

# WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models

**Authors: Hongliang He, Wenlin Yao, Kaixin Ma,  
Wenhao Yu, Yong Dai, Hongming Zhang,  
Zhenzhong Lan, Dong Yu**

**Zhejiang University, Tencent AI Lab, Westlake  
University**

汇报人: 袁沐寒  
2024年11月 22 日





1.研究背景

2.WebVoyager的系统架构

3.实验设计与评估

4.局限与未来发展

# 大型语言模型的发展

## GPT-3/GPT-4:

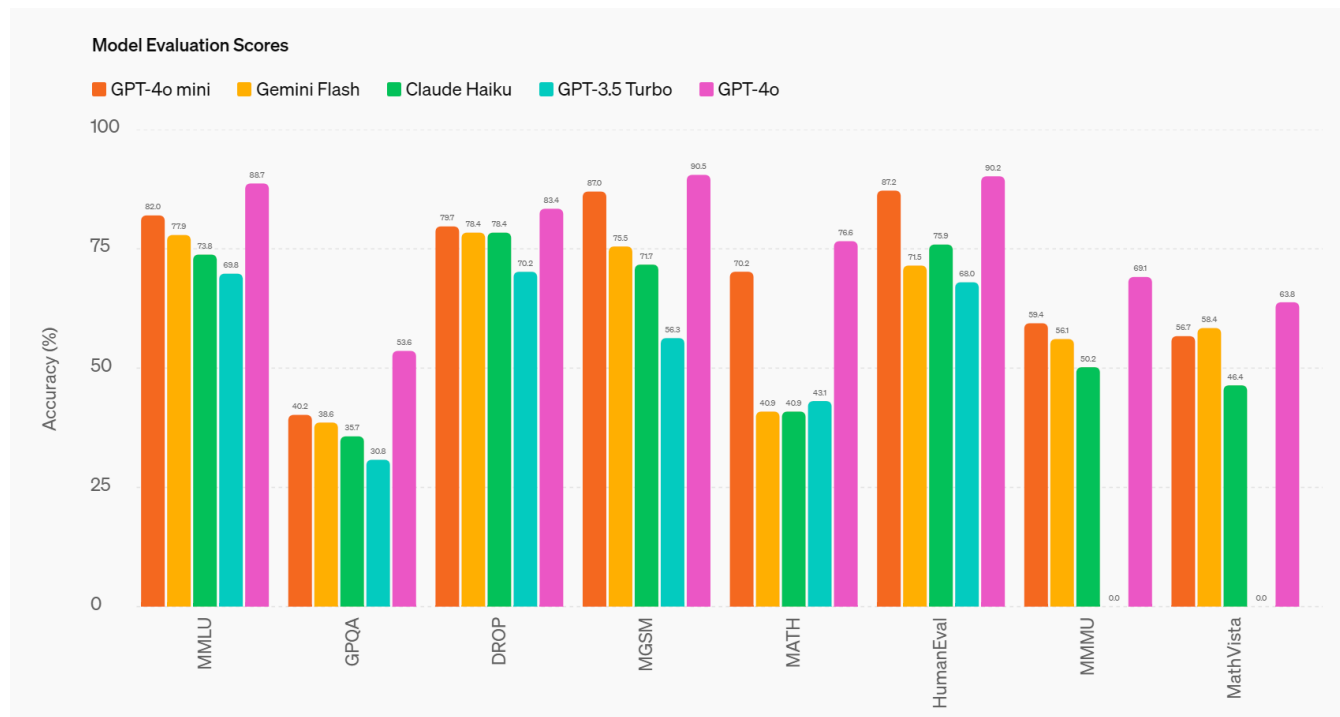
大规模语言模型，提升NLP性能

## 文本生成与理解:

出色的文本生成、问答能力

## 自动化对话:

推动ChatGPT、Bing Chat等应用



# 多模态模型引入

视觉与语言结合：  
GPT-4V等多模态模型

图像理解与文本推理：  
适应图像识别和文本处理

WebVoyager应用：  
结合截图与HTML元素执行网页任务

## (2) Prompt:

Describe the pointed region in the image.

Method	Validation set								Test set							
	in.		near.		out.		overall		in.		near.		out.		overall	
	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
OSCAR	85.4	11.9	84.0	11.7	80.3	10.0	83.4	11.4	84.8	12.1	82.1	11.5	73.8	9.7	80.9	11.3
Human	84.4	14.3	85.0	14.3	95.7	14.0	87.1	14.2	80.6	15.0	84.6	14.7	91.6	14.2	85.3	14.6
VIVO	92.2	12.9	87.8	12.6	87.5	11.5	88.3	12.4	89.0	12.9	87.8	12.6	80.1	11.1	86.6	12.4
VisuL	103.7	12.7	95.6	13.4	83.8	11.9	94.3	13.1	98.0	13.6	95.2	13.4	78.0	11.5	92.5	13.1
UFO	103.9	14.5	95.5	13.8	83.5	12.3	94.3	13.6	98.9	14.3	94.7	13.9	77.9	12.1	92.3	13.6
mPLUG	-	-	-	-	-	-	114.8	14.8	-	-	-	-	-	-	-	-
SivaVLM	113.7	-	110.9	-	115.2	-	115.2	-	113.7	-	110.9	-	115.2	-	115.2	-
LEMOM	118.0	15.4	116.3	15.1	120.2	14.5	117.3	15.0	112.8	15.2	115.5	15.1	110.1	13.7	114.3	14.9
UniversalCap	123.2	15.0	121.5	15.3	125.4	14.4	122.1	15.0	118.9	15.4	120.6	15.2	114.3	14.1	119.3	15.1
CoCa	-	-	-	-	-	-	122.4	15.5	-	-	-	-	-	-	120.6	15.5
GIT <sub>0</sub>	100.7	13.8	97.7	13.5	89.6	12.5	96.6	13.4	-	-	-	-	-	-	-	-
GIT <sub>L</sub>	107.7	14.9	107.8	14.5	102.5	13.7	106.9	14.4	-	-	-	-	-	-	-	-
GIT	129.8	16.3	124.1	16.0	127.1	15.7	125.5	16.0	122.4	16.2	123.9	16.0	122.0	15.7	123.4	15.9
GIT2	126.9	16.1	125.8	16.2	130.6	15.8	126.9	16.1	124.2	16.4	125.5	16.1	122.3	15.6	124.8	16.1

## GPT-4V:

The highlighted red region in the image is a table cell that contains a numerical value of 122.3. It is located in the "Test set" section of the table, under the "out" column and the "GIT2" row. This value represents the overall performance metric for the GIT2 method on the test set, in the "out" category.

## Prompt:

Which wisdom tooth is hardest to remove?



Note: Same query image as shown above, resized to save space

## GPT-4V:

The lower wisdom teeth, especially those that are impacted, are typically considered the hardest to remove. This is because the lower jawbone is denser than the upper jawbone, and the impacted wisdom teeth may be more difficult to access. Additionally, the roots of the lower wisdom teeth are often longer and more curved than those of the upper wisdom teeth, making them more challenging to remove.

## Prompt:

Localize each person in the image using bounding box. What is the image size of the input image?



## GPT-4V:

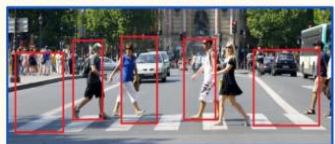
The image size of the input image is 1000 x 333 pixels.

Bounding boxes for each person:

- Person 1 (leftmost): (x1, y1) = (22, 108), (x2, y2) = (170, 300)
- Person 2: (x1, y1) = (200, 80), (x2, y2) = (290, 270)
- Person 3: (x1, y1) = (350, 70), (x2, y2) = (460, 280)
- Person 4: (x1, y1) = (540, 80), (x2, y2) = (640, 270)
- Person 5 (rightmost): (x1, y1) = (760, 100), (x2, y2) = (920, 290)

Note: These bounding box coordinates are approximate and may not be exact.

## Visualization of Detected Bbox:



## Prompt:

Count the number of apples in the image.



## GPT-4V:

There are 7 apples in the image.

# Web Agent的需求与局限

## 单一模态输入:

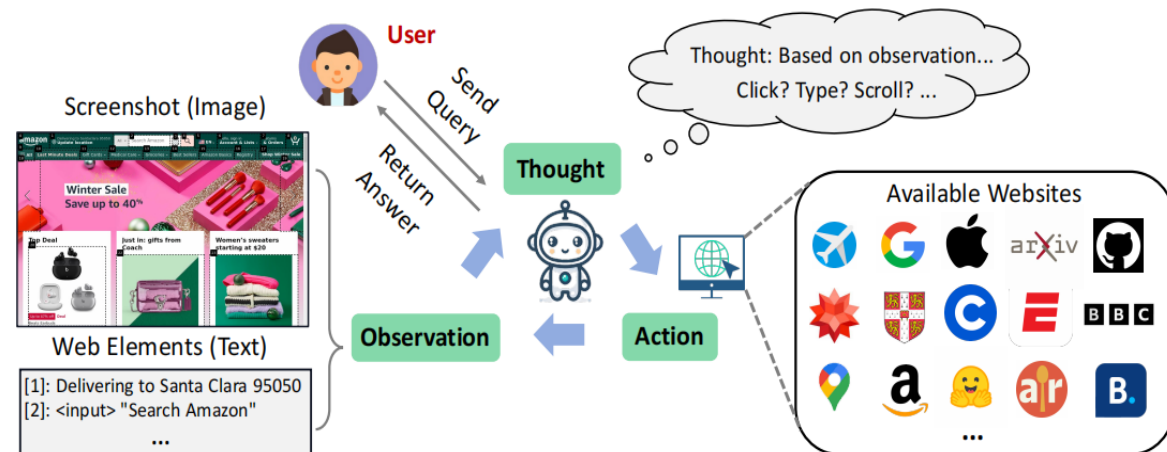
仅使用文本, 无法处理视觉信息 (如按钮、图像)

## 简化评估环境:

多在静态网页或模拟器中测试, 缺乏真实场景适应性

## 任务复杂性不足:

仅执行单步任务, 无法应对多步、动态网页任务



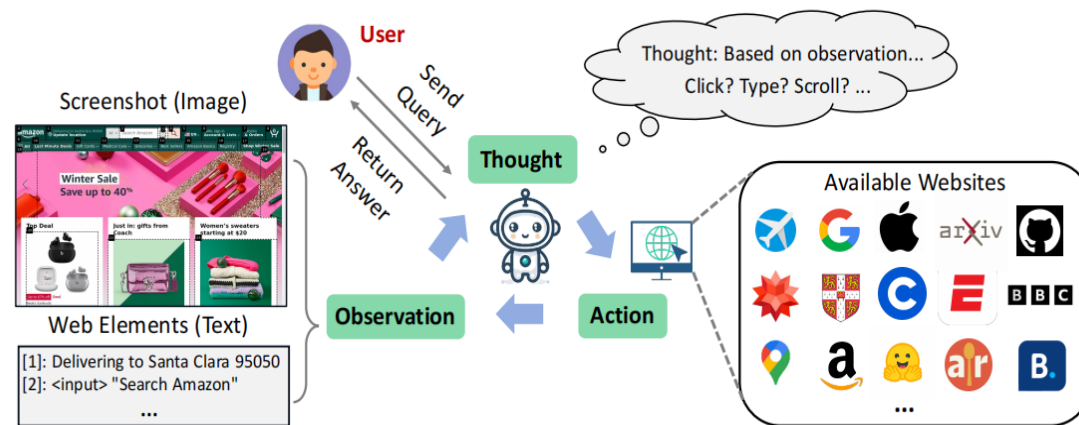
# 研究目标与主要贡献

## 研究目标：

构建WebVoyager，使其成为一个完全自动化的多模态Web Agent，能够在动态网页环境中完成从查询到决策的全流程任务。

## 主要贡献：

- 提出首个基于视觉和文本输入的Web Agent，能够执行端到端的网页任务。
- 建立新的评估基准，从15个热门网站中收集真实任务，以验证WebVoyager的能力。
- 开发自动化评估协议，利用GPT-4V作为评估工具，提高评估的规模化和一致性。





1. 研究背景
2. WebVoyager的系统架构
3. 实验设计与评估
4. 局限与未来发展

# WebVoyager系统概述

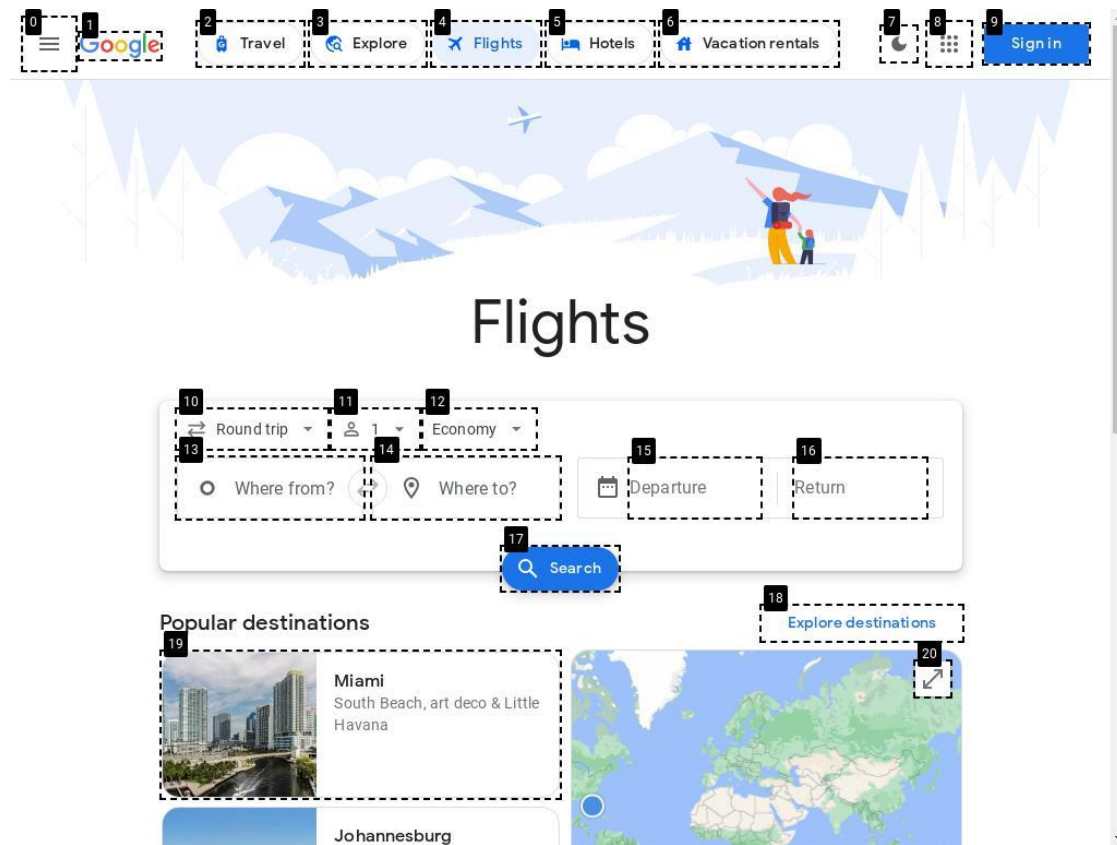
- **输入：**结合网页截图（图像）和HTML元素（文本）。
- **任务类型：**能够执行查询、信息检索、在线交互等任务。
- **输出：**WebVoyager自动完成任务并返回答案，如信息查找、商品价格查询等。
- **核心目标：**实现完全自动化的网页任务执行，无需人工干预。





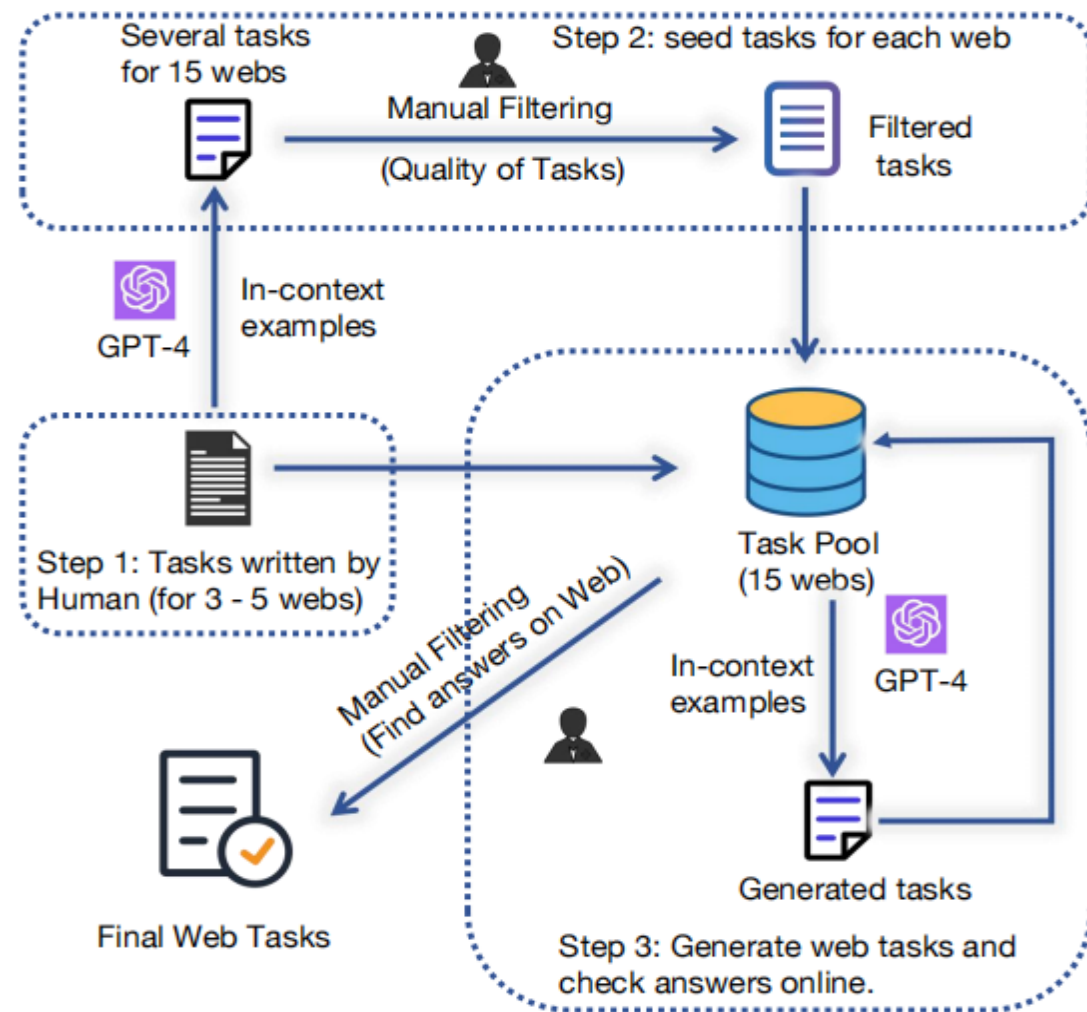
# 视觉与文本的输入处理

- **视觉输入：** WebVoyager通过网页截图捕捉页面内容，**标注每个交互元素**（如按钮、输入框）并为其分配标签，帮助模型做出决策。
- **文本输入：** 提取HTML中的**交互元素信息**（如按钮文字、输入框提示），结合**页面结构信息**（如ARIA标签）。
- **处理方式：** 结合**视觉元素**（如页面布局、按钮位置）和**文本信息**（如交互提示、元素类型）共同决定下一步操作。



# 动作决策与执行

- **观察：** 在每一步， WebVoyager从网页截图和HTML元素中提取信息，获取当前页面的状态。
- **思考：** 模型根据当前观察生成推理，选择适当的操作（如点击、输入、滚动等）。
- **执行：** 通过Selenium等自动化工具，模型执行选择的操作，在页面中完成交互动作。
- **操作示例：** 例如， 点击搜索框并输入查询内容， 或滚动页面以查找相关信息。



# 多步交互与上下文更新

- **多步交互**: WebVoyager能够处理需要多个步骤的复杂任务, 如从网页中查找并购买商品, 任务执行依赖于每一步的操作反馈。
- **上下文更新**: 每次操作后, WebVoyager会根据反馈更新上下文, 包括**历史操作**和**新的网页观察**。这个更新过程帮助模型做出下一步决策。
- **操作反馈**: 如果某一步操作失败 (如按钮点击无效), WebVoyager会根据新的页面**反馈调整**下一步操作, 避免错误重复。

Formally, we denote the Environment as  $\mathcal{E}$ , the large Multimodal Model as  $\mathcal{M}$ , the Observation Space as  $\mathcal{O}$ , and the Action Space as  $\mathcal{A}$ . At time step  $t$ , the model receives the context  $c_t$  as inputs, which consist of historical actions  $a_i$  and observations  $o_i$ , defined as:  $c_t = (o_1, a_1, \dots, o_{t-1}, a_{t-1}, o_t, I)$ . The model produces the action  $a_t$  at time  $t$ ,  $a_t = \mathcal{M}(c_t)$ , which is then executed in the environment. After execution, the environment sends back the observation at time  $t + 1$ ,  $o_{t+1} = \mathcal{E}(o_t, a_t)$ . Then the context will be updated and this interaction process continues until the model generates a terminating action or the maximum step is reached.

Inspired by the paradigm of ReAct Prompting (Yao et al., 2022b), we also prompt our agent to generate a thought process first before generating the action code. Hence  $a_t$  can be further composed into  $(s_t, \hat{a}_t)$  where  $s_t$  and  $\hat{a}_t$  represent the natural language thought and action code respectively. Figure 7 in Appendix A presents the System Prompt we designed for the action prediction step. Also, it's worth noting that excessive observations of web pages from longer episodes may confuse the agent. Therefore, we perform context clipping to remove outdated web page information and only keep the three most recent observations in the inputs, and we keep the entire history of thoughts and actions to better guide the agent.

Imagine you are a robot browsing the web, just like humans. Now you need to complete a task. In each iteration, you will receive an Observation that includes a screenshot of a webpage and some texts. This screenshot will feature Numerical Labels placed in the TOP LEFT corner of each Web Element. Carefully analyze the visual information to identify the Numerical Label corresponding to the Web Element that requires interaction, then follow the guidelines and choose one of the following actions:

1. Click a Web Element.
2. Delete existing content in a textbox and then type content.
3. Scroll up or down.

...

Correspondingly, Action should STRICTLY follow the format:

- Click [Numerical\_Label]
- Type [Numerical\_Label]; [Content]
- Scroll [Numerical\_Label or WINDOW]; [up or down]
- Wait
- GoBack
- Google
- ANSWER; [content]

Key Guidelines You MUST follow:

\* Action guidelines \*

- 1) Execute only one action per iteration.

...

\* Web Browsing Guidelines \*

- 1) Don't interact with useless web elements like Login, Sign-in, donation that appear in Webpages.

...

Your reply should strictly follow the format:

Thought: {Your brief thoughts (briefly summarize the info that will help ANSWER)}

Action: {One Action format you choose}

Then the User will provide:

Observation: {A labeled screenshot Given by User}

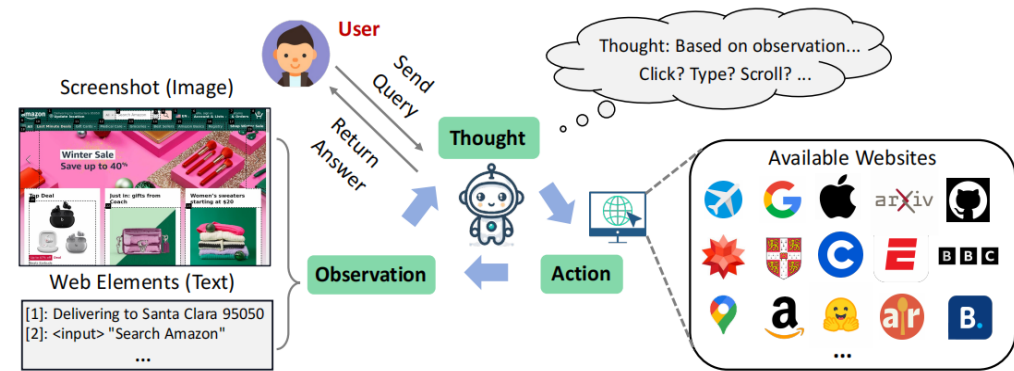


1. 研究背景
2. WebVoyager的系统架构
3. 实验设计与评估
4. 局限与未来发展



# 数据集概述与任务设计

- 数据集构建：包含15个热门网站，涵盖电商、地图查询、词典等场景。
- 任务类型：包括查询、商品对比、实时数据获取等，任务设计为多步骤，具有挑战性。
- 目标：确保任务的多样性与复杂性，覆盖广泛应用场景。



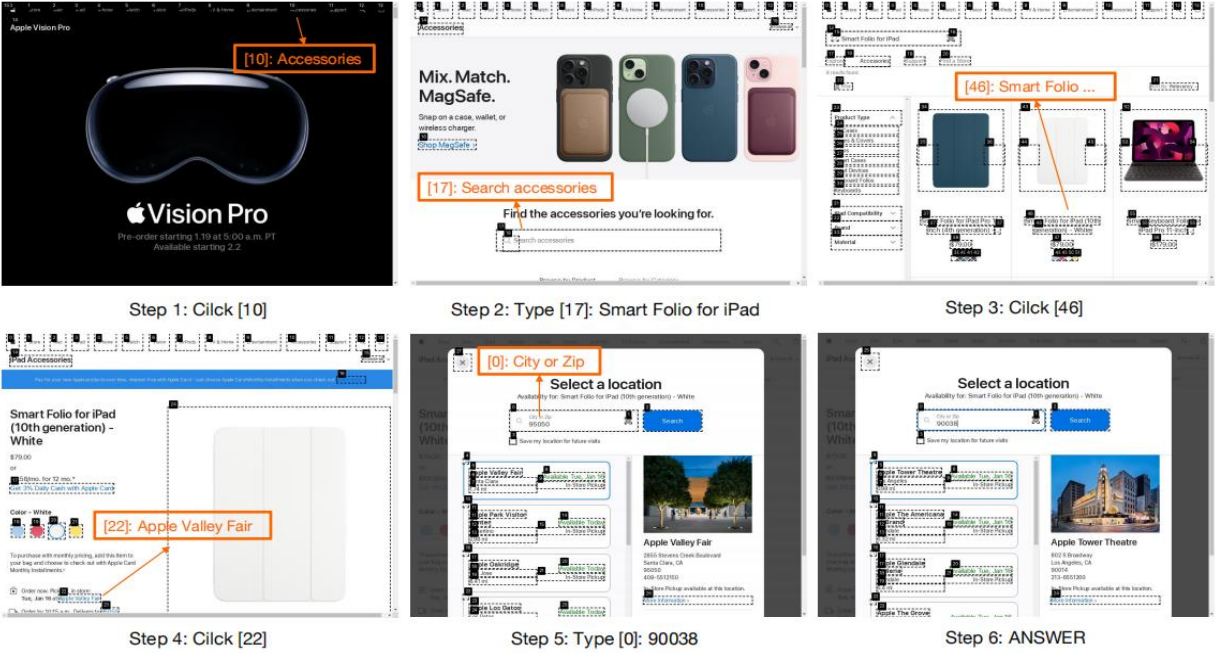
	Allrecipes	Amazon	Apple	ArXiv	GitHub	Booking	ESPN	Coursera
GPT-4 (All Tools)	11.1%	17.1%	44.2%	14.0%	48.8%	22.7%	31.8%	31.0%
WebVoyager <sub>Text-only</sub>	<b>55.6%</b>	31.7%	34.9%	32.6%	61.0%	2.3%	36.4%	23.8%
WebVoyager	53.3%	<b>58.5%</b>	<b>65.1%</b>	<b>51.2%</b>	<b>63.4%</b>	<b>43.2%</b>	<b>38.6%</b>	<b>73.8%</b>
WebVoyager <sub>Text-only</sub> *	57.8%±0.0%	43.1%±1.4%	36.4%±3.5%	50.4%±1.4%	63.4%±2.5%	2.3%±0.0%	38.6%±2.3%	24.6%±1.4%
WebVoyager*	51.1%±2.2%	52.9%±1.4%	62.8%±2.3%	52.0%±1.3%	59.3%±3.7%	32.6%±2.7%	47.0%±1.3%	57.9%±2.7%
WebVoyager <sub>Claude</sub> *	45.9%±3.4%	58.6%±4.2%	58.1%±4.0%	55.0%±7.0%	56.9%±1.4%	19.0%±1.3%	46.2%±1.3%	68.2%±1.3%
WebVoyager <sub>GPT-4o</sub> *	56.3%±1.3%	53.7%±2.5%	56.6%±1.3%	60.5%±0.0%	57.7%±3.7%	43.9%±3.5%	44.0%±2.7%	65.1%±2.8%

	Cambridge Dictionary	BBC News	Google Flights	Google Map	Google Search	Huggingface	Wolfram Alpha	Overall
GPT-4 (All Tools)	25.6%	9.5%	2.4%	53.7%	60.5%	37.2%	52.2%	30.8%
WebVoyager <sub>Text-only</sub>	62.8%	45.2%	7.1%	61.0%	67.4%	20.9%	58.7%	40.1%
WebVoyager	<b>65.1%</b>	<b>61.9%</b>	<b>59.5%</b>	<b>70.7%</b>	<b>76.7%</b>	<b>44.2%</b>	<b>63.0%</b>	<b>59.1%</b>
WebVoyager <sub>Text-only</sub> *	66.7%±3.6%	45.2%±2.4%	7.1%±0.0%	62.6%±2.8%	75.2%±1.3%	31.0%±1.4%	60.2%±1.3%	44.3%±0.6%
WebVoyager*	71.3%±1.3%	60.3%±2.8%	51.6%±1.4%	64.3%±2.8%	77.5%±2.7%	55.8%±2.3%	60.9%±2.2%	57.1%±0.2%
WebVoyager <sub>Claude</sub> *	71.3%±3.6%	66.7%±4.8%	15.1%±5.5%	55.3%±1.4%	72.9%±1.3%	53.5%±4.7%	51.5%±5.4%	52.8%±1.4%
WebVoyager <sub>GPT-4o</sub> *	82.2%±1.3%	54.8%±2.4%	28.6%±0.0%	56.9%±2.8%	63.6%±1.3%	42.6%±3.6%	65.2%±2.2%	55.5%±0.8%

# 自动评估方案

- 评估工具：使用GPT-4V结合视觉与文本信息，自动评估任务成功率和操作精度。
- 一致性：与人类标注一致性高达85.3%，Kappa值0.7，确保评估可靠性。
- 高效性：自动化评估提高效率，减少人工标注工作量。

	Success Rate	Consistency	
		Agreement	$\kappa$
k=1	47.7%	75.3%	0.51
k=2	55.3%	79.7%	0.59
k=3	54.3%	81.3%	0.62
Full	58.3%	85.3%	0.70





# 实例1

- 第一步: 输入 [15], 选择雅加达
- 第二步: 点击 [60]
- 第三步: 点击 [63]
- 第四步: 点击 [99]
- 第五步: 向下滚动页面
- 第六步: 向上滚动页面
- 第七步: 点击 [41]
- 第八步: 点击 [54]
- 第九步: 提供答案



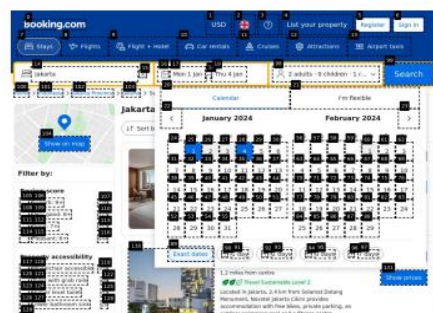
Step 1: Type [15]; Jakarta



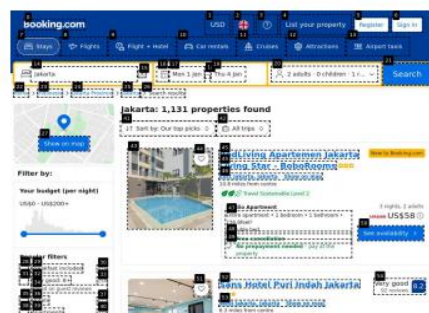
Step 2: Click [60]



Step 3: Click [63]



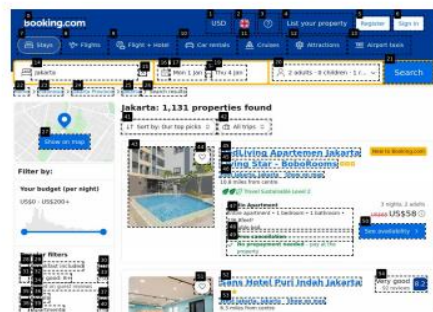
Step 4: Click [99]



Step 5: Scroll down



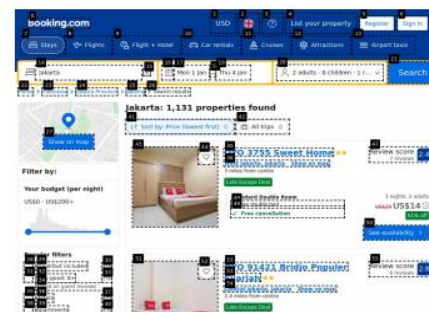
Step 6: Scroll up



Step 7: Click [41]



Step 8: Click [54]

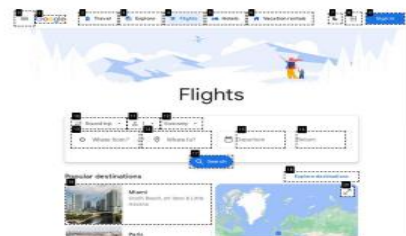


Step 9: ANSWER

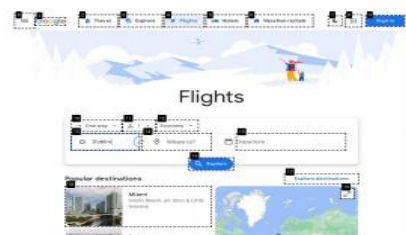


# 实例2

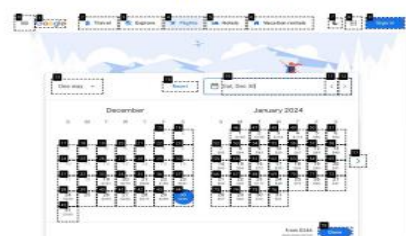
步骤 1: 点击链接 [10]  
步骤 2: 接着点击链接 [12]  
步骤 3: 输入城市名 [13]“都柏林”  
步骤 4: 再输入 [14]“希腊雅典”  
步骤 5-11: 通过连续点击链接  
[16], [41], [78], [19], [34], [2] 来  
查找答案



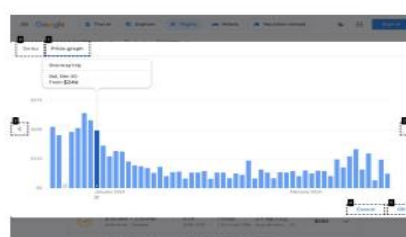
Step 1: Click [10]



Step 4: Type [14]: Athens, Greece



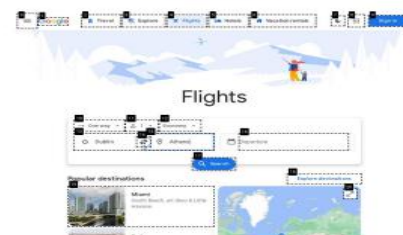
Step 7: Click [78]



Step 10: Click [2]



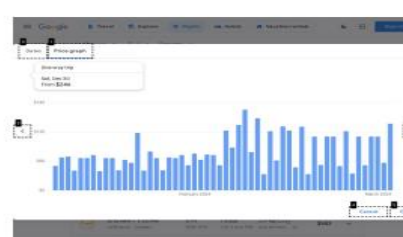
Step 2: Click [12]



Step 5: Click [16]



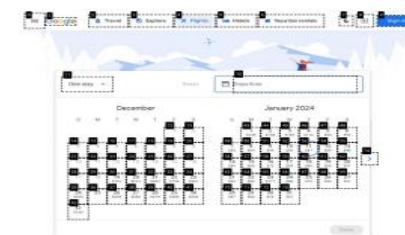
Step 8: Click [19]



Step 11: ANSWER



Step 3: Type [13]: Dublin



Step 6: Click [41]

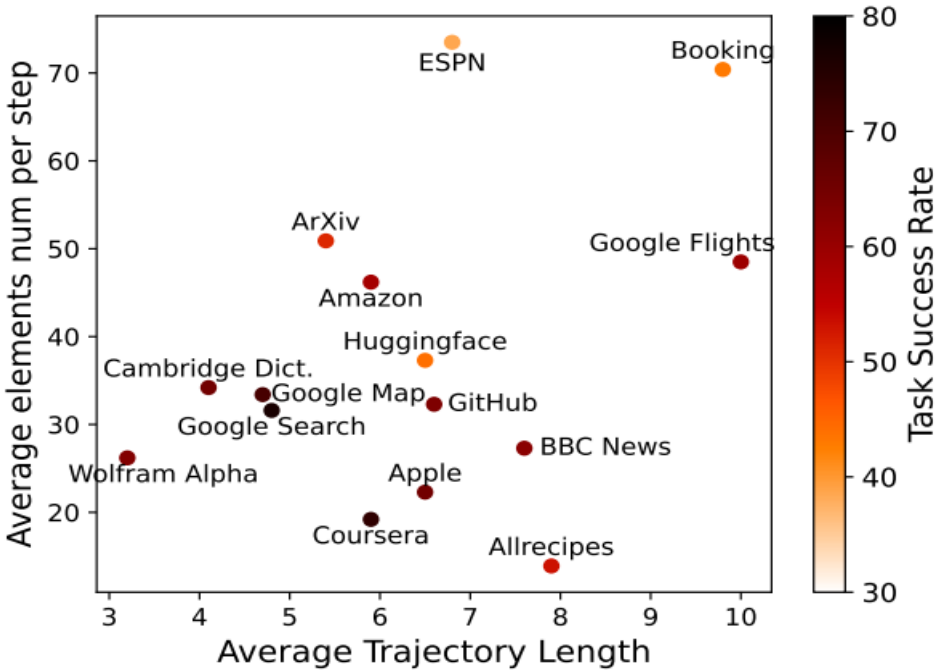


Step 9: Click [34]

# 评估结果与性能对比

- **任务成功率：** WebVoyager 任务成功率为**59.1%**，显著高于基线模型**40.1%**。
- **优势：** 多模态输入提高了 WebVoyager 在复杂任务中的表现，尤其在**视觉复杂任务**中更为突出。

Main reasons for Failure	Ratio
Navigation Stuck	44.4%
Visual Grounding Issue	24.8%
Hallucination	21.8%
Prompt Misalignment	9.0%





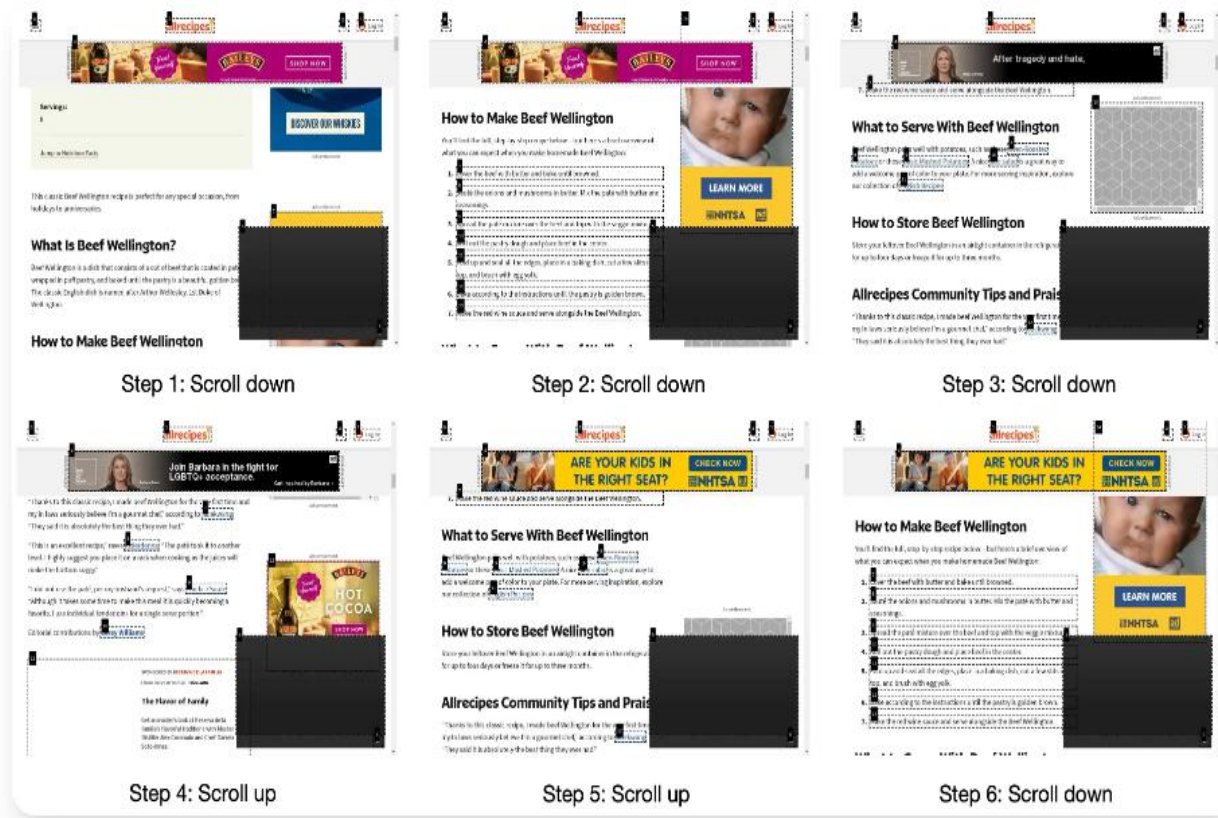
1. 研究背景
2. WebVoyager的系统架构
3. 数据集与自动评估方案
4. 局限与未来发展

# WebVoyager的局限性

## 导航停滞：

当任务涉及多个步骤时，WebVoyager可能由于网页结构复杂、加载延迟等原因导致**导航停滞**，无法顺利完成任务。

Allrecipes 网站的一个错误实例。任务是：“查找 Allrecipes 上的威灵顿牛排食谱，要求评论不少于 200 条，平均评分至少为 4.5 星或以上。列出制作这道菜所需的主要食材。”结果智能体未能正确浏览网页，也没找到食材列表。



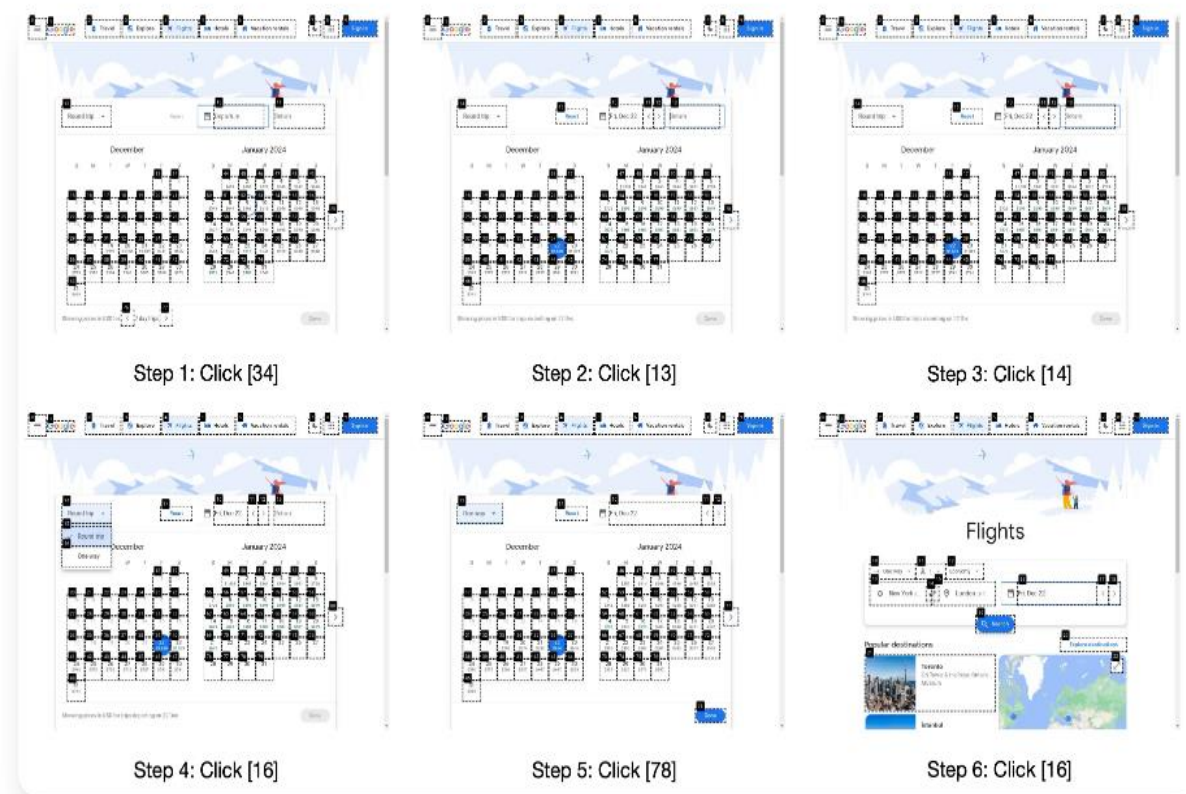


# WebVoyager的局限性

## 视觉定位问题：

WebVoyager在复杂视觉场景中的定位准确性较低，尤其在网页中有多个视觉元素时，容易出现错误的元素选择。

Google Flights 操作过程中的一个错误案例。任务描述是：“在所有符合条件的单程航班中寻找最低票价。”对于一位成人从 JFK 飞往希思罗的航班，日期定在 1 月 22 日。但是，智能体在选择日期时出错，没有选中 1 月 22 日。

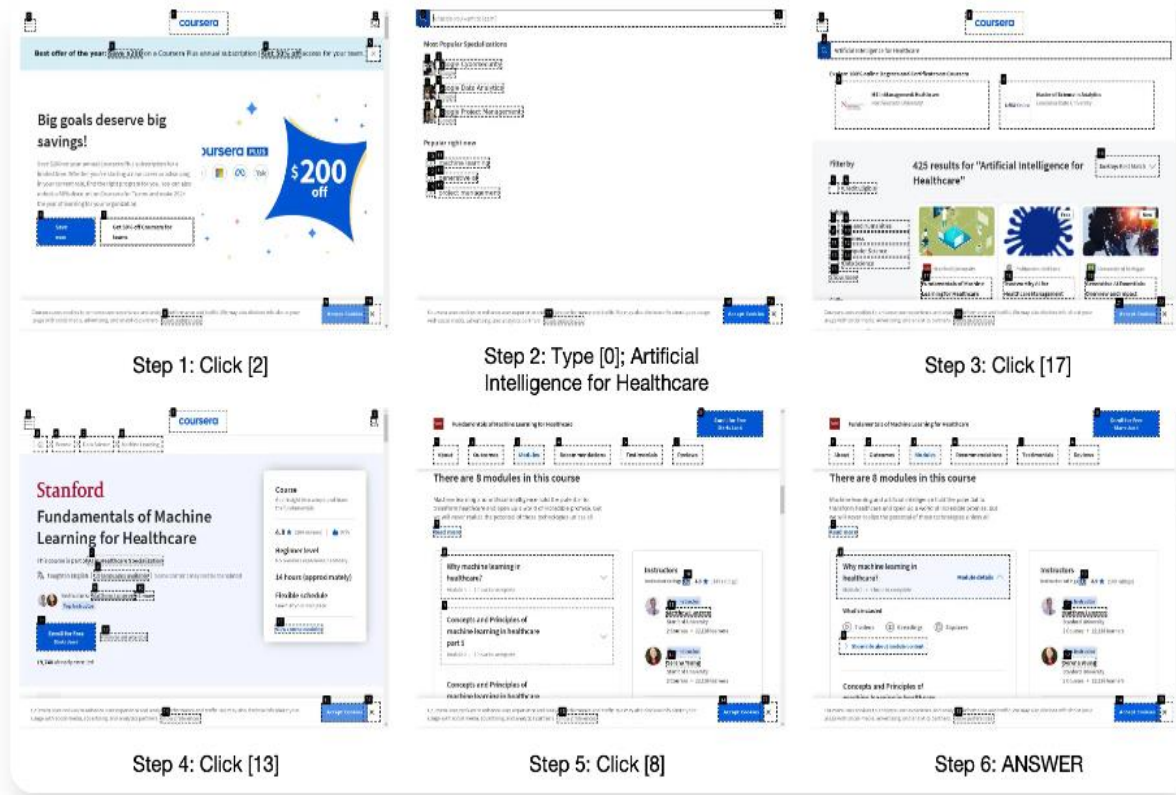


# WebVoyager的局限性

## 幻觉错误：

模型可能会给出不符合实际网页内容的错误回答或操作，特别是在推理过程中产生了错误的判断。

Coursera 网站的一个错误实例。任务是：“在 Coursera 上找到一门关于医疗保健领域人工智能的课程，并记录下课程时长及评估中的测验数量。”结果：“第一模块包含评估中的 3 个测验。”但它未能提供其他模块的信息。



# 未来发展方向

- **强化视觉定位：**未来可通过增强视觉识别模块，提升WebVoyager在视觉密集型任务中的表现，减少定位错误。
- **改进上下文管理：**在长任务链中，优化WebVoyager对任务上下文的理解和管理，避免信息丢失，提高任务执行的连续性。
- **增强任务推理：**通过增强多步推理能力，减少幻觉错误，提高任务完成的准确性。

signals have also been explored. Wei et al., 2023) combines TS (Raffel et al., 2023) with a Vision Transformer (ViT) to na- ture screenshots and HTML text. PIX2 et al., 2023) instead solely relies on screenshots as inputs to predict agent actions. vious works that only consider a single or simplified web environments, we build a modal agent that can complete tasks on websites in this work. Concurrently with SeeAct (Zheng et al., 2024) also leverages Multimodal Models (LMMs) for inter- preting and actions on websites. The best SeeAct agent still relies on a cross-encoder model to select candidate for interaction. In contrast, WebVoyager requires any additional modules.

### 3 WebVoyager

We aim to build an agent that can browse the web autonomously without human intervention. Given an user instruction, our WebVoyager first instantiates a web browser and then performs actions with visual and textual (i.e., HTML) elements from the web. The agent produces an action on the inputs at every step, which is then used by the browser environment. The process continues until the agent decides to stop. The WebVoyager, including environment, inter- action space, and action space, an

#### 3.1 Browsing Environment

We develop an automated web-brows- ing environment using Selenium<sup>1</sup>. Unlike WebA et al., 2023), we do not host any web and allow the agent to explore the open web which poses unique challenges such as pop-up windows, constant updates, etc. For online interaction with real websites, we believe that this setting truly reflects the use cases (e.g., the agent needs access information from the web), and a suc- cessful agent should be able to adapt to these and consistently solve the problem on

<sup>1</sup><https://www.selenium.dev/>

<sup>2</sup>Regarding CAPTCHAs (Completely Auto- mated Turing test to tell Computers and Humans Apart), we believe it is important to respect the rules of the web and prompt the agent to retrieve information from sources.

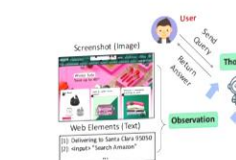


Figure 1: The overall workflow of WebVoyager. WebVoyager browses the web online. At each step, WebVoyager selects a web element and its contents. Once the task is completed, a user query: 'Find the cost of a 2-year protection for a PS4, identifies the 2-year protection price, a

making for WebVoyager.

Another challenge is the evaluation of an end-to-end web agent. Existing benchmarks, such as Mind2Web (Deng et al., 2023), primarily focus on stepwise and offline evaluation, where agents follow a predefined "golden" trajectory for action selection. This approach, however, may not fully account for the variety of viable strategies to accomplish a task, as it only reflects one possible plan. This limitation could lead to a biased evaluation and difficulties in fairly comparing different methods. To accurately evaluate the capabilities of web agents in end-to-end task completion, we propose an automated evaluation protocol using GPT-4V. Specifically, we save screenshots throughout the online navigation process and then use GPT-4V to evaluate these trajectories together with the final results automatically. Human evaluations are also conducted to verify the results and the analysis shows that our evaluation protocol achieves 85.3% agreement with human judges, indicating GPT-4V can serve as a reliable evaluator for online agents.

We conduct evaluations on a newly collected dataset, which is semi-automatically generated using a self-instruct (Wang et al., 2022) method, comprising 643 web tasks from 15 commonly accessed websites. We also evaluate WebVoyager on 90 web-related tasks of level 1 and level 2 from the GAIA (Malon et al., 2023), and 50 interactive open-web tasks from SeeAct (Zheng et al., 2024). We compare our WebVoyager with 1) GPT-4 (All Tools)<sup>2</sup>, and 2) WebVoyager in a text-only setting which employs the textual accessibility tree proposed in WebArena (Zhou et al., 2023) to describe web pages.

<sup>2</sup>GPT-4 (All Tools) is an integrated tool-based agent released by OpenAI in Oct. 2023. See <https://chat.openai.com/>

## WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models

Hongliang He<sup>1,3\*</sup>, Wenlin Yao<sup>2</sup>, Kaixin Ma<sup>2</sup>, Wenhao Yu<sup>2</sup>, Yong Dai<sup>2</sup>,  
Hongming Zhang<sup>2</sup>, Zhenzhong Lan<sup>1</sup>, Dong Yu<sup>2</sup>  
<sup>1</sup>Zhejiang University, <sup>2</sup>Tencent AI Lab, <sup>3</sup>Westlake University  
hehongliang@westlake.edu.cn, wenlinyao@global.tencent.com

### Abstract

The rapid advancement of large language models (LLMs) has led to a new era marked by the development of autonomous applications in real-world scenarios, which drives innovation in creating advanced web agents. Existing web agents typically only handle one input modality and are evaluated only in simplified web simulators or static web snapshots, greatly limiting their applicability in real-world scenarios. To bridge this gap, we introduce WebVoyager, an innovative Large Multimodal Model (LMM) powered web agent that can complete user instructions end-to-end by interacting with real-world websites. Moreover, we establish a new benchmark by compiling real-world tasks from 15 popular websites and introduce an automatic evaluation protocol leveraging multimodal understanding abilities of GPT-4V to evaluate open-ended web agents. We show that WebVoyager achieves a 59.1% task success rate on our benchmark, significantly surpassing the performance of both GPT-4 (All Tools) and the WebVoyager (text-only) setups, underscoring the exceptional capability of WebVoyager. The proposed automatic evaluation metric achieves 85.3% agreement with human judgment, indicating its effectiveness in providing reliable and accurate assessments of web agents.<sup>1</sup>

### 1 Introduction

The recent advancement of large language models (LLMs), such as ChatGPT and GPT-4 (OpenAI, 2023), have sparked significant interest in developing LLM-based autonomous agents (AutoGPT, 2022) for complex task execution (Qin et al., 2023; Schick et al., 2023). Recent studies have explored the construction of text-based web browsing environments and how to instruct large language model agents to perform web navigation (Nakano et al., 2021; Gur et al., 2023; Zhou et al., 2023; Lu et al.,

<sup>\*</sup>Work done during the internship at Tencent AI Lab.  
<sup>1</sup>Our code and data will be released at <https://github.com/MinorJerry/WebVoyager>.

2023). The primary challenge in these works lies in managing complex and verbose HTML texts, and solutions include simplifying and structuring HTML (Nakano et al., 2021; Zhou et al., 2023; Gur et al., 2023; Deng et al., 2023).

However, existing approaches overlook a critical functionality of browsing: rendering HTML into visual webpages. Particularly, vision capability is crucial for utilizing tools such as web browsers, as rendered web pages are inherently designed with user experience (UX), emphasizing intuitive information and structured presentation. This design principle of rendering makes visual analysis more effective than mere HTML representation. At present, large multimodal models (LMMs), particularly GPT-4V (OpenAI, 2023) and Gemini (Team et al., 2023), demonstrate a remarkable ability to integrate intricate visual cues with textual information. Existing studies such as Pix2Struct (Lee et al., 2023) and WebArena (Zhou et al., 2023), have initiated explorations into using screenshots as inputs for decision-making in web navigation, yet these are preliminary and do not represent a deep exploration. Therefore, building multimodal web agents to leverage the environment rendered by browsers through screenshots, thus mimicking human web browsing behavior, is now a viable approach to enhance web navigation abilities.

We introduce WebVoyager (Figure 1), a multimodal web agent designed to autonomously accomplish web tasks online from start to finish, managing the entire process end-to-end without any intermediate human intervention. WebVoyager processes the user query by making observations from screenshots and textual content in interactive web elements, formulates a thought on what action to take (such as clicking, typing, or scrolling, etc.), and then executes that action on the websites. Inspired by Set-of-Mark Prompting (Yang et al., 2023a), we mark interactive web elements on screenshots (see Figure 2) to facilitate decision-



## 总结与展望

- **现有优势：**WebVoyager在多模态输入和任务成功率上具有显著优势，尤其在视觉密集型任务中表现出色。
- **持续改进：**通过加强视觉定位、任务推理和上下文管理，WebVoyager将能够更好地适应复杂网页任务。
- **长远应用：**未来WebVoyager可扩展至更多复杂的网页任务和应用场景，成为通用的网页自动化工具，推动智能代理技术的发展。







# Thanks

汇报人：袁沐寒  
2024年11月 22 日