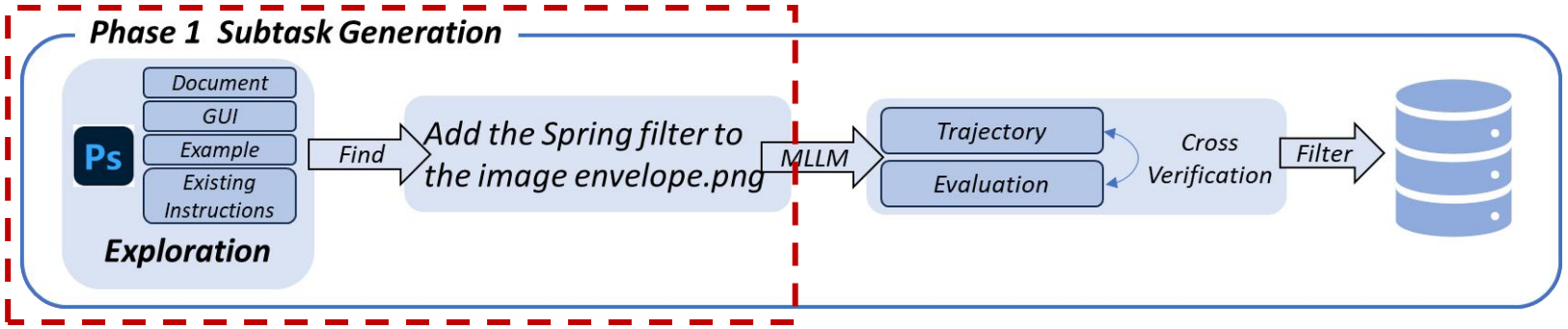


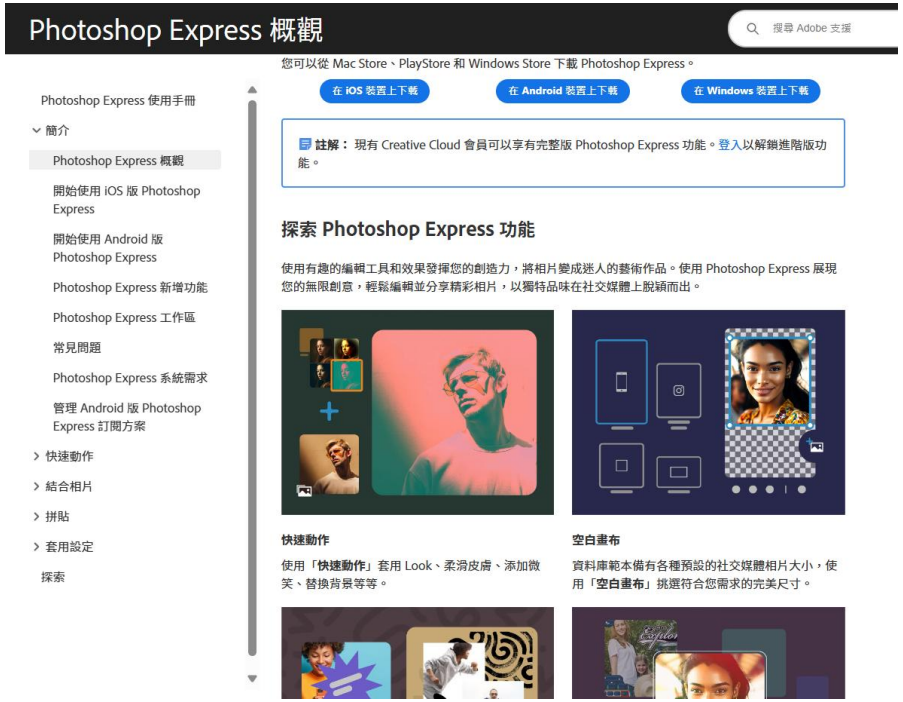
Compositional Arena

3. 研究方法

3.1 生成子任务

- 子任务指令生成





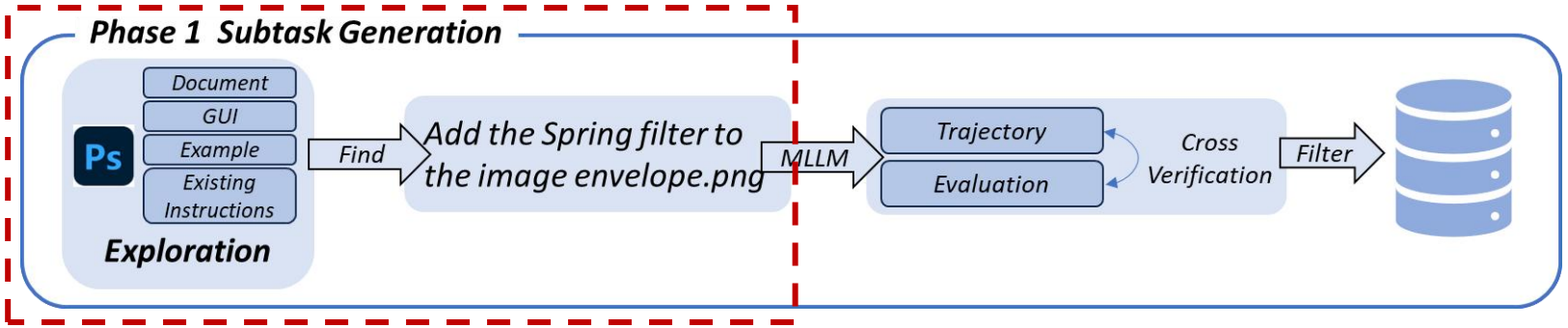
应用程序文档

应用程序GUI

3. 研究方法

3.1 生成子任务

- 子任务指令生成



生成的子任务共有8个字段

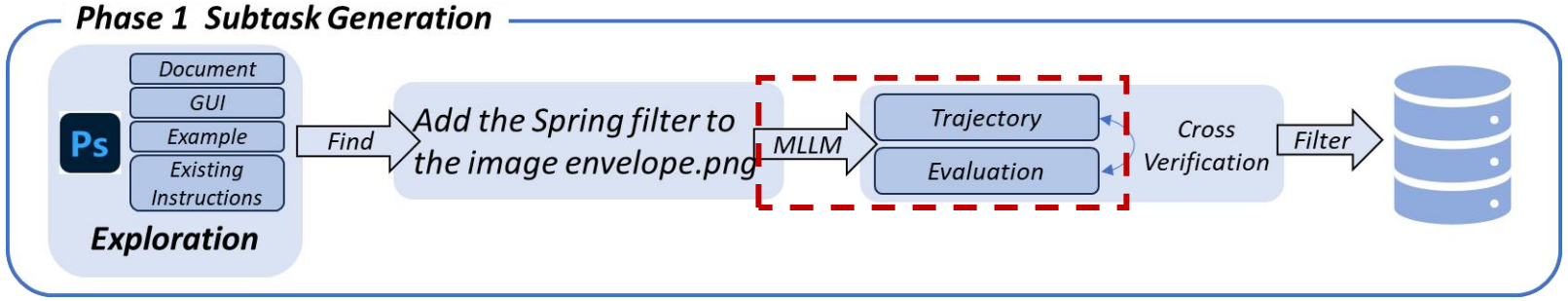
1. **id**: 子任务的uuid，用来唯一标识
2. **instruction**: 子任务的指令，其中包含参数的占位符
3. **related_app**: 子任务所属的app
4. **parameter**: 子任务的参数，为了与环境交互需要先实例化
5. **OS**: 子任务所处的操作系统
6. **input**: 用于组合子任务，表示该子任务执行前必须存在的资源
7. **output**: 用于组合子任务，表示该子任务执行后生成的资源
8. **candidate**: 子任务的候选参数

```
{
    "id": "4b8cf280-11db-ae8a-a196-45d56d931d52",
    "instruction": "Open the picture '{img_path}', add the '{filter_name}' filter from the 'Basic' series to the image.",
    "related_app": "Adobe Photoshop Express",
    "parameter": {
        "img_path": "C:\\Users\\user\\Desktop\\image\\envelope.png",
        "filter_name": "Spring"
    },
    "OS": "Windows",
    "input": [
        "img_path"
    ],
    "output": [
        "img_in_ps"
    ],
    "candidate": {
        "img_path": [
            "C:\\Users\\user\\Desktop\\image\\envelope.png",
            "C:\\Users\\user\\Desktop\\image\\beach.png",
            "C:\\Users\\user\\Desktop\\image\\cat.jpg",
            "C:\\Users\\user\\Desktop\\image\\dog.jpg",
            "C:\\Users\\user\\Desktop\\image\\city.jpg",
            "C:\\Users\\user\\Desktop\\image\\cliff.jpg"
        ],
        "filter_name": ["Spring","Normal","Vibrant","Autumn"]
    }
}
```

3. 研究方法

3.1 生成子任务

- 子任务执行轨迹生成
- 子任务评估函数生成



执行轨迹

Available APIs (手工设计):

```
def check_mouse_clicks(text: str) -> bool:
def check_keyboard_types(text: str) -> bool:
def check_file_exists(file_path: str) -> bool:
def check_text_exists_via_ocr(text: str) -> bool:
def check_text_exists_via_control(text: str) -> bool:
def check_text_exists(text: str) -> bool:
def check_text_exists_in_clipboard(text: str) -> bool:
```

Code LLM

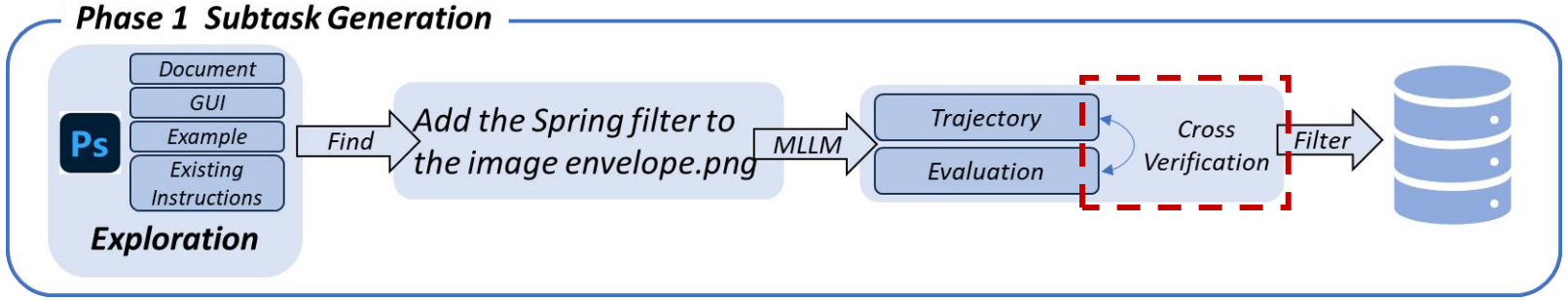
```
def evaluate_agent_task_completion():
    # Check if the file 'The Evolution of Urbanization Rate.xlsx' was opened
    if not check_file_exists(file_path='C:\\Users\\user\\Desktop\\office\\The
        return False

    # Check if the 'A' column was selected
    if not check_text_exists_via_control(text='A'):
        return False

    # Check if the content was centered
    if not check_text_exists_via_control(text='Center'):
        return False

    # All checks passed, task is considered complete
    return True
```

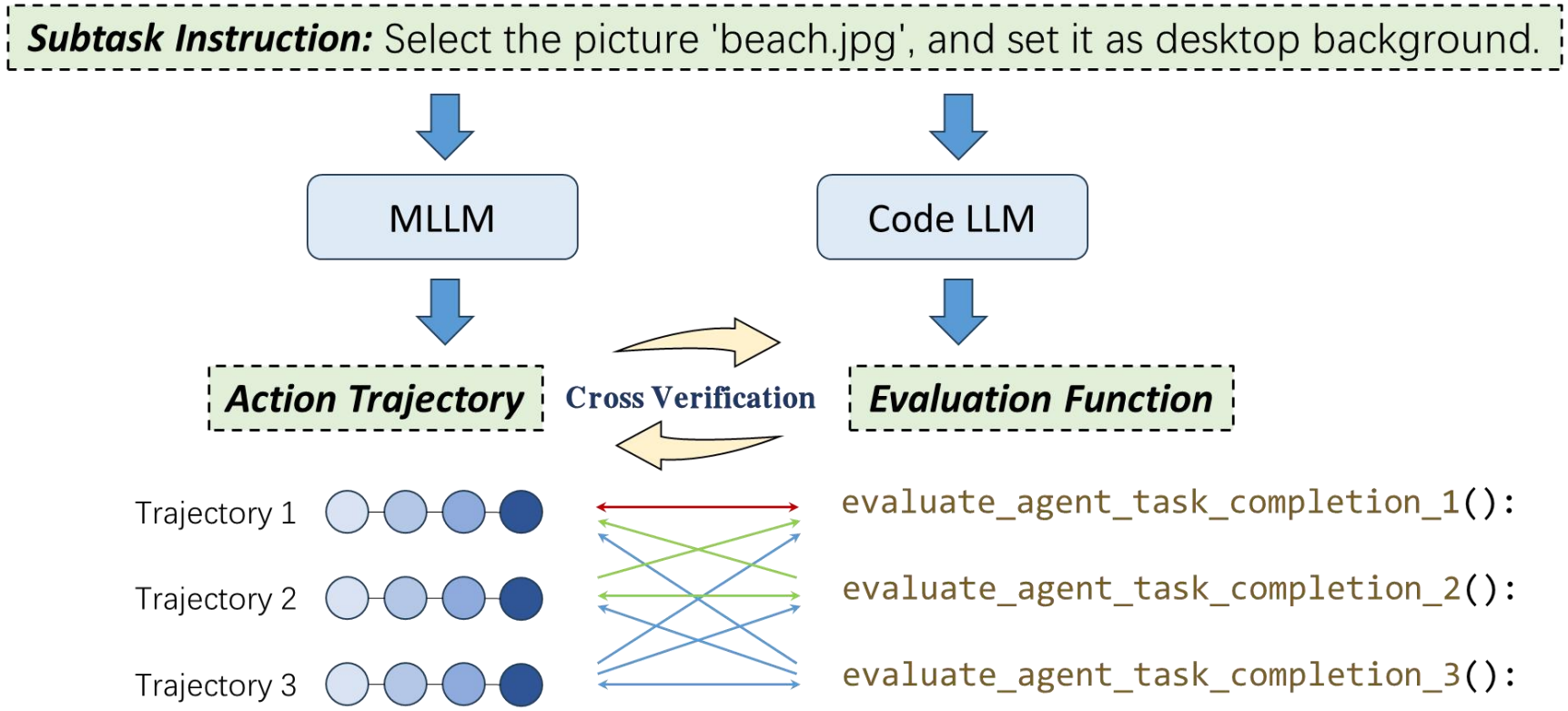
评估函数

3. 研究方法

3.1 生成子任务

- 交叉验证，迭代优化子任务执行轨迹和评估函数质量

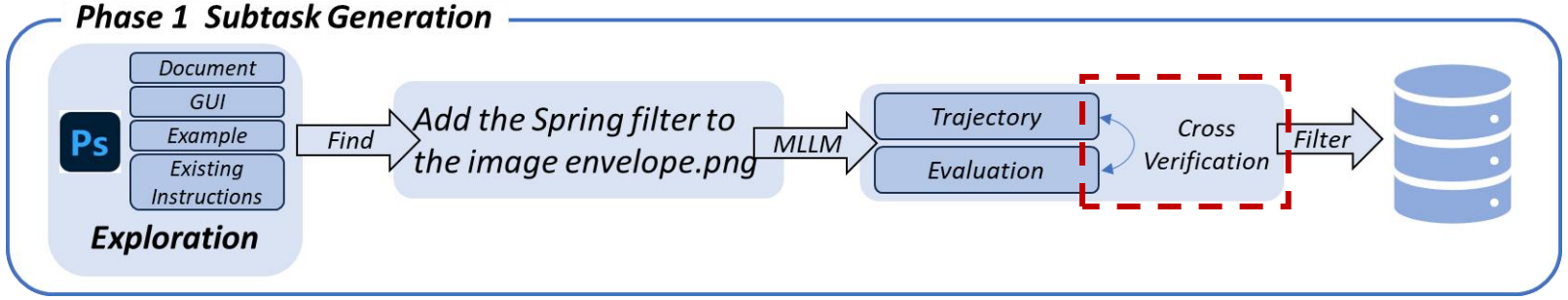


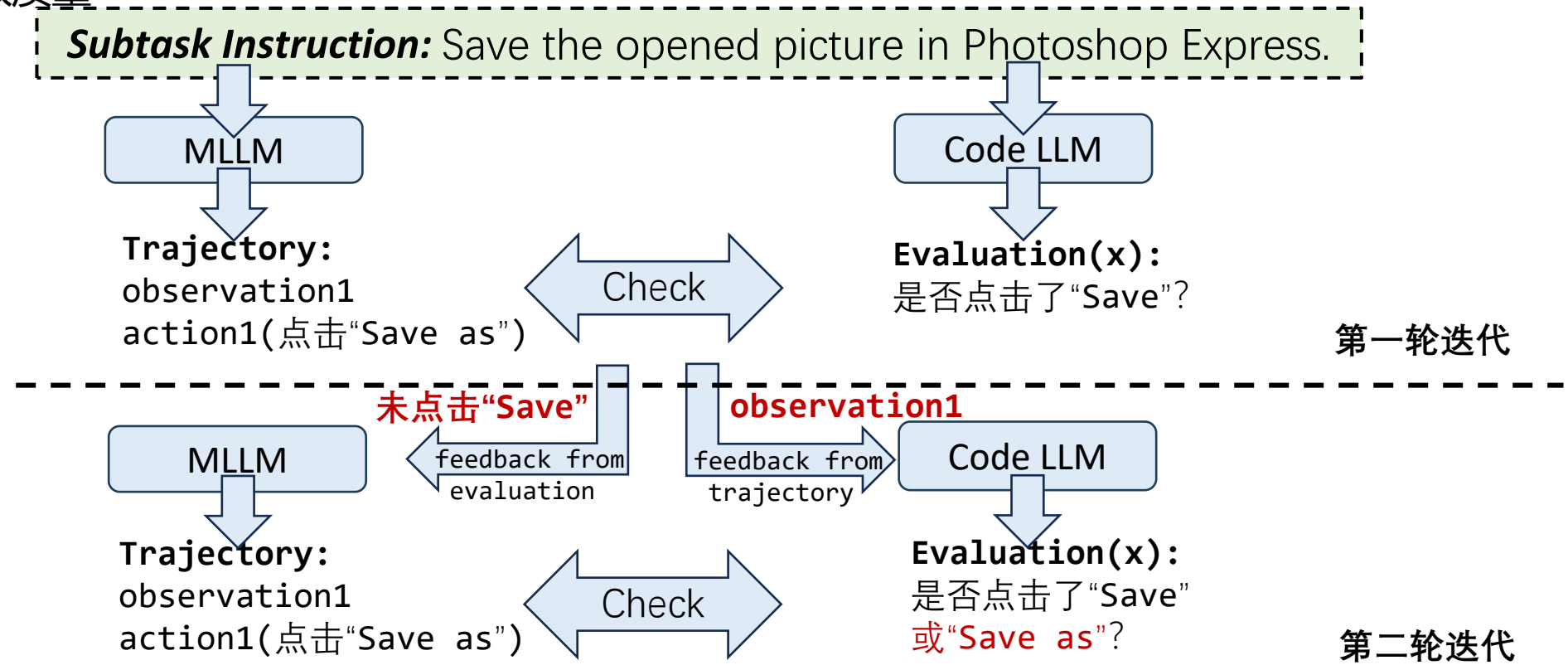


3. 研究方法

3.1 生成子任务

- 交叉验证，迭代优化子任务执行轨迹和评估函数质量

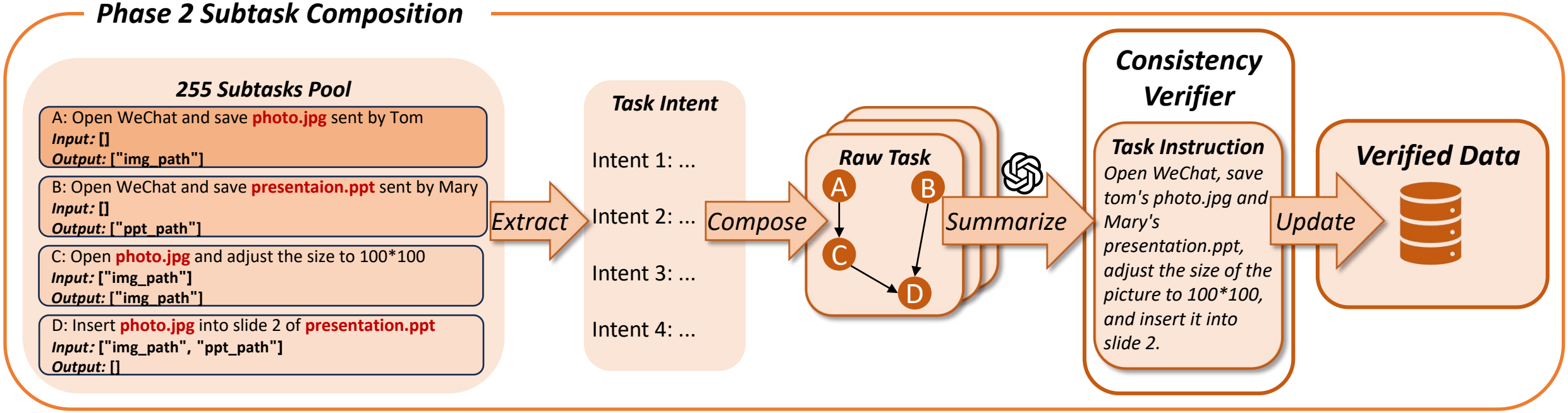




3. 研究方法

3.2 组合子任务

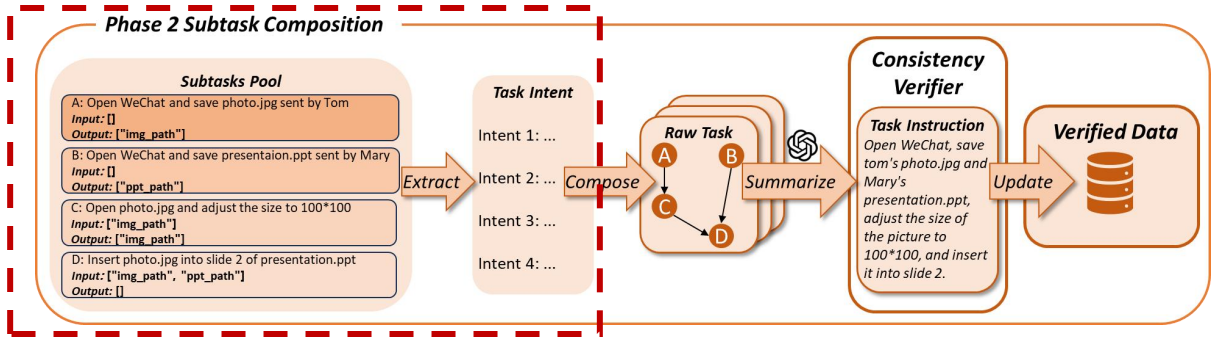
- 从子任务池中提取可能可以组合在一起的任务意图。
- 对于每一个任务意图，借助input和output枚举所有组合情况，每一个组合对应一张DAG。
- 将DAG中每个subtask的instruction总结为完整的task instruction。
- 对DAG和task instruction的语义一致性进行校验，再进行人工清洗，过滤出准确的任务数据。



3. 研究方法

3.2 组合子任务

- 从子任务池中提取任务意图




```
{
  "id": "ffbe926c-cac9-6acd-e4ec-bb9546621e72",
  "instruction": "Open '{xlsx_path}', click 'Page Layout', switch 'Margins' to 'Wide Margins'",
  "related_app": "Excel",
  "parameter": {
    "xlsx_path": "C:\\Users\\user\\Desktop\\office\\The Evolution of Urbanization Rate.xlsx"
  },
  "OS": "Windows",
  "input": ["xlsx_path"],
  "output": ["xlsx_in_processing"],
  "candidate": {
    "xlsx_path": "C:\\Users\\user\\Desktop\\office\\The Evolution of Urbanization Rate.xlsx"
  }
}
```

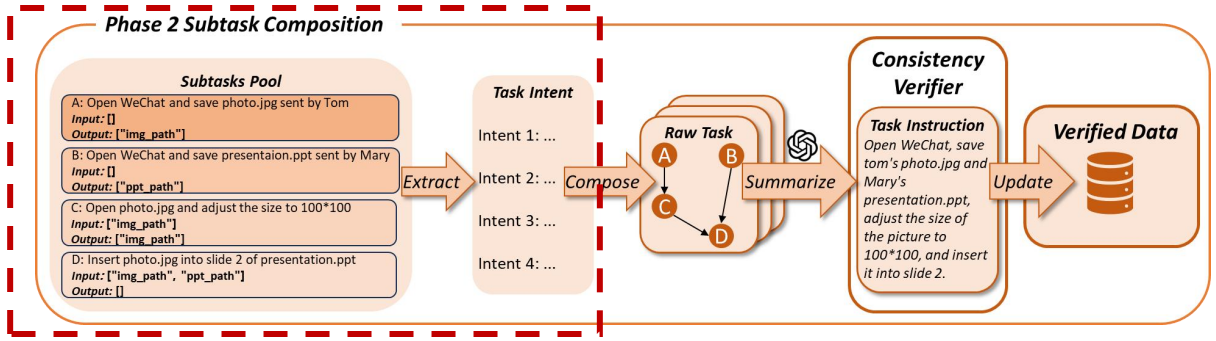

- 将子任务转换为Python函数形式
- Input转为函数的输入参数，Output转为函数返回值
- 便于后续LLM明确子任务间的关联关系

```
# Open '{xlsx_path}', click 'Page Layout', switch 'Margins' to 'Wide Margins'
def set_wide_margins_in_xlsx(xlsx_path: xlsx_path) -> xlsx_in_processing:
    pass
```

3. 研究方法

3.2 组合子任务

- 从子任务池中提取任务意图



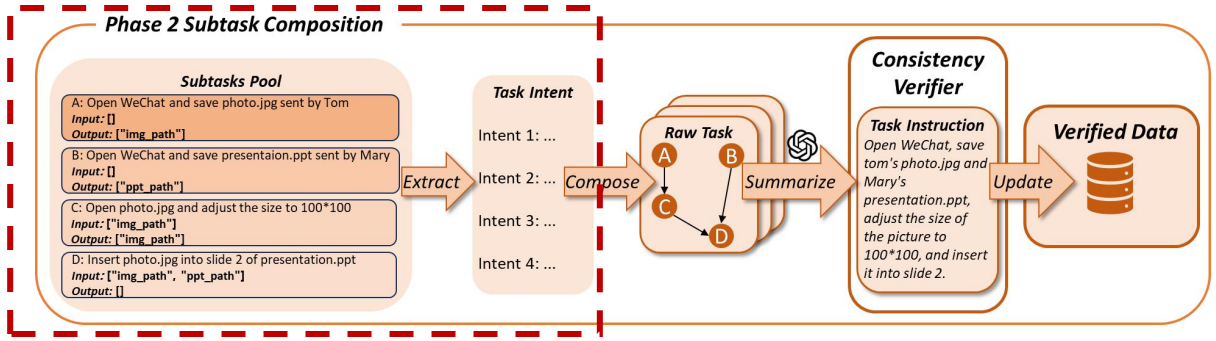

```python
# Open '{xlsx_path}', click 'Page Layout', switch 'Margins' to 'Wide Margins'
def set_wide_margins_in_xlsx(xlsx_path: xlsx_path) -> xlsx_in_processing:
    pass

# Using the file explorer, navigate to {dir_path} and new a Text Document named {file_name}
def create_text_document_in_dir(dir_path: dir_path, file_name: dir_path) -> file_path:
    pass

# Using the file explorer, navigate to {dir_path} and paste the file from the clipboard here
def paste_file_to_directory(dir_path: file_in_clipboard) -> file_path:
    pass

# Using the file explorer, navigate to {dir_path}, select the picture {img_name}, and rotate it left 90 degrees
def rotate_image_left_90_degrees(dir_path: img_path, img_name: img_path) -> img_path:
    pass
```

子任务池

3. 研究方法

3.2 组合子任务

- 从子任务池中提取任务意图



Task Intent

Translate Excel data to Chinese and update it in a Word document

Organize and compress project files into a zip folder

Enhance and save a photo with a special effect

Convert a PowerPoint presentation to PDF and extract specific pages

Rotate and add a photo to a PowerPoint slide

Update playlist cover and description in Spotify

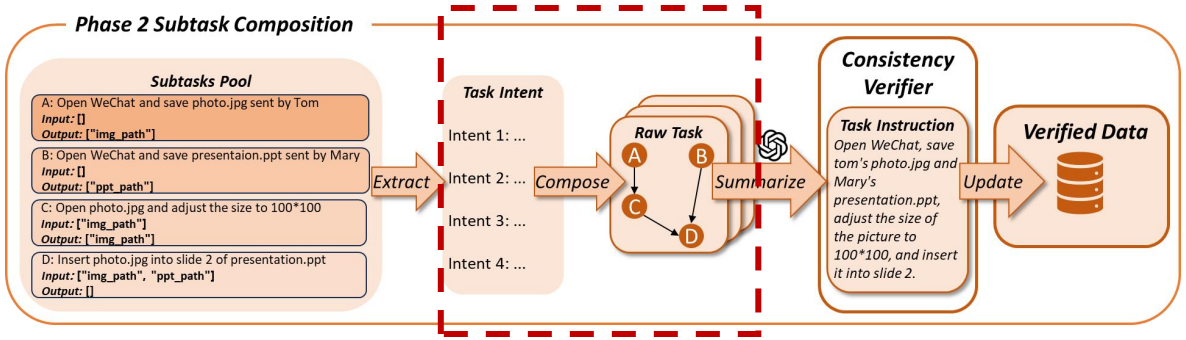
.....

任务意图

3. 研究方法

3.2 组合子任务

- 根据任务意图来组合DAG
- 这里会涉及subtask的实例化过程



intent 1： Enhance and save a photo with a special effect

Subtask 1: open ‘image.jpg’ in photoshop

Subtask 2: apply ‘spring’ filter to the current image.

Subtask 1: open ‘dog.jpg’ in photoshop

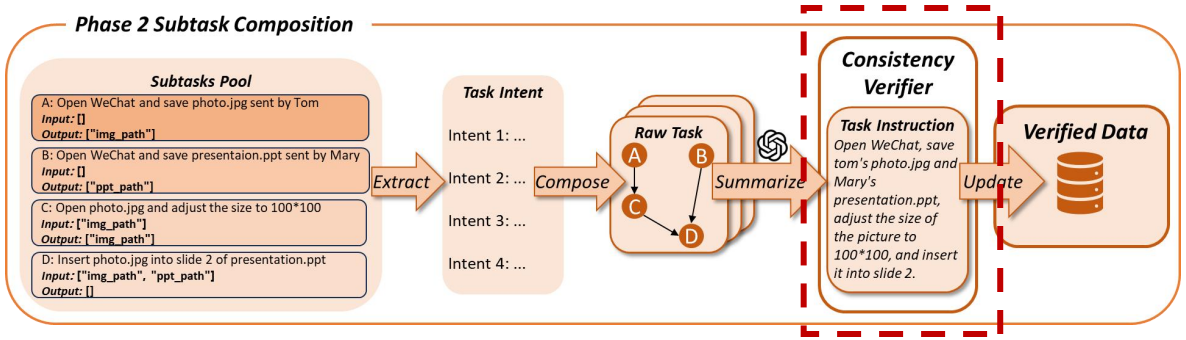
Subtask 2: apply ‘autumn’ filter to the current image.

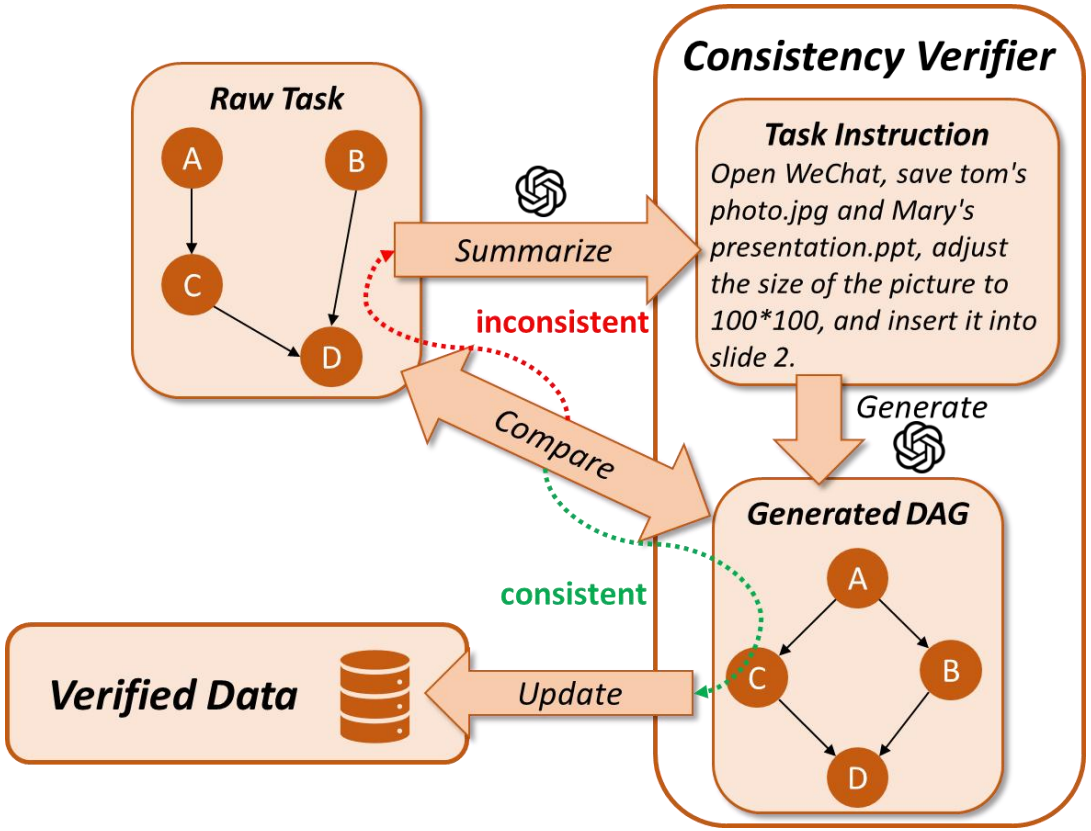
任务意图

Compositional Arena
3. 研究方法

3.2 组合子任务

- 对DAG和task instruction的语义一致性进行校验





3.3 环境

- Desktop：OSWorld（finished）
- Web：（WebArena / VisualWebArena）/ Mind2Web-Live
  - Paper: arxiv.org/pdf/2307.13854
  - Repo: web-arena-x/webarena: Code repo for "WebArena: A Realistic Web Environment for Building Autonomous Agents"
- Mobile：AndroidWorld
  - Paper: 2405.14573
  - Repo: google-research/android_world: AndroidWorld is an environment and benchmark for autonomous agents

3. 研究方法

3.4 方法总结

- Stage 1：子任务合成 (**70%**)

  1.1 先在本地电脑部署好交互式环境（GitHub - quick start）

  1.2 之后编写exploration代码，实现MLLM在环境内的自由游走

  1.3 优化prompt，使得MLLM在结束一轮exploration后总结出一条子任务的instruction （1 / 3）

  1.4 优化prompt，使MLLM能够根据instruction迭代地完成任务，并记录轨迹数据 （2 / 3）

  1.5 优化prompt，使code LLM能够根据instruction输出评估函数（3 / 3）

  1.6 实现交叉验证的代码，迭代地优化模型输出的评估函数和轨迹数据 (API)

  至此，子任务合成结束，已经可以拿到关键的三部分数据了（子任务指令、子任务执行轨迹、子任务评估函数）

- Stage 2：组合为任务 (30%)

  2.1 手工设计一批input / output 资源类型 (原则：确保连接后的相邻subtask可以无缝衔接！！！ text in clipboard)

  2.2 借助MLLM为每条子任务赋予input / output （finished）

  2.3 编写代码根据input & output将子任务组合为任务（finished）