PLAN-AND-ACT: Improving Planning of Agents for Long-Horizon Tasks

PLAN-AND-ACT: 提升智能体在长时任务中的规划能力

Lutfi Eren Erdogan [*1] Nicholas Lee [*1] Sehoon Kim [1] Suhong Moon [1] Hiroki Furuta [2] Gopala Anumanchipalli [1] Kurt Keutzer [1] Amir Gholami [13]

Lutfi Eren Erdogan [*1] Nicholas Lee [*1] Sehoon Kim [1] Suhong Moon [1] Hiroki Furuta [2] Gopala Anumanchipalli [1] Kurt Keutzer [1] Amir Gholami [13]

Abstract

摘要

Large language models (LLMs) have shown remarkable advancements in enabling language agents to tackle simple tasks. However, applying them for complex, multi-step, long-horizon tasks remains a challenge. Recent work have found success by separating high-level planning from low-level execution, which enables the model to effectively balance high-level planning objectives and low-level execution details. However, generating accurate plans remains difficult since LLMs are not inherently trained for this task.

大型语言模型 (LLMs) 在使语言智能体完成简单任务方面展现了显著进展。然而，将其应用于复杂的、多步骤的长时任务仍然具有挑战性。近期研究通过将高层规划与低层执行分离取得了成功，这使模型能够有效平衡高层规划目标与低层执行细节。然而，由于 LLMs 本身并非为此任务训练，生成准确的计划仍然困难。

To address this, we propose PLAN-AND-ACT, a novel framework that incorporates explicit planning into LLM-based agents and introduces a scalable method to enhance plan generation through a novel synthetic data generation method. PLAN-AND-ACT consists of a PLANNER model which generates structured, high-level plans to achieve user goals, and an EXECUTOR model that translates these plans into environment-specific actions. To train the PLANNER effectively, we introduce a synthetic data generation method that annotates ground-truth trajectories with feasible plans, augmented with diverse and extensive examples to enhance generalization. We evaluate PLAN-AND-ACT using web navigation as a representative long-horizon planning environment, demonstrating a state-of-the-art 57.58% success rate on the WebArena-Lite benchmark as well as a text-only state-of-the-art 81.36% success rate on WebVoy-ager.

为此，我们提出了 PLAN-AND-ACT，一种将显式规划融入基于 LLM 的智能体的新框架，并引入了一种通过新颖的合成数据生成方法提升计划生成能力的可扩展方法。PLAN-AND-ACT 包含一个 PLANNER 模型，用于生成结构化的高层计划以实现用户目标，以及一个 EXECUTOR 模型，将这些计划转化为特定环境的动作。为了有效训练 PLANNER，我们引入了一种合成数据生成方法，该方法为真实轨迹注释可行计划，并通过多样且丰富的示例增强泛化能力。我们以网页导航作为代表性的长时规划环境对 PLAN-AND-ACT 进行了评估，展示了在 WebArena-Lite 基准上的最新 57.58% 成功率，以及在 WebVoy-ager 文本-only 任务上达到 81.36% 的最新成功率。

1. Introduction

1. 引言

Large language models (LLMs) have significantly advanced in capability, enabling their application as language agents that can interact with environments through sequences of

大型语言模型 (LLMs) 能力显著提升，使其能够作为语言智能体，通过一系列动作与环境交互

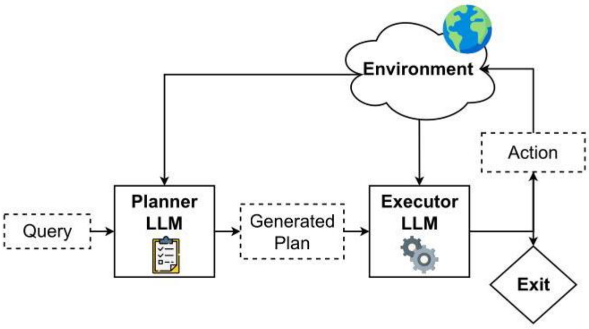


Figure 1. An illustration of PLAN-AND-ACT System Diagram. First, the PLANNER LLM processes the initial user query and generates an initial step by step plan (Section 3.1). This is then passed to the EXECUTOR LLM which uses the plan and generates an actions to interact with its Environment. The environment feedback is then fed back to both the EXECUTOR so it can generate subsequent actions and/or to the PLANNER in case a new plan needs to be generated. Existing methods have shown this separation of high-level planning and low-level execution can improve accuracy. However, a major challenge is that LLMs are not generally trained to generate such plan/low-level action, a problem that we focus on solving in this paper.

图 1. PLAN-AND-ACT 系统示意图。首先，PLANNER LLM 处理初始用户查询并生成逐步的初始计划 (第 3.1 节)。随后该计划传递给 EXECUTOR LLM，后者根据计划生成与环境交互的动作。环境反馈随后被送回 EXECUTOR 以生成后续动作，或在需要生成新计划时反馈给 PLANNER。现有方法表明，高层规划与低层执行的分离能提升准确性。然而，LLMs 通常未被训练以生成此类计划/低层动作，这是本文重点解决的问题。

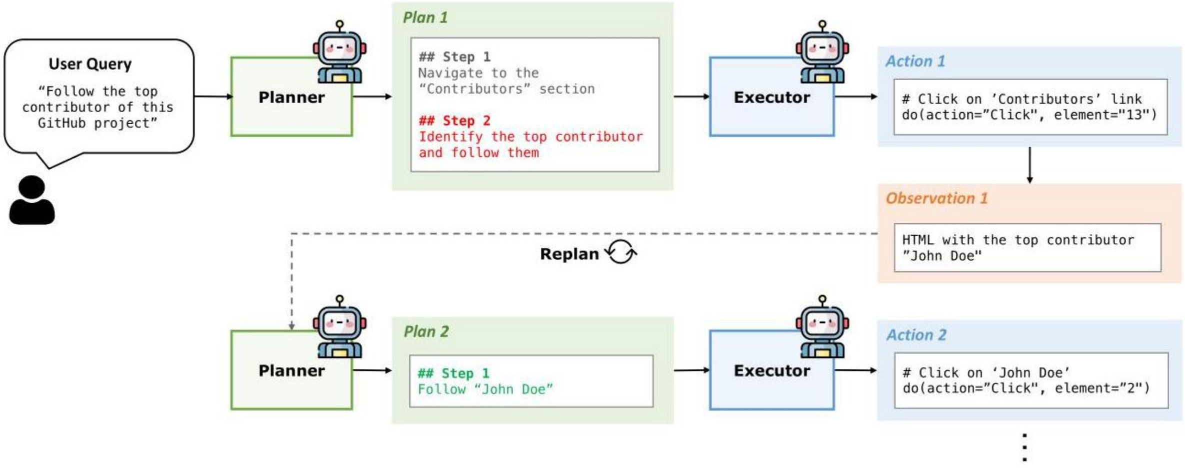
actions. These agents are designed to tackle complex, multistep, long-horizon tasks by leveraging the model's reasoning and decision-making capabilities. At the heart of building such effective agents lies a fundamental challenge: planning. Even for seemingly simple tasks, an agent must understand the goal, break it down into manageable steps, and adapt those steps as circumstances change. However, despite these advancements, planning remains a significant challenge for several reasons. First, agents often struggle to break down high-level user goals (like ”book me a flight to New York”) into specific, actionable steps (like ”open the airline web-site”, ”enter travel dates”, etc.). Second, as tasks grow longer and more complex, maintaining a coherent strategy becomes increasingly difficult - agents lose track of what they've accomplished and what remains to be done. Third, real-world environments are dynamic and unpredictable, requiring agents to constantly revise their plans. These challenges are further amplified by the scarcity of high-quality training data that demonstrate effective planning strategies.

动作。这些智能体旨在利用模型的推理和决策能力，处理复杂的多步骤长时任务。构建此类高效智能体的核心挑战在于规划。即使是看似简单的任务，智能体也必须理解目标，将其拆解为可管理的步骤，并根据环境变化调整这些步骤。然而，尽管取得了进展，规划仍面临诸多挑战。首先，智能体常难以将高层用户目标 (如“帮我订一张飞往纽约的机票”) 拆解为具体可执行的步骤 (如“打开航空公司网站”、“输入旅行日期”等)。其次，随着任务变得更长更复杂，保持连贯策略愈发困难——智能体可能忘记已完成的内容和待完成的任务。第三，现实环境动态且不可预测，要求智能体不断修正计划。这些挑战因缺乏展示有效规划策略的高质量训练数据而更加突出。

---

*Equal contribution [1] UC Berkeley [2] University of Tokyo [3] ICSI. Correspondence to: Amir Gholami <amirgh@berkeley.edu>.

* 同等贡献 [1] 加州大学伯克利分校 [2] 东京大学 [3] ICSI。通讯作者:Amir Gholami <amirgh@berkeley.edu>。


User Query
“Follow the top contributor of this GitHub project”
Planner
Plan 1
## Step 1
Navigate to the “Contributors” section
## Step 2
Identify the top contributor and follow them
Executor
Action 1
Observation 1
HTML with the top contributor "John Doe"
Replan
Planner
Plan 2
## Step 1
Follow “John Doe”
Executor
Action 2

```
# Click on ’Contributors’ link
do(action="Click", element="13")
```

```
# Click on ’John Doe’
do(action="Click", element="2")
```

Figure 2. PLAN-AND-ACT System Diagram. Given the initial user query, the PLANNER (Section 3.1) breaks it down into a high-level plan, which is given to the EXECUTOR (Section 3.2) which uses the plan to guide its actions. Once the action has been taken and the HTML changes, the PLANNER dynamically generates a new plan that incorporates the changes in the environment (Section 3.3).

图 2. PLAN-AND-ACT 系统示意图。给定初始用户查询，PLANNER(第 3.1 节) 将其拆解为高层计划，传递给 EXECUTOR(第 3.2 节)，后者依据计划指导动作。动作执行后，HTML 发生变化，PLANNER 动态生成包含环境变化的新计划 (第 3.3 节)。

Previous approaches to improve agent performance for long-range tasks such as web navigation [35, 54, 13, 60] and device control [5, 33] have shown promise. However, most rely on a single model to directly translate user requests into actions. This creates a difficult balancing act - the model must simultaneously reason about the high-level strategy while managing the low-level execution details [54]. Under this load, models often lose sight of their ultimate objectives and struggle to maintain consistent behavior [42]. Recent work has explored the use of Reinforcement Learning (RL) [44] to improve performance [2, 35], but these methods can be unstable and highly sensitive to hyperparameters and reward design [36, 8].

以往提升代理在长程任务 (如网页导航 [35, 54, 13, 60] 和设备控制 [5, 33]) 中表现的方法显示出一定潜力。然而, 大多数方法依赖单一模型直接将用户请求转化为动作。这带来了艰难的平衡——模型必须同时推理高层策略并管理低层执行细节 [54]。在这种负载下，模型常常忽视最终目标，难以保持行为一致性 [42]。近期研究探索了使用强化学习 (Reinforcement Learning, RL)[44] 提升性能 [2, 35], 但这些方法可能不稳定且对超参数和奖励设计高度敏感 [36, 8]。

To equip LLMs to effectively plan for long-horizon tasks more reliably, we introduce PLAN-AND-ACT, a framework that incorporates explicit planning into LLM-based agents. Unlike traditional approaches like ReAct based methods [35, 13, 55] that rely on a single model to directly map user queries to a sequence of actions, PLAN-AND-ACT adopts a framework similar to LLMCompiler [19] which consists of two modules: a PLANNER and an EXECUTOR (Figure 1). The PLANNER model generates a sequence of plans that outline the high-level steps required to achieve the goal, while the Executor translates these steps into environment-specific actions. Furthermore, our framework provides an effective and scalable solution for generating training data to train the PLANNER without requiring manual annotation or a sandbox environment. The difference with LLMCompiler is that we introduce a scalable method that allows finetuning the PLANNER and EXECUTOR components. In particular, our contributions are as follows:

为了使大型语言模型 (LLMs) 更可靠地规划长远任务，我们提出了 PLAN-AND-ACT 框架，该框架将显式规划引入基于 LLM 的代理。不同于依赖单一模型直接将用户查询映射为动作序列的传统方法，如 ReAct 方法 [35, 13, 55]，PLAN-AND-ACT 采用类似 LLMCompiler[19] 的框架，由两个模块组成: 规划器 (PLANNER) 和执行器 (EXECUTOR)(见图 1)。规划器生成实现目标所需的高层步骤序列，执行器则将这些步骤转化为特定环境的动作。此外，我们的框架提供了一种有效且可扩展的训练数据生成方案，用于训练规划器，无需人工标注或沙箱环境。与 LLMCompiler 的区别在于，我们引入了一种可扩展方法，允许对规划器和执行器组件进行微调。具体贡献如下:

- We propose PLAN-AND-ACT, a framework that improves planning for long-horizon tasks through explicit separation of planning and execution. As shown in Figure 2, our architecture consists of a PLANNER that breaks down user requests into structured plans, and an EXECUTOR that implements these plans through environment-specific actions (Section 3).

- 我们提出了 PLAN-AND-ACT 框架，通过显式分离规划与执行，提升长远任务的规划能力。

- 如图 2 所示，我们的架构包括一个将用户请求分解为结构化计划的规划器，以及一个通过环境特定动作实现这些计划的执行器 (第 3 节)。

- To train the PLANNER model effectively, we introduce a synthetic data generation pipeline to generate planner data with and without access to extra ground truth data. First, we use an LLM to analyze successful action trajectories (sequences of actions like clicking, typing, etc.) and generate the corresponding high-level plans through grounded plan generation, ensuring these plans are grounded in actual executable actions (Section 4.2). Second, we synthetically augment our dataset by using these initial plans as seed data to generate additional diverse planning examples (Section 4.3). This comprehensive approach enables us to create high-quality training data despite the scarcity of real-world planning examples.

- 为有效训练规划器模型，我们引入了合成数据生成流程，支持有无额外真实数据的规划器数据生成。首先，利用 LLM 分析成功的动作轨迹 (如点击、输入等动作序列)，通过基于实际可执行动作的计划生成，生成对应的高层计划 (第 4.2 节)。其次，利用这些初始计划作为种子数据，合成扩充数据集，生成更多多样化的规划示例 (第 4.3 节)。该综合方法使我们在真实规划示例稀缺的情况下，仍能创建高质量训练数据。

- To demonstrate the efficacy of our approach on long-horizon tasks, we evaluate PLAN-AND-ACT in the WebArena-Lite [25] environment for web navigation, achieving SOTA result of 53.94% (Table 1).

- 为验证方法在长远任务中的有效性，我们在 WebArena-Lite[25] 网页导航环境中评估 PLAN-AND-ACT，取得了 53.94% 的最新最佳成绩 (表 1)。

2. Related Work

2. 相关工作

2.1. Language Agents as Web Agents

2.1. 语言代理作为网页代理

As LLM agents become more widespread, they have been increasingly applied as web agents in order to traverse and operate web pages [12, 18] through the GUI interaction [33, 37, 58, 52, 59] or API interaction [41], which encourages the recent introduction of datasets [38, 4, 37] and benchmarks [56, 61, 20, 25]. Some works involve intricately

随着大型语言模型代理的普及，它们越来越多地被用作网页代理，通过图形用户界面 (GUI) 交互 [33, 37, 58, 52, 59] 或 API 交互 [41] 遍历和操作网页 [12, 18]，这推动了相关数据集 [38, 4, 37] 和基准测试 [56, 61, 20, 25] 的出现。一些工作设计了复杂的提示代理系统协同导航网页 [31, 3, 9, 54, 60, 7, 39, 21, 34, 43]。

designed systems of prompted agents working together to navigate the web [31, 3, 9, 54, 60, 7, 39, 21, 34, 43]. Prior work such as [54, 60, 43, 34] use a hierarchical planning framework in a similar direction to PLAN-AND-ACT, but are all prompting methods using closed-source models such as GPT-4o as their base model. In contrast PLAN-AND-ACT provides a simple, systematic way to generate high quality training data to train LLMs on web tasks.

先前工作如 [54, 60, 43, 34] 采用了与 PLAN-AND-ACT 类似方向的分层规划框架，但均为基于闭源模型 (如 GPT-4) 的提示方法。相比之下，PLAN-AND-ACT 提供了一种简单系统的方法，生成高质量训练数据以训练 LLM 完成网页任务。

In addition, our method uses a very simple 2-agent framework, which is significantly simpler compared to other prior work with planning. AgentOccam [54] uses a ”Planning via Generation” technique where the planning is incorporated into the action space and the model plans in a tree-like fashion. WebPilot [60] has a significantly more complex infrastructure with 6 different agents in total. AdaPlanner [43] has an In-Plan and Out-of-Plan Refiner to facilitate replanning when the plan is wrong and a skill-discovery module that is orthogonal to our method and can be used in conjunction. ADaPT [34] uses recursive decomposition to decomposes tasks when the executor fails, whereas our dual-agent architecture simply replans at each step. All of these methods use more excessive prompting to improve performance, while our method has a simple PLAN-AND-ACT structure at runtime.

此外，我们的方法采用了一个非常简单的双智能体框架，相较于其他带有规划的先前工作显著简化。AgentOccam [54] 使用“通过生成进行规划”的技术，将规划融入动作空间，模型以树状方式进行规划。WebPilot [60] 拥有一个复杂得多的基础设施，总共包含 6 个不同的智能体。AdaPlanner [43] 设有计划内和计划外的细化器，以便在计划错误时重新规划，并且拥有一个与我们方法正交且可结合使用的技能发现模块。ADaPT [34] 采用递归分解，当执行器失败时分解任务，而我们的双智能体架构则是在每一步简单地重新规划。所有这些方法都使用了更多的提示来提升性能，而我们的方法在运行时采用了简单的“规划-执行”结构。

Other work that have discussed generating training data for Web Agents such as DigiRL [2], WebRL [35], AutoWebGLM [23], and NNetNav [28] provide more complex techniques for collecting diverse trajectories, which are complementary to PLAN-AND-ACT, as our pipeline (Section 4.1) is simple and can be interchanged. Furthermore, they only produce trajectory data, but do not planning data (Section 4.2). They also rely on external simulators to generate data, whereas our method can generate synthetic planning data without a simulator (Section 4.3).

其他讨论为 Web 智能体生成训练数据的工作，如 DigiRL [2]、WebRL [35]、AutoWebGLM [23] 和 NNetNav [28]，提供了更复杂的技术来收集多样化的轨迹，这些技术与“规划-执行”方法互为补充，因为我们的流程 (第 4.1 节) 简单且可替换。此外，它们仅生成轨迹数据，而不生成规划数据 (第 4.2 节)。它们还依赖外部模拟器生成数据，而我们的方法可以在无模拟器的情况下生成合成规划数据 (第 4.3 节)。

Some other prior work in robotics [40, 29, 17] use hierarchical LLM Agents to decompose tasks and plan for robotic-s/embodied agents which shares some similarity with our work. However, similar to other prior planning based web agents, they do not contain a framework for collecting or generating synthetic data for training open source LLMs.

机器人领域的一些先前工作 [40, 29, 17] 使用分层大型语言模型 (LLM) 智能体来分解任务并为机器人/具身智能体规划，这与我们的工作有一定相似性。然而，与其他基于规划的网页智能体类似，它们并未包含用于收集或生成用于训练开源 LLM 的合成数据的框架。

The other agents have pretrained LLMs on HTML to better understand webpages [13], have used the vision capabilities of multi-modal vision and language models (VLMs) to augment web agents with visual understanding of the webpage [6], or adopt RL to improve the performance of these models through interaction [35, 2, 55].

其他智能体则通过在 HTML 上预训练 LLM 以更好地理解网页 [13]，利用多模态视觉与语言模型 (VLM) 的视觉能力增强网页智能体对网页的视觉理解 [6]，或采用强化学习 (RL) 通过交互提升模型性能 [35, 2, 55]。

2.2. Synthetic Data Generation

2.2. 合成数据生成

Synthetic generation has seen a surge in popularity in recent years. Beginning with pioneering work such as Self-Instruct [49] and Alpaca [45], many recent papers have used

近年来，合成生成技术迅速流行。始于 Self-Instruct [49] 和 Alpaca [45] 等开创性工作，许多近期论文利用

the power of synthetic data generation and augmentation to boost the performance of LLMs [53, 24, 5, 47, 10, 27].

合成数据生成与增强的力量来提升大型语言模型 (LLM) 的性能 [53, 24, 5, 47, 10, 27]。

In the context of LLMs as web-agents, some papers use existing, reusable environments [61, 20, 25] to collect training data trajectories from their web-agents in an on-policy fashion [35, 55, 32, 30]. A common pattern is to collect new trajectories from an LLM and to filter the trajectories for failed instances. Patel et al. [32] mixed real and synthetic data from WebArena [61] to improve the performance of their model. Yang et al. [55] executed multiple trajectories on failed tasks to supplement the training data. NNetscape Navigator [28] is an interaction-first technique that explored websites and retroactively labeled trajectories with instructions in order to generate more training data. WebRL [35] used a Self-Instruct style prompt to generate synthetic user queries which were used to collect trajectories with their model, which were labeled with an ORM. Notably, similar to Lee et al. [24], failed trajectories were used to seed the user instruction generation pipeline which is then used to collect more targeted execution trajectories.

在将 LLM 作为网页智能体的背景下，一些论文使用现有的可复用环境 [61, 20, 25] 以在线策略方式收集网页智能体的训练数据轨迹 [35, 55, 32, 30]。一个常见模式是从 LLM 收集新轨迹并筛选失败实例。Patel 等人 [32] 混合了来自 WebArena [61] 的真实和合成数据以提升模型性能。Yang 等人 [55] 在失败任务上执行多条轨迹以补充训练数据。NNetscape Navigator [28] 是一种以交互为先的技术，探索网站并事后用指令标注轨迹以生成更多训练数据。WebRL [35] 使用 Self-Instruct 风格的提示生成合成用户查询，进而用其模型收集轨迹，并用 ORM 标注。值得注意的是，类似于 Lee 等人 [24]，失败轨迹被用来启动用户指令生成流程，随后用于收集更有针对性的执行轨迹。

3. System Architecture

3. 系统架构

As discussed in previous work [42, 54], at the heart of effective agents lies the challenge of balancing high-level reasoning with low-level execution. When a single model must simultaneously perform long horizon planning and then also execute multiple low-level actions for each part of the plan, it faces a difficult cognitive load that often leads to suboptimal decisions or inconsistent behavior. This challenge becomes especially acute for long-horizon tasks, where the agent must maintain a coherent strategy across many steps while adapting to changes in the environment.

如先前工作所述 [42, 54]，高效智能体的核心挑战在于平衡高层推理与低层执行。当单一模型必须同时执行长远规划并为计划的每个部分执行多个低层动作时，面临的认知负担极大，常导致次优决策或行为不一致。该挑战在长远任务中尤为突出，智能体必须在多步中保持连贯策略，同时适应环境变化。

To address this fundamental challenge, our framework separates these responsibilities into two specialized components: a PLANNER that focuses on strategic decision-making and an EXECUTOR that specializes in implementing those decisions (Figure 2). This separation allows each component to excel at its core task. The PLANNER can reason about high-level strategy without getting bogged down in implementation details, while the EXECUTOR can focus on translating abstract plans into concrete actions [19, 48].

为应对这一根本挑战，我们的框架将职责分为两个专门组件：专注于战略决策的规划器 (PLANNER) 和专注于执行决策的执行器 (EXECUTOR)(见图 2)。这种分离使每个组件能专注于其核心任务。规划器可专注于高层策略推理而不被实现细节干扰，执行器则专注于将抽象计划转化为具体行动 [19, 48]。

While our framework is adaptable to various structured decision-making environments, we focus on web agents due to the web's dynamic and complex nature, which involves diverse actions and long-horizon tasks. For web tasks, the PLANNER takes a user query (like ”Follow the top contributor of this GitHub project”) and breaks it down into clear, manageable steps (such as ”Navigate to the Contributors section” followed by ”Identify and follow the top contributor”). The EXECUTOR then takes these steps and translates them into precise actions in the web environment,

虽然我们的框架可适用于各种结构化决策环境，但我们重点关注网页代理，因网页环境动态且复杂，涉及多样化操作和长远任务。对于网页任务，规划器 (PLANNER) 接收用户查询 (如“关注此 GitHub 项目的顶级贡献者”)，并将其分解为清晰且可管理的步骤 (例如“导航至贡献者部分”，然后“识别并关注顶级贡献者”)。执行器 (EXECUTOR) 随后将这些步骤转化为网页环境中的精确操作，

like clicking specific links or typing in search boxes. The observation space that we will use for this task is HTML as the text-representation of the environment.

如点击特定链接或在搜索框中输入内容。我们用于此任务的观察空间是 HTML，作为环境的文本表示。

3.1. PLANNER

3.1. 规划器 (PLANNER)

The PLANNER takes the user query and breaks it down into a structured plan that dictates the essential high level steps required to accomplish the task. This plan is used to as a guide for the EXECUTOR at runtime, providing a clear road-map while allowing some flexibility for the EXECUTOR. By handling most of the reasoning and task decomposition, the PLANNER streamlines decision making and task execution.

规划器接收用户查询并将其分解为结构化计划，指明完成任务所需的关键高层步骤。该计划在运行时作为执行器的指导，提供清晰的路线图，同时允许执行器一定的灵活性。通过承担大部分推理和任务分解，规划器简化了决策和任务执行过程。

Consider the example in Figure 2, which shows an example of this module in action, in the context of a web task. We have the original instruction Follow the top contributor of this GitHub project and the goal is for our WebAgent to execute this task on GitHub. The PLANNER processes this instruction and breaks down the task into a step-by-step plan, consisting of (i) Navigating to the Contributors section and (ii) Identifying and Following the top contributor.

以图 2 中的示例为例，展示了该模块在网页任务中的实际应用。原始指令为“关注此 GitHub 项目的顶级贡献者”，目标是让我们的网页代理在 GitHub 上执行此任务。规划器处理该指令并将任务分解为逐步计划，包括 (i) 导航至贡献者部分和 (ii) 识别并关注顶级贡献者。

3.2. EXECUTOR

3.2. 执行器 (EXECUTOR)

The EXECUTOR is an LLM Agent that takes the plan from the PLANNER (Section 3.1) and runs it in the environment. It is responsible for calling tools, retrieving data, or making changes in the environment required by the plan.

执行器是一个大型语言模型 (LLM) 代理，接收规划器 (第 3.1 节) 生成的计划并在环境中执行。它负责调用工具、检索数据或根据计划对环境进行必要的更改。

For example, in the web task shown in Figure 2, once the PLANNER generates a plan for the query Follow the top contributor of this GitHub project, the EXECUTOR only needs to translate the step Navigate to the ”Contributors” section into a click action on the HTML. As we can see, EXECUTOR takes HTML as an input and outputs a grounded concrete action in the environment. Importantly, after executing an action, the EXECUTOR performs garbage collection, removing unnecessary data such as redundant HTML before executing the next action. More explicit examples of the PLANNER-EXECUTOR trajectories can be found in Appendix A.1.

例如，在图 2 展示的网页任务中，一旦规划器为查询“关注此 GitHub 项目的顶级贡献者”生成计划，执行器只需将“导航至‘贡献者’部分”这一步骤转化为对 HTML 的点击操作。如图所示，执行器以 HTML 为输入，输出环境中具体的实际操作。重要的是，执行器在执行操作后会进行垃圾回收，移除冗余 HTML 等不必要数据，然后再执行下一步操作。更多规划器-执行器轨迹的具体示例见附录 A.1。

3.3. Dynamic Replanning

3.3. 动态重新规划

A key limitation of the previous approach is that the initial plan is static throughout execution, which makes it vulnerable to unexpected variations in the environment. For instance, in the web-navigation example, static plans are unequipped to handle dynamic content interpretation, such as analyzing search results or transaction histories. The EXECUTOR can fail to correctly process content that is unknown a priori at planning time. Furthermore, static plans can have issues with unexpected failures, such as searching a keyword returning nothing. If the plans are static, the EXECUTOR may blindly follow the steps in the original plan rather than trying a different approach. As shown in

> 之前方法的一个关键限制是初始计划在执行过程中保持静态，这使其容易受到环境中意外变化的影响。例如，在网页导航任务中，静态计划无法处理动态内容的解析，如分析搜索结果或交易历史。执行器可能无法正确处理规划时未知的内容。此外，静态计划在遇到意外失败时也存在问题，比如搜索关键词无结果。如果计划是静态的，执行器可能会盲目遵循原计划步骤，而不尝试其他方法。如任务分解的先前研究 [19] 所示，即使是简单任务，静态规划也存在根本缺陷。

previous work in task decomposition [19], static planning can have fundamental drawbacks even in straightforward tasks.

> 先前任务分解工作 [19] 表明，即使在简单任务中，静态规划也存在根本性缺陷。

To address this limitation, we introduce dynamic replanning, where the PLANNER updates the plan after each EXECUTOR step rather than relying solely on the initial plan. After each iteration, the PLANNER takes in the current state as well as the previous plans and actions and generates a new plan for how the EXECUTOR can complete the user instruction.

> 为解决此限制，我们引入动态重新规划，即规划器在每次执行器执行步骤后更新计划，而非仅依赖初始计划。每次迭代后，规划器接收当前状态以及之前的计划和操作，生成执行器完成用户指令的新计划。

Conveniently, dynamic replanning allows the planner to retain key information within the evolving plan. For instance, consider the example in Figure 2. The original plan did not know who the top contributor was, so it could only contain the step Identify the top contributor and follow them. After the contributor was identified upon execution of the action clicking ”Contributors” link, the PLANNER incorporates this information into the remaining plan. Since the plan carries forward the relevant context, this approach also allows us to address challenges related to memory for long-horizon tasks without requiring an explicit memory module [54, 60]. More detailed examples of dynamic replanning can be found at Appendix A.2.

> 动态重新规划使规划器能够在不断演进的计划中保留关键信息。例如，图 2 中的示例，原始计划并不知道顶级贡献者是谁，因此只能包含“识别顶级贡献者并关注他们”这一步骤。在执行点击“贡献者”链接操作后，规划器将识别出的贡献者信息纳入剩余计划。由于计划携带相关上下文，该方法还帮助我们解决了长远任务中的记忆挑战，无需显式记忆模块 [54, 60]。更多动态重新规划的详细示例见附录 A.2。

This approach aligns with our architectural philosophy where the PLANNER serves as the ”control room” for reasoning and decision-making, while the EXECUTOR focuses solely on translating plans into environment-specific actions.

该方法符合我们的架构理念，即规划器作为推理和决策的“控制室”，而执行器专注于将计划转化为环境特定的操作。

3.4. Chain of Thought Reasoning

3.4. 思维链推理 (Chain of Thought Reasoning)

Currently, the PLANNER and EXECUTOR generate plans and actions directly. However, recent advances in chain-of-thought (CoT) prompting and inference-time scaling [22, 51, 11] have shown that eliciting intermediate, step-by-step rationales can substantially improve performance. Thus, before having the PLANNER and EXECUTOR generate the plan and action respectively, we also have them generate a CoT reasoning trace in order to improve performance.

目前，规划器和执行器直接生成计划和操作。然而，近期思维链 (CoT) 提示和推理时扩展 [22, 51, 11] 的进展表明，诱导中间的逐步推理过程能显著提升性能。因此，在规划器和执行器分别生成计划和操作之前，我们让它们生成思维链推理轨迹，以提升表现。

4. Synthetic Data Generation

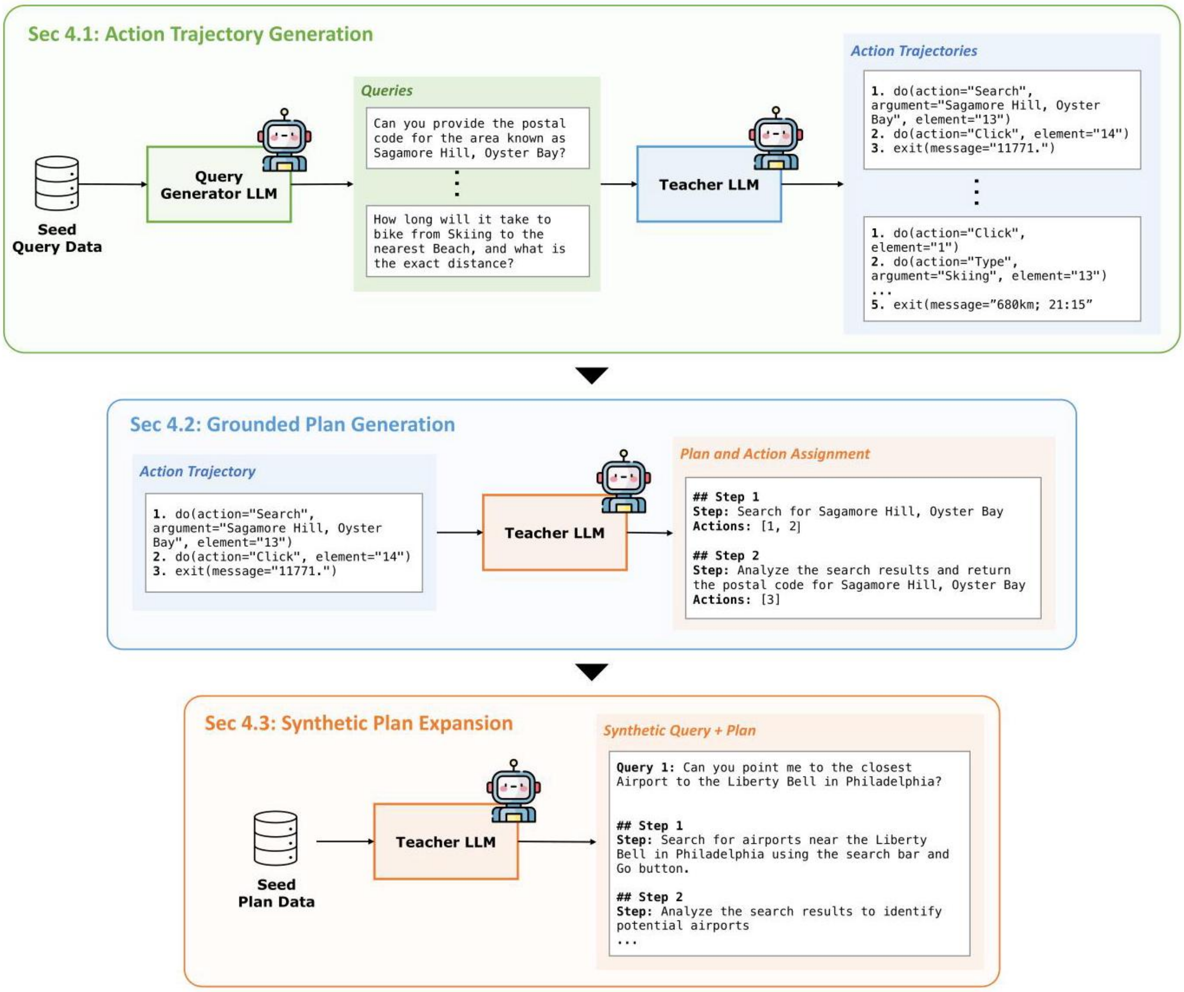
4. 合成数据生成

To motivate the need for creating synthetic data, we first evaluated the performance of existing off-the-shelf LLMs on WebArena-Lite which involves challenging user queries and reported the results in Table 1. We observe a baseline performance of 9.85%, which increases to 14.21% with PLAN-AND-ACT. While this is a noticeable improvement, the result is far from satisfactory.

为了说明创建合成数据的必要性，我们首先评估了现有现成大型语言模型 (LLMs) 在 WebArena-Lite 上的表现，该任务涉及具有挑战性的用户查询，结果见表 1。我们观察到基线表现为 9.85%，使用 PLAN-AND-ACT 后提升至 14.21%。虽然这是显著的进步，但结果仍远未令人满意。

There are several contributing factors to this low performance. Most notably, LLMs are not trained to perform this form of long horizon planning, especially for web tasks. This affects both the PLANNER and EXECUTOR. The PLAN-

导致这一低性能的因素有多方面。最显著的是，LLMs 并未被训练用于执行这种长远规划，尤其是针对网页任务。这影响了规划器 (PLANNER) 和执行器 (EXECUTOR)。计划-



```
Can you provide the postal code for the area known as Sagamore Hill, Oyster Bay?
⋮
How long will it take to bike from Skiing to the nearest Beach, and what is the exact distance?
```

```
1. do(action="Search", argument="Sagamore Hill, Oyster Bay", element="13")
2. do(action="Click", element="14")
3. exit(message="11771.")
⋮
1. do(action="Click", element="1")
2. do(action="Type", argument="Skiing", element="13")
...
5. exit(message="680km; 21:15")
```

```
1. do(action="Search", argument="Sagamore Hill, Oyster Bay", element="13")
2. do(action="Click", element="14")
3. exit(message="11771.")
```

```
## Step 1
Step: Search for Sagamore Hill, Oyster Bay
Actions: [1, 2]

## Step 2
Step: Analyze the search results and return the postal code for Sagamore Hill, Oyster Bay
Actions: [3]
```

```
Query 1: Can you point me to the closest Airport to the Liberty Bell in Philadelphia?

## Step 1
Step: Search for airports near the Liberty Bell in Philadelphia using the search bar and Go button.

## Step 2
Step: Analyze the search results to identify potential airports
...
```

Figure 3. Synthetic Data Generation Pipeline. In the Action Trajectory Generation stage (Section 4.1), user queries from the training data are given to a Teacher LLM, which outputs synthetic user instructions. From there, a demonstrator actor LLM attempts to execute the query on the webpage. After the trajectory is finished, an ORM LLM is used to filter for successful trajectories. In the Grounded Plan Generation stage (Section 4.2), a Teacher LLM takes the trajectory and creates a synthetic high-level plan and grounds each step with explicit actions in the trajectory. In the Synthetic Plan Expansion stage (Section 4.3), the plans from the training data are sampled and given to the Teacher LLM, which generates new synthetic plans.

图 3. 合成数据生成流程。在动作轨迹生成阶段 (第 4.1 节), 训练数据中的用户查询被输入教师 LLM, 输出合成用户指令。随后，示范者执行者 LLM 尝试在网页上执行查询。轨迹完成后，使用结果监督奖励模型 (ORM)LLM 筛选成功轨迹。在有根计划生成阶段 (第 4.2 节)，教师 LLM 接收轨迹，创建合成的高层计划，并用轨迹中的明确动作为每一步提供依据。在合成计划扩展阶段 (第 4.3 节)，从训练数据中采样计划，输入教师 LLM，生成新的合成计划。

NER cannot generate accurate plans if the LLM has not seen these websites in its pretraining and the trajectories that are needed to accomplish the query. The EXECUTOR has also most likely not been trained on getting an instruction along with the HTML of a page and output a web action. Overall, we cannot expect an off-the-shelf LLM to have this capability if it has not been trained on planning/executing tasks during pretraining such as in a specialized domain, such as web navigation.

如果 LLM 在预训练中未见过这些网站及完成查询所需的轨迹，命名实体识别 (NER) 无法生成准确计划。执行器也很可能未接受过结合页面 HTML 和指令输出网页动作的训练。总体而言，如果未在预训练中针对规划/执行任务 (如网页导航等专业领域) 进行训练，我们不能指望现成 LLM 具备此能力。

Importantly this issue cannot be solved with prompting alone. While performing prompt engineering and including in-context examples can help with simple tasks, the LLM struggles when given non-trivial queries as evident by the low baseline accuracy.

重要的是，这一问题仅靠提示工程无法解决。虽然设计提示和包含上下文示例能帮助完成简单任务，但面对复杂查询时，LLM 表现依然不佳，基线准确率低即是明证。

Given that prompting alone cannot solve this issue, we have to perform finetuning of the LLM to improve the performance of the PLANNER and EXECUTOR. However, in order to finetune the model, we need to ensure that we have sufficient amount of data for finetuning. In particular, the PLANNER requires data that has a user query and its corresponding plan breakdown. The EXECUTOR needs data that includes HTML input for each of the plan steps along with the desired web action as its output. However, such data is not available. Manually collecting this data is an option but is both costly and time-consuming as the human annotator needs to write down a plan of action followed by execution corresponding to each of the steps.

鉴于仅靠提示无法解决问题，我们必须对 LLM 进行微调以提升规划器和执行器的性能。然而，为了微调模型，需要确保有足够的微调数据。特别是，规划器需要包含用户查询及其对应计划分解的数据；执行器则需要包含每个计划步骤的 HTML 输入及期望网页动作输出的数据。但此类数据并不可得。人工收集虽可行，但成本高且耗时，因为人工标注者需为每一步编写行动计划并执行。

Here we propose an alternative approach that allows a scalable method to collect and generate high-quality synthetic training data. Our method leverages an LLM-based annotation pipeline that processes existing action trajectories to generate the corresponding structured plans, which is depicted in Figure 3.

在此，我们提出一种可扩展的方法，用于收集和生成高质量合成训练数据。该方法利用基于 LLM 的标注流水线，处理现有动作轨迹以生成对应的结构化计划，如图 3 所示。

4.1. Action Trajectory Generation

4.1. 动作轨迹生成

The simplest way to collect more data for the EXECUTOR is to collect more trajectories from the environment, a technique that has been used in previous works [35, 28].

为执行器收集更多数据的最简单方法是从环境中收集更多轨迹，这一技术已被先前工作 [35, 28] 采用。

To achieve this, we use an Alpaca-style [45] data generation pipeline. Motivated by Qi et al. [35], we randomly sample user queries from the training data and use them as seed prompts for an LLM to generate new, similar queries. For the web-navigation task, these queries are filtered initially by an LLM to filter out impossible trajectories for the web-agent. These newly generated instructions are then given to a demonstrator agent which tries to complete the task in the environment, which we collect as a synthetic trajectory. Finally, we then score this trajectory by using an outcome-supervised reward model (ORM) to filter for successful and unsuccessful trajectories. This initial process is depicted in Figure 3 in the first row for the web-navigation task and the prompts for the query generation are adapted from [35].

为此，我们采用类似 Alpaca[45] 的数据生成流水线。受 Qi 等人 [35] 启发，我们从训练数据中随机采样用户查询，作为 LLM 生成新相似查询的种子提示。针对网页导航任务，这些查询首先由 LLM 筛选，剔除网页代理无法完成的轨迹。随后，将新生成的指令交给示范代理尝试在环境中完成任务，收集为合成轨迹。最后，使用结果监督奖励模型 (ORM) 对轨迹进行评分，筛选成功与失败轨迹。该初始过程如图 3 第一行所示，查询生成提示参考 [35]。

4.2. Grounded Plan Generation

4.2. 有根计划生成

While we can use synthetically created user queries and then collect the resulting trajectories to train the EXECUTOR, this approach presents challenges when used to create synthetic data for the PLANNER.

虽然我们可以使用合成用户查询并收集相应轨迹来训练执行器, 但该方法在为规划器生成合成数据时存在挑战。

A naive approach would be to provide a teacher LLM with a user query and prompt it to generate a step-by-step plan to accomplish the task. However, this runs into a fundamental limitation: the teacher LLM lacks access to the actual website or environment and has not been pretrained on such tasks. Attempting this method results in generated synthetic plans that are often misaligned with how the tasks need to be performed on the web.

一种简单做法是向教师 LLM 提供用户查询，提示其生成逐步完成任务的计划。但这存在根本限制: 教师 LLM 无法访问实际网站或环境，且未在此类任务上进行预训练。尝试此法往往导致生成的合成计划与网页任务的实际执行方式不符。

To address this, we leverage the in-context learning capabilities of LLMs, which allows them to generalize on tasks outside their pretraining distribution. Specifically, we take advantage of this capability and provide the teacher LLM the trajectories that we created in Sec. 4.1 and prompt it to ”reverse-engineer” structured plans from these trajectories. Given the trajectory, we prompt the LLM to analyze the sequence of actions and to synthesize a coherent plan that will be used to guide the EXECUTOR downstream. To make sure that the plan is grounded to the actual environment, we further prompt the model to also include which low-level ac-

为了解决这一问题，我们利用大型语言模型 (LLMs) 的上下文学习能力，使其能够在预训练分布之外的任务上进行泛化。具体而言，我们利用这一能力，向教师 LLM 提供我们在第 4.1 节中创建的轨迹，并提示其从这些轨迹中“逆向工程”出结构化计划。给定轨迹后，我们提示 LLM 分析动作序列，并综合出一个连贯的计划，用以指导后续的执行器 (EXECUTOR)。为了确保计划与实际环境相契合，我们进一步提示模型包含哪些低级操作—

tions in the trajectory would be assigned to which high-level actions in the plan, to ensure that the plan matches actual execution of the trajectory. This ensures that the generated plans align with the real execution environment, making the both accurate and executable.

轨迹中的动作将被分配到计划中的哪些高级动作，以确保计划与轨迹的实际执行相匹配。这保证了生成的计划与真实执行环境一致，使其既准确又可执行。

This is depicted in the second row of Figure 3, where the action trajectories of the webagent is transformed into a set of high-level actions that we want the PLANNER to output. This approach is similar to [28], although our method generates high-level plans while their technique generates synthetic user queries from the trajectories.

如图 3 第二行所示，webagent 的动作轨迹被转换为一组我们希望规划器 (PLANNER) 输出的高级动作。该方法与文献 [28] 类似，尽管我们的方法生成的是高级计划，而他们的技术则是从轨迹生成合成用户查询。

4.2.1. SYNTHETIC DATA GENERATION FOR DYNAMIC REPLANNING

4.2.1. 用于动态重新规划的合成数据生成

It is important to also create synthetic data that captures dynamic replanning. This is important because a lot of user queries require planning based on dynamic observations that are only known during the plan execution. Examples queries that require such planning are: ”Analyze the search results and select the most relevant item” or ”Find the most recent pending order”.

同样重要的是生成能够反映动态重新规划的合成数据。这一点很关键，因为许多用户查询需要基于仅在计划执行过程中才能获得的动态观察进行规划。需要此类规划的示例查询包括:“分析搜索结果并选择最相关的项目”或“查找最新的待处理订单”。

We can use a similar algorithm to generate synthetic replanning data. The main difference is that for replanning data generation, we need to supply the teacher LLM with original plan data along with the trajectory that the webagent has taken to reach the point that requires replanning. You can find detailed prompts in Appendix A.9.

我们可以使用类似的算法来生成合成的重新规划数据。主要区别在于，对于重新规划数据的生成，我们需要向教师大语言模型 (LLM) 提供原始计划数据以及网络代理 (webagent) 为达到需要重新规划的点所采取的轨迹。详细的提示可见附录 A.9。

4.2.2. SYNTHETIC DATA GENERATION FOR CHAIN-OF-THOUGHT-REASONING

4.2.2. 用于链式思维推理的合成数据生成

Similarly, we also need to generate synthetic data to elicit CoT reasoning for both the PLANNER and EXECUTOR, since not all models have been trained to generate reasoning traces. We use an algorithm similar to Section

4.2 to generate reasoning traces for both plan and action generation. For plan reasoning generation, we have the teacher LLM generate reasoning before outputting the plan while for action reasoning generation, we provide the original plan data and the trajectory that the webagent has taken, along with the expected correct action and prompt the teacher LLM to generate a reasoning trace for that action.

同样地，我们还需要生成合成数据以引出 PLANNER(规划器) 和 EXECUTOR(执行器) 的链式思维 (CoT) 推理，因为并非所有模型都经过训练以生成推理轨迹。我们使用与第 4.2 节类似的算法来生成计划和动作生成的推理轨迹。对于计划推理生成，我们让教师大语言模型 (LLM) 在输出计划之前生成推理；而对于动作推理生成，我们提供原始计划数据和 webagent 所采取的轨迹，以及预期的正确动作，促使教师 LLM 为该动作生成推理轨迹。

4.3. Synthetic Plan Expansion

4.3. 合成计划扩展

The previous approach requires a simulator environment to collect actions and then create a synthetic plan by reverse-engineering the actions. Collecting successful trajectories that passes the ORM in Sec. 4.1 can be time consuming since the teacher model may generate a lot of unacceptable trajectories. This will affect the amount of data that we can generate both for the EXECUTOR as well as the PLANNER. This issue is noticeably worse for the PLANNER since each

之前的方法需要一个模拟环境来收集动作,然后通过逆向工程这些动作来创建合成计划。收集通过第 4.1 节中 ORM 的成功轨迹可能非常耗时，因为教师模型可能会生成大量不可接受的轨迹。这将影响我们为执行器 (EXECUTOR) 和规划器 (PLANNER) 生成的数据量。这个问题对规划器 (PLANNER) 来说尤其严重，因为每个

successful trajectory that passes the ORM model entails on average 8 different steps which provide 8 training data points for the EXECUTOR, but only 1 plan. However, we can effectively address this data imbalance, by expanding the synthetic plans.

成功通过 ORM 模型的轨迹平均包含 8 个不同步骤，这为 EXECUTOR 提供了 8 个训练数据点，但只有 1 个计划。然而，我们可以通过扩展合成计划有效地解决这一数据不平衡问题。

Specifically, we expand the PLANNER dataset by generating similar query-plan pairs that resemble the existing data, similar to the Alpaca style query generation in Section 4.1.

具体来说，我们通过生成与现有数据相似的查询-计划对来扩展 PLANNER 数据集，类似于第 4.1 节中 Alpaca 风格的查询生成。

Similar to Section 4.1, we initially randomly sample query-plan pairs from the synthetic PLANNER training data. These examples serve as implicit constraints, guiding the language model to generate structurally consistent and semantically valid query-plan pairs while maintaining diversity.

类似于第 4.1 节，我们最初从合成的 PLANNER 训练数据中随机抽取查询-计划对。这些示例作为隐式约束，引导语言模型生成结构一致且语义有效的查询-计划对，同时保持多样性。

Using this pipeline with GPT-4o, we expanded the synthetic plan data to 10,000 additional user query-plan pairs. This approach demonstrated significant advantages in both efficiency and scalability, reducing data generation time to under an hour while simultaneously addressing the overfit-ting problem through increased data diversity. The resulting synthetic dataset exhibits a broad spectrum of use cases, contributing to improved model generalization (Section 5.2). This process is depicted in the third row of Figure 3 and the prompts for this are in Appendix A.5.

使用该流程与 GPT-4o，我们将合成计划数据扩展到了额外的 10,000 个用户查询-计划对。该方法在效率和可扩展性方面表现出显著优势，将数据生成时间缩短至不到一小时，同时通过增加数据多样性解决了过拟合问题。生成的合成数据集涵盖了广泛的使用场景，有助于提升模型的泛化能力 (第 5.2 节)。该过程如图 3 第三行所示，相关提示见附录 A.5。

Targeted Plan Augmentation: While this large-scale data generation enhanced diversity and reduced underfitting for the PLANNER, it is not adaptive, and doesn't take into account what kinds of tasks are more difficult for the model in training. A key advantage of our approach is that we have explicit control over the data generation by allowing us to analyze the model failures and selectively refine the dataset.

目标计划增强: 虽然这种大规模数据生成提高了多样性并减少了规划器 (PLANNER) 的欠拟合，但它缺乏自适应性，未考虑模型训练中哪些任务更具挑战性。我们方法的一个关键优势在于，通过允许分析模型失败并有选择地优化数据集，实现了对数据生成的明确控制。

Motivated by previous works [24, 35, 55] around adaptive curriculum-learning, we ran our model through a held-out validation set which revealed several failure patterns in model performance. From there, our goal was to identify training data examples that seemed relevant to the failure patterns, instances where, if the model had seen more similar examples during training, it might improve performance on these tasks. To this end, we used an LLM to classify training data points that could be relevant to the failure nodes on each website and used them as seed data to generate 5,000 more synthetic plans. The prompts used to do this are in Appendix A. 7 for classification and Appendix A. 8 for generation. As we can see in Table 1, this targeted plan augmentation was able to significantly improve performance.

受之前关于自适应课程学习 [24, 35, 55] 工作的启发，我们在一个保留的验证集上运行了我们的模型，发现了模型性能中的若干失败模式。基于此，我们的目标是识别与这些失败模式相关的训练数据样本，即如果模型在训练时见过更多类似样本，可能会提升在这些任务上的表现。为此，我们使用大型语言模型 (LLM) 对可能与每个网站的失败节点相关的训练数据点进行分类，并将其作为种子数据生成了 5000 个合成计划。用于分类的提示见附录 A.7，生成的提示见附录 A.8。如表 1 所示，这种有针对性的计划增强显著提升了性能。

5. Results

5. 结果

5.1. Experimental Setup

5.1. 实验设置

- Environment: We run ablations on PLAN-AND-ACT using WebArena-Lite [20], a benchmark containing 165 test cases across diverse websites including OpenStreetMap,

- 环境: 我们在 PLAN-AND-ACT 框架上使用 WebArena-Lite [20] 进行消融实验，该基准包含 165 个跨多个网站的测试用例，包括 OpenStreetMap、

Reddit, GitLab, a content management system (CMS), and OneStopShop (OSS). WebArena-Lite uses a binary success metric (1 for complete task success, 0 for failure) and provides training data while being more computationally efficient than the full WebArena [61] benchmark. We also evaluate PLAN-AND-ACT on the full WebArena dataset as well as the WebVoyager [14] dataset, which is a dynamic, realworld web dataset.

Reddit、GitLab、内容管理系统 (CMS) 和 OneStopShop(OSS)。WebArena-Lite 采用二元成功指标 (任务完全成功为 1，失败为 0)，提供训练数据，且计算效率高于完整的 WebArena [61] 基准。我们还在完整的 WebArena 数据集及动态真实网页数据集 WebVoyager [14] 上评估 PLAN-AND-ACT。

- Models: For our primary PLAN-AND-ACT framework, we utilize LLaMA-3.3-70B-Instruct model by fine-tuning separate instances for both the PLANNER and EXECUTOR components. For our dynamic replanning experiments, we use a LLaMA-3.3-70B-Instruct model fine-tuned using LoRA [16] (due to computational constraints). Each component is trained on our synthesized datasets as described in previous sections. We use GPT-40 as the User Query Generator (Section 4.1), Plan Generator (Section 4.2) and Synthetic Plan Generator (Section 4.3). We use WebRL-Llama-3.1-70B [35] as the actor model and ORM-Llama-3.1-8B [35] as the filter model for filtering for successful trajectories. For generating CoT traces Section 3.4, we used DeepSeek-R1-Distill-Llama-70B [11] as the teacher model. We also applied PLAN-AND-ACT using LLaMA-3.1-8B-Instruct and QWQ-32B [46] for the WebArena and WebVoyager experiments.

- 模型: 在我们的主要 PLAN-AND-ACT 框架中，使用 LLaMA-3.3-70B-Instruct 模型，分别微调 PLANNER 和 EXECUTOR 组件的独立实例。动态重新规划实验中，因计算限制，使用基于 LoRA [16] 微调的 LLaMA-3.3-70B-Instruct 模型。各组件均在前述合成数据集上训练。用户查询生成器 (第 4.1 节)、计划生成器 (第 4.2 节) 及合成计划生成器 (第 4.3 节) 均采用 GPT-40。动作模型使用 WebRL-Llama-3.1-70B [35]，成功轨迹过滤模型使用 ORM-Llama-3.1-8B [35]。生成链式思维 (CoT) 轨迹 (第 3.4 节) 时，采用 DeepSeek-R1-Distill-Llama-70B [11] 作为教师模型。我们还在 WebArena 和 WebVoyager 实验中应用了 LLaMA-3.1-8B-Instruct 和 QWQ-32B [46]。

- Baselines: We compare PLAN-AND-ACT against several strong baselines to evaluate its effectiveness. These include zero-shot LLaMA-3.3-70B-Instruct without any fine-tuning, LLaMA-3.3-70B-Instruct fine-tuned specifically on the WebArena-Lite training set (ReAct-style prompting), and the WebRL-Llama-3.1-70B model, which is the current SOTA model on WebArena-lite. On Webarena-lite, we also compared against GPT-4-Turbo, GPT-4o, AWM [50], and WebPilot [60]. For the full WebArena dataset, we evaluated against NNetNav [28], AutoWebGLM [23], WebPilot, and AgentOccam [54]. For the WebVoyager dataset, we evaluated against NNet-Nav, OpenWebVoyager [15], Wilbur[26], and Agent-E [1]. These models were evaluated with a success rate metric, which requires complete task completion for a positive score.

- 基线: 为评估 PLAN-AND-ACT 的有效性，我们与多个强基线进行了比较，包括未微调的零-shot LLaMA-3.3-70B-Instruct、专门在 WebArena-Lite 训练集上微调的 LLaMA-3.3-70B-Instruct(ReAct 风格提示) 以及当前 WebArena-Lite 的 SOTA 模型 WebRL-Llama-3.1-70B。在 WebArena-Lite 上，还比较了 GPT-4-Turbo、GPT-4o、AWM [50] 和 WebPilot [60]。在完整 WebArena 数据集上，评估了 NNetNav [28]、AutoWebGLM [23]、WebPilot 和 AgentOccam [54]。在 WebVoyager 数据集上，评估了 NNet-Nav、OpenWebVoyager [15]、Wilbur[26] 和 Agent-E [1]。这些模型均以成功率指标评估，要求任务完全完成才计为成功。

- Hyperparameters: The hyperparameters we used to train our PLANNER and EXECUTOR modules are found in Table 5. For the data generation in Section 4.1 and Section 4.3, we use 5 seed data points to generate 10 new synthetic data points.

- 超参数: 我们训练 PLANNER 和 EXECUTOR 模块所用的超参数见表 5。第 4.1 节和第 4.3 节的数据生成中，使用 5 个种子数据点生成 10 个新的合成数据点。

5.2. Static PLANNER Results

5.2. 静态 PLANNER 结果

Table 1 shows the results of our experiments.

表 1 展示了我们的实验结果。

The columns represent a different versions of the EXECUTOR (LLaMA-3.3-70B). The first column is a base EXECU-

各列代表不同版本的 EXECUTOR(LLaMA-3.3-70B)。第一列为基础 EXECU-

Table 1. Task success rate (SR) of PLAN-AND-ACT on WebArena-Lite, a human-verified subset of WebArena. The rows represent incremental improvements to the PLANNER, while the columns show results for different EXECUTOR. For the executor, the first column is a base prompted EXECUTOR, the second column is for an EXECUTOR finetuned on WebArena-lite data, and the third column shows results when finetuned on both WebArena-lite training data and the 923 synthetically generated data from Section 4.1. The first row shows the results without a PLANNER. For the results in this row, the Executors were trained ReAct style, with no plans. The second to sixth row shows reported results from WebRL [35] including the current SOTA on WebArena-lite. The seventh row shows the result when using a base zero-shot PLANNER. The eigth row adds finetuning with the WebArena-lite data and the ninth row adds finetuning with the additional data generated in Section 4.1. The tenth and eleventh row are after finetuning with the 10,000 synthetic plans generated in Section 4.3 and with the additional 5,000 synthetic plans generated by Targeted Augmentation. The 12th row shows the results after we introduce dynamic replanning into the architecture. The last row shows the results after adding CoT reasoning. Scores are averaged across all websites in the WebArena environment.

PLANNER Design	EXECUTOR Design		
	Base	+ Finetuning	+ Synthetic Traj.
No Planner	9.85	36.36	36.97
GPT-4-Turbo	-	-	17.6*
GPT-4o	-	-	13.9*
AWM + GPT-4-0613 [50]	-	-	35.5*
WebPilot + GPT-4o [60]	-	-	37.2*
WebRL-3.1-70B [35]	-	-	49.1*
Base	14.21	17.16	23.63
+ Finetuning	22.42	16.36	20.60
+ Synthetic Trajectories (Section 4.1)	24.24	27.28	30.30
+ Plan Expansion (Section 4.3)	27.10	38.18	39.40
+ Targeted Augmentation (Section 4.3)	29.63	42.42	43.63
+ Dynamic Replanning (Section 3.3)	44.24	48.48	53.94
+ CoT (PLAN-AND-ACT) (Section 3.4)	-	-	57.58

规划器设计	执行器设计		
	基础	+ 微调	+ 合成轨迹
无规划器	9.85	36.36	36.97
GPT-4-Turbo	-	-	17.6*
GPT-4o	-	-	13.9*
AWM + GPT-4-0613 [50]	-	-	35.5*
WebPilot + GPT-4o [60]	-	-	37.2*
WebRL-3.1-70B [35]	-	-	49.1*
基础	14.21	17.16	23.63
+ 微调	22.42	16.36	20.60
+ 合成轨迹 (第 4.1 节)	24.24	27.28	30.30
+ 计划扩展 (第 4.3 节)	27.10	38.18	39.40
+ 定向增强 (第 4.3 节)	29.63	42.42	43.63
+ 动态重新规划 (第 3.3 节)	44.24	48.48	53.94
+ 链式思维 (计划与执行)(第 3.4 节)	-	-	57.58

TOR, which is not finetuned. The second column has an EXECUTOR that was trained only on 1,113 WebArena-lite training data points, and the third column being an EXECUTOR trained on both the WebArena-lite training data as well as the 923 synthetically generated action trajectories from Section 4.1.

TOR，未经过微调。第二列的 EXECUTOR 仅在 1,113 个 WebArena-lite 训练数据点上训练，第三列的 EXECUTOR 则同时在 WebArena-lite 训练数据和第 4.1 节中合成生成的 923 条动作轨迹上训练。

No PLANNER. The first row shows the results for each of these Executors when trained with the baseline ReAct prompt [57] without a PLANNER. As we can see, doubling the amount of action trajectory data does not significantly improve performance, showing a 0.61% improvement from just training on the WebArena-lite data. We cited this as motivation in Section 4.1 to focus on improving the PLANNER through plan generation.

无 PLANNER。第一行展示了在没有 PLANNER 的情况下，使用基线 ReAct 提示 [57] 训练的各个 Executor 的结果。可以看到，动作轨迹数据量翻倍并未显著提升性能，仅比单独使用 WebArena-lite 数据训练提升了 0.61%。我们在第 4.1 节中引用此结果作为改进 PLANNER 计划生成的动机。

Base PLANNER. The seventh row shows the results of enhancing each EXECUTOR with a base PLANNER (LLaMA- 3.3-70B), which is not finetuned. As we can see, the performance improves for the base EXECUTOR, but fails to improve over the baseline for the trained Executors. What this can be attributed to is that since the PLANNER is not trained on data that is grounded to these specific websites, these plans are suboptimal and can confuse the EXECUTOR.

基础 PLANNER。第七行展示了用基础 PLANNER(LLaMA-3.3-70B，未微调) 增强各 EXECUTOR 后的结果。可以看到，基础 EXECUTOR 的性能有所提升，但训练过的 Executors 未能超越基线。这可能是因为 PLANNER 未在与这些特定网站相关的数据上训练，导致生成的计划次优，反而干扰了 EXECUTOR。

Finetuned PLANNER. The eighth row shows the results

微调 PLANNER。第八行展示了结果

of using a PLANNER that has been finetuned only on the 1,113 plans in the WebArena-lite training data. As we can see, naively finetuning the PLANNER did not improve performance for the finetuned EXECUTOR. Here, we found that the PLANNER was overfitting to the training data and did not generalize to general plans for new, unseen tasks.

使用仅在 WebArena-lite 训练数据中 1,113 个计划上微调的 PLANNER。可以看到，简单微调 PLANNER 并未提升微调 EXECUTOR 的性能。我们发现 PLANNER 过拟合训练数据，未能泛化到新的、未见过任务的一般计划。

Finetuned PLANNER with data expansion. The ninth through eleventh row show the results when iteratively different synthetic data augmentation strategies. The ninth row shows the results after augmenting the PLANNER with extra synthetic trajectories from Section 4.1. The tenth row shows the results after using these grounded trajectories to generate 10,000 synthetic query-plan pairs from Section 4.3, and the eleventh row shows the performance after also adding in the 5,000 targeted synthetic query-plan pairs.

带数据扩展的微调 PLANNER。第九至第十一行展示了采用不同合成数据增强策略的迭代结果。第九行展示了用第 4.1 节额外合成轨迹增强 PLANNER 后的结果。第十行展示了利用这些有根轨迹生成 1 万条合成查询-计划对 (第 4.3 节) 的结果，第十一行则是在此基础上再加入 5,000 条针对性合成查询-计划对后的性能。

From these rows, we can see that a properly trained PLANNER consistently improves performance across all of the different Executors. Even with a base EXECUTOR, adding a PLANNER increases success rate from 9.85% to 29.63%. This validates our core hypothesis that explicit planning helps bridge the gap between high-level user intentions and low-level actions.

从这些行可以看出，经过适当训练的 PLANNER 在所有不同 EXECUTOR 上均持续提升性能。即使是基础 EXECUTOR，加入 PLANNER 后成功率也从 9.85% 提升至 29.63%。这验证了我们的核心假设: 显式规划有助于弥合用户高层意图与低层动作之间的差距。

The impact of generated data expansion is particularly no-

生成数据扩展的影响尤为显著。

table. Each expansion of the training dataset yields performance improvements, with the most substantial gains coming from the addition of the 10,000 directly generated plans, increasing success rates by approximately 10 percentage points. The failure-analysis-guided examples provided a final boost of 4-5 percentage points, highlighting the value of targeted data generation. Interestingly, the EXECUTOR's performance scales with training data size, but shows diminishing returns after the initial 1,113 examples, suggesting that the bottleneck may lie more in plan quality than action execution.

每次训练数据集的扩展都带来性能提升，其中最显著的提升来自新增的 1 万条直接生成的计划，成功率提升约 10 个百分点。基于失败分析指导生成的示例则带来了额外 4-5 个百分点的提升，凸显了针对性数据生成的价值。有趣的是，EXECUTOR 的性能随训练数据规模增长而提升，但在最初的 1,113 个样本后收益递减，表明瓶颈可能更多在于计划质量而非动作执行。

5.3. Dynamic Replanning Results

5.3. 动态重新规划结果

Despite these improvements, our detailed error analysis revealed fundamental limitations in the static PLANNER architecture. While the EXECUTOR performed well on concrete, deterministic tasks like navigating to specific pages, sorting tables, or posting comments on Reddit, it struggled with tasks requiring analysis of dynamic content. This aligns with our hypothesis in Section 3.3, that a static PLANNER would push some complex reasoning onto the EXECUTOR when it should focus solely on grounding abstract plans into specific actions. More explicit examples of this can be found in Appendix A.2.

尽管有这些改进，我们的详细错误分析揭示了静态 PLANNER 架构的根本局限。EXECUTOR 在具体确定性任务 (如导航至特定页面、排序表格或在 Reddit 发帖) 表现良好，但在需要分析动态内容的任务上表现欠佳。这与我们在第 3.3 节的假设一致，即静态 PLANNER 会将部分复杂推理任务转嫁给 EXECUTOR，而 PLANNER 应专注于将抽象计划具体化为动作。更多具体示例见附录 A.2。

5.3.1. REPLANNING RESULTS

5.3.1. 重新规划结果

To address the previous limitation we finetuned the PLANNER with additional replanning data (as discussed in Section 3.3). The results are shown in the 12th row of Table 1. As we can see, the addition of this capability significantly increases the performance of the model by 10.31% over the static PLANNER, achieving 53.94%

accuracy on WebArena-Lite. Notably, this result surpasses the previous SOTA WebRL-3.1-70B on the second row by 4.84%. You can see some explicit examples where the PLANNER with dynamic planning is able to refine and improve the plan in Section A.2, and a comprehensive breakdown of the performance by website can be found in Figure 4.

为解决上述限制，我们对 PLANNER 进行了额外的重新规划数据微调 (如第 3.3 节所述)。结果见表 1 第 12 行。可以看到，该能力的加入使模型性能较静态 PLANNER 提升了 10.31%，在 WebArena-Lite 上达到 53.94% 的准确率。值得注意的是，该结果比第二行的前 SOTA WebRL-3.1-70B 高出 4.84%。附录 A.2 中展示了 PLANNER 动态规划能够细化和改进计划的具体示例，网站性能的详细分解见图 4。

Importantly, even with a Base EXECUTOR (which has not been finetuned at all), we were able to significantly improve the performance of the model by 34.39%, achieving 44.24% accuracy, just by providing a high-quality and dynamic plan. This result highlights the importance of explicit planning and justifies our framework with separate PLANNER and EXECUTOR, demonstrating that a well-formed plan can substantially enhance performance even with an untrained EXECUTOR.

重要的是，即使使用未经过任何微调的基础执行器 (Base EXECUTOR)，我们也能够通过提供高质量且动态的计划，将模型性能显著提升 34.39%，达到 44.24% 的准确率。该结果凸显了明确规划的重要性，并验证了我们将规划器 (PLANNER) 与执行器 (EXECUTOR) 分离的框架，表明即使执行器未经过训练，一个良好构建的计划也能大幅提升性能。

5.3.2. CHAIN OF THOUGHT RESULTS

5.3.2. 思维链 (CHAIN OF THOUGHT) 结果

In the 13th row of Table 1, it shows the result of adding CoT reasoning to the PLANNER and EXECUTOR. As we can see, it improves performance by 4.36% and sets a new SOTA of

表 1 第 13 行展示了在规划器 (PLANNER) 和执行器 (EXECUTOR) 中加入思维链推理 (CoT) 的结果。正如我们所见，其性能提升了 4.36%，并创下了新的最先进水平 (SOTA)。

Base Model	CoT	Performance (%)
Llama-3.3-70B	✘	53.94
Llama-3.1-8B	✓	53.33
QWQ-32B	✓	54.88
Llama-3.3-70B	✓	57.58

基础模型	链式思维 (CoT)	性能 (%)
Llama-3.3-70B	✘	53.94
Llama-3.1-8B	✓	53.33
QWQ-32B	✓	54.88
Llama-3.3-70B	✓	57.58

Table 2. Comparison of PLAN-AND-ACT models with/without Chain-of-Thought. The first row shows the performance using a 70B model without the CoT data while the other rows show the performance using the CoT data.

表 2. 带有/不带有链式思维 (Chain-of-Thought, CoT) 的 PLAN-AND-ACT 模型比较。第一行展示了使用 70B 模型且不含 CoT 数据的性能，其他行则展示了使用 CoT 数据的性能。

57.58% on WebArena-lite.

WebArena-lite 上的准确率为 57.58%。

In order to quantify the improvement from using CoT reasoning, we also finetuned a Llama-3.1-8B-instruct model and a QWQ-32B on the exact same data as the model in the 13th row in Table 2. As we can see, the 8B model on the second row performs on par with the non-CoT 70B model on the third row, which shows just how important CoT is.

为了量化使用 CoT 推理带来的提升，我们还在与表 2 第 13 行模型相同的数据上微调了 Llama-3.1-8B-instruct 模型和 QWQ-32B 模型。正如我们所见，第二行的 8B 模型表现与第三行的不含 CoT 的 70B 模型相当，这充分说明了 CoT 的重要性。

5.4. WebArena Results

5.4. WebArena 结果

Method	Base Model	Acc. (%)
NNetNav [28]	Llama-3.1-8b	16.3
AutoWebGLM [23]	ChatGLM3-6B	18.2
WebPilot [60]	GPT-4o	37.2
AgentOccam [54]	GPT-4-Turbo	43.1
AgentOccam-Judge [54]	GPT-4-Turbo	45.7
PLAN-AND-ACT	Llama-70B	45.7
PLAN-AND-ACT	QWQ-32B	48.15

方法	基础模型	准确率 (%)
NNetNav [28]	Llama-3.1-8b	16.3
AutoWebGLM [23]	ChatGLM3-6B	18.2
WebPilot [60]	GPT-4o	37.2
AgentOccam [54]	GPT-4-Turbo	43.1
AgentOccam-Judge [54]	GPT-4-Turbo	45.7
计划与执行 (PLAN-AND-ACT)	Llama-70B	45.7
计划与执行 (PLAN-AND-ACT)	QWQ-32B	48.15

Table 3. Comparison of methods on the WebArena benchmark. As you can see, PLAN-AND-ACT performs on-par with other prior work.

表 3. WebArena 基准测试方法比较。如您所见，PLAN-AND-ACT 的表现与其他先前工作相当。

We also evaluated our approach on the full WebArena dataset in Table 3. As we can see, PLAN-AND-ACT performs better or on-par with most other prior work.

我们还在表 3 中对完整的 WebArena 数据集评估了我们的方法。正如我们所见，PLAN-AND-ACT 的表现优于或与大多数其他先前工作持平。

5.5. WebVoyager Results

5.5. WebVoyager 结果

Furthermore, we evalauted PLAN-AND-ACT on the Web-Voyager benchmark by finetuning a llama-3.1-8B model as well as a QWQ-32B model, which you can see in Table 4. Our goal was to evaluate our approach on real-world web-tasks, as opposed to the simulator based tasks in WebArena.

此外，我们通过微调 llama-3.1-8B 模型和 QWQ-32B 模型，在 Web-Voyager 基准上评估了 PLAN-AND-ACT，详见表 4。我们的目标是评估该方法在真实网页任务上的表现，而非 WebArena 中的模拟器任务。

Since WebVoyager does not have any training data, we used the text-only WebVoyager model with GPT-4o to collect 1500 synthetic trajectories (Section 4.1) and used QWQ- 32B to annotate the plans (Section 4.2), CoT reasoning (Section 3.4) and to generate 10k synthetic plans (Section 4.3).

由于 WebVoyager 没有任何训练数据，我们使用仅文本的 WebVoyager 模型结合 GPT-4o 收集了 1500 条合成轨迹 (第 4.1 节)，并使用 QWQ-32B 对计划 (第 4.2 节)、链式推理 (CoT，见第 3.4 节) 进行注释，以及生成了 1 万条合成计划 (第 4.3 节)。

Our 8B model out performs all previous open source models and our 32 B model out performs all prior work and sets a new SOTA for text-only WebVoyager.

我们的 8B 模型优于所有先前的开源模型，而我们的 32 B 模型超越了所有先前工作，创造了仅文本 WebVoyager 的新最高水平 (SOTA)。

Technique	Base Model	Acc. (%)
NNetNav [28]	Llama-3.1-8b	34.2
OpenWebVoyager [15]	Idefics2-8b-inst.	27.4
WebVoyager [14] (text)	GPT-4-Turbo	44.3
Wilbur [26]	GPT-4-Turbo	52.6
WebVoyager [14]	GPT-4-Turbo	57.1
PLAN-AND-ACT	Llama-3.1-8b	58.08
Agent-E [1]	GPT-4-Turbo	73.1
PLAN-AND-ACT	QWQ-32B	81.36

技术	基础模型	准确率 (%)
NNetNav [28]	Llama-3.1-8b	34.2
OpenWebVoyager [15]	Idefics2-8b-inst.	27.4
WebVoyager [14](文本)	GPT-4-Turbo	44.3
Wilbur [26]	GPT-4-Turbo	52.6
WebVoyager [14]	GPT-4-Turbo	57.1
计划与执行	Llama-3.1-8b	58.08
Agent-E [1]	GPT-4-Turbo	73.1
计划与执行	QWQ-32B	81.36

Table 4. Comparison of techniques on the WebVoyager benchmark. PLAN-AND-ACT outperforms all open-source prior work and sets a new text-only SOTA on WebVoyager.

表 4. WebVoyager 基准测试中技术的比较。PLAN-AND-ACT 优于所有开源的先前工作，并在 WebVoyager 上创下了新的纯文本最高水平 (SOTA)。

6. Conclusion

6. 结论

It has been shown that separating high-level reasoning (PLANNER) from low-level execution (EXECUTOR), improves alignment between user queries and executable actions, enhancing task consistency and adaptability to dynamic environments. However, a major challenge with this approach is that out-of-the-box LLMs are not efficient at generating accurate plans, importantly for environments that are out of their pretraining distribution such as web tasks. In this work, we introduced PLAN-AND-ACT, a novel framework that enhances the ability of LLM agents to tackle complex, long-horizon tasks through a scalable synthetic data generation.

研究表明，将高层次推理 (PLANNER，规划器) 与低层次执行 (EXECUTOR，执行器) 分离，可以提升用户查询与可执行动作之间的对齐度，增强任务的一致性和对动态环境的适应性。然而，这种方法的主要挑战在于，开箱即用的大型语言模型 (LLM) 在生成准确计划方面效率不高，尤其是在其预训练分布之外的环境中，如网页任务。本文提出了 PLAN-AND-ACT，一种通过可扩展的合成数据生成，提升 LLM 代理处理复杂长时任务能力的新框架。

A key advantage of our method is its scalability and efficiency in data generation. While methods like WebRL require on-policy data collection through environment interaction, our synthetic data generation pipeline can rapidly produce high-quality training examples. For instance, we generated 15,000 synthetic training examples in under an hour using GPT-4o, whereas collecting a similar number of trajectories through environment interaction would take days or weeks, as each trajectory requires actual execution of web actions. This scalability allowed us to match state-of-the-art performance with a relatively simple training procedure of supervised fine-tuning on synthetic data.

我们方法的一个关键优势是其数据生成的可扩展性和高效性。与需要通过环境交互进行在线数据收集的 WebRL 等方法不同，我们的合成数据生成流程能够快速产出高质量训练样本。例如，使用 GPT-4o，我们在不到一小时内生成了 15,000 个合成训练样本，而通过环境交互收集相同数量的轨迹则需数天甚至数周，因为每条轨迹都需实际执行网页操作。这种可扩展性使我们能够通过相对简单的监督微调训练程序，在合成数据上达到最先进的性能。

Our results demonstrate that PLAN-AND-ACT significantly outperforms existing methods in web-based navigation tasks within the WebArena-Lite benchmark. Through a combination of synthetic data generation, plan expansion, and targeted failure case refinement, we showed that our framework consistently improves success rates. Notably, the introduction of dynamic replanning further enhanced model robustness by adapting execution strategies based on real-time observations.

我们的结果表明，PLAN-AND-ACT 在 WebArena-Lite 基准的网页导航任务中显著优于现有方法。通过合成数据生成、计划扩展和针对失败案例的精细调整，我们展示了该框架持续提升成功率的能力。值得注意的是，引入动态重新规划进一步增强了模型的鲁棒性，使执行策略能够基于实时观察进行调整。

By focusing solely on improving the planning component while keeping a standard WebArena-Lite style EXECUTOR, we demonstrate the potential of our approach even without sophisticated execution strategies. This modularity suggests that future work could further improve performance by enhancing the EXECUTOR with techniques like chain-of-thought reasoning while maintaining the benefits of our efficient planning framework.

通过仅专注于提升规划组件，同时保持标准的 WebArena-Lite 风格执行器，我们展示了即使没有复杂执行策略，该方法也具备潜力。这种模块化设计表明，未来工作可通过引入如链式思维推理等技术来增强执行器，同时保持我们高效规划框架的优势，从而进一步提升性能。

Beyond web navigation, our modular framework holds promise for broader applications in various digital environments, which are long-horizon decision making tasks. Future work will explore integrating multi-modal inputs, reinforcement learning-based plan optimization, and memory-enhanced reasoning to further improve agent capabilities.

除了网页导航，我们的模块化框架在各种数字环境中的长时决策任务中也展现出广泛应用前景。未来工作将探索整合多模态输入、基于强化学习的计划优化以及增强记忆推理，以进一步提升代理能力。

Limitations

局限性

One main drawback is that Action Trajectory Generation Section 4.1 does depend on having a baseline model that can successfully complete the web tasks. The synthetic data generation pipeline introduced in Section 4.3 is able to mitigate some of these concerns with a sufficient amount of training data. However, for datasets that do not have any training data, such as WebVoyager, the pipeline will depend on having a base model to collect trajectories.

一个主要缺点是，动作轨迹生成 (第 4.1 节) 依赖于能够成功完成网页任务的基线模型。第 4.3 节介绍的合成数据生成流程能够通过足够的训练数据缓解部分问题。然而，对于如 WebVoyager 这类没有任何训练数据的数据集，该流程仍需依赖基线模型来收集轨迹。

Furthermore, our current framework does dynamic replanning (3.3) after every action, which can be inefficient and slow down performance. Future work can address these concerns by having the EXECUTOR decide when it needs to replan, or by having the PLANNER delegate tasks to separate subagents.

此外，我们当前框架在每次动作后都进行动态重新规划 (3.3 节)，这可能效率低下并拖慢性能。未来工作可通过让执行器决定何时需要重新规划, 或让规划器将任务委派给不同子代理来解决这些问题。

Acknowledgments

致谢

We thank Chris Joseph John and Anshul Verma for their help with the running of some of the benchmarks. We acknowledge gracious support from Apple team, as well as Nvidia for providing GPU hardware. We also appreciate the support from Microsoft through their Accelerating Foundation Model Research. Furthermore, we appreciate support from Google Cloud, the Google TRC team, and specifically Jonathan Caton, Divvy Thakkar, and Prof. David Patterson. Prof. Keutzer's lab is sponsored by the Intel corporation, Intel One-API, Intel VLAB team, the Intel One-API center of excellence, as well as funding through BDD, BAIR, and Furiosa. We appreciate great feedback and support from Ellick Chan, Saurabh Tangri, Andres Rodriguez, and Kit-tur Ganesh. Sehoon Kim and Suhong Moon would like to acknowledge the support from the Korea Foundation for Advanced Studies (KFAS). Hiroki Furuta is supported by JSPS KAKENHI Grant Number JP22J21582. Our conclusions do not necessarily reflect the position or the policy of our sponsors, and no official endorsement should be inferred.

感谢 Chris Joseph John 和 Anshul Verma 在部分基准测试运行中的帮助。感谢 Apple 团队的慷慨支持，以及 Nvidia 提供的 GPU 硬件支持。感谢微软通过其加速基础模型研究项目的支持。此外，感谢 Google Cloud、Google TRC 团队，特别是 Jonathan Caton、Divvy Thakkar 和 David Patterson 教授。Keutzer 教授的实验室由 Intel 公司、Intel One-API、Intel VLAB 团队、Intel One-API 卓越中心以及 BDD、BAIR 和 Furiosa 的资助支持。感谢 Ellick Chan、Saurabh Tangri、Andres Rodriguez 和 Kit-tur Ganesh 的宝贵反馈和支持。Sehoon Kim 和 Suhong Moon 感谢韩国高等研究基金会 (KFAS) 的支持。Hiroki Furuta 获得 JSPS KAKENHI 资助，编号 JP22J21582。我们的结论不必然代表资助方的立场或政策，亦不应被视为官方认可。

References

参考文献

[1] Abuelsaad, T., Akkil, D., Dey, P., Jagmohan, A., Vem-paty, A., and Kokku, R. Agent-e: From autonomous web navigation to foundational design principles in agentic systems. arXiv preprint arXiv:2407.13032, 2024.

Abuelsaad, T., Akkil, D., Dey, P., Jagmohan, A., Vem-paty, A., 和 Kokku, R. Agent-e: 从自主网页导航到代理系统的基础设计原则。arXiv 预印本 arXiv:2407.13032, 2024。

[2] Bai, H., Zhou, Y., Cemri, M., Pan, J., Suhr, A., Levine, S., and Kumar, A. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. arXiv preprint arXiv:2406.11896, 2024.

Bai, H., Zhou, Y., Cemri, M., Pan, J., Suhr, A., Levine, S., 和 Kumar, A. Digirl: 通过自主强化学习训练野外设备控制代理。arXiv 预印本 arXiv:2406.11896, 2024。

[3] Chae, H., Kim, N., Ong, K. T.-i., Gwak, M., Song, G., Kim, J., Kim, S., Lee, D., and Yeo, J. Web agents with world models: Learning and leveraging environment dynamics in web navigation. arXiv preprint arXiv:2410.13232, 2024.

Chae, H., Kim, N., Ong, K. T.-i., Gwak, M., Song, G., Kim, J., Kim, S., Lee, D., 和 Yeo, J. 具备世界模型的网络代理: 在网页导航中学习和利用环境动态。arXiv 预印本 arXiv:2410.13232, 2024。

[4] Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2web: Towards a generalist agent for the web. Advances in Neural Information Processing Systems, 36, 2024.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., 和 Su, Y. Mind2web: 迈向通用网络智能体。《神经信息处理系统进展》(Advances in Neural Information Processing Systems), 第 36 卷, 2024 年。

[5] Erdogan, L. E., Lee, N., Jha, S., Kim, S., Tabrizi, R., Moon, S., Hooper, C., Anumanchipalli, G., Keutzer, K., and Gholami, A. Tinyagent: Function calling at the edge. arXiv preprint arXiv:2409.00608, 2024.

Erdogan, L. E., Lee, N., Jha, S., Kim, S., Tabrizi, R., Moon, S., Hooper, C., Anumanchipalli, G., Keutzer, K., 和 Gholami, A. Tinyagent: 边缘的函数调用。arXiv 预印本 arXiv:2409.00608，2024 年。

[6] Furuta, H., Lee, K.-H., Nachum, O., Matsuo, Y., Faust, A., Gu, S. S., and Gur, I. Multimodal web navigation with instruction-finetuned foundation models. arXiv preprint arXiv:2305.11854, 2023.

Furuta, H., Lee, K.-H., Nachum, O., Matsuo, Y., Faust, A., Gu, S. S., 和 Gur, I. 基于指令微调基础模型的多模态网页导航。arXiv 预印本 arXiv:2305.11854, 2023。

[7] Furuta, H., Matsuo, Y., Faust, A., and Gur, I. Exposing limitations of language model agents in sequential-task compositions on the web. arXiv preprint arXiv:2311.18751, 2023.

Furuta, H., Matsuo, Y., Faust, A., 和 Gur, I. 揭示语言模型代理在网络顺序任务组合中的局限性。arXiv 预印本 arXiv:2311.18751, 2023。

[8] Furuta, H., Lee, K.-H., Gu, S. S., Matsuo, Y., Faust, A., Zen, H., and Gur, I. Geometric-averaged preference optimization for soft preference labels. arXiv preprint arXiv:2409.06691, 2024.

Furuta, H., Lee, K.-H., Gu, S. S., Matsuo, Y., Faust, A., Zen, H., 和 Gur, I. 软偏好标签的几何平均偏好优化。arXiv 预印本 arXiv:2409.06691, 2024。

[9] Gu, Y., Zheng, B., Gou, B., Zhang, K., Chang, C., Srivastava, S., Xie, Y., Qi, P., Sun, H., and Su, Y. Is your llm secretly a world model of the internet? model-based planning for web agents. arXiv preprint arXiv:2411.06559, 2024.

Gu, Y., Zheng, B., Gou, B., Zhang, K., Chang, C., Srivastava, S., Xie, Y., Qi, P., Sun, H., 和 Su, Y. 你的大型语言模型 (LLM) 是否暗中成为了互联网的世界模型？基于模型的网络代理规划。arXiv 预印本 arXiv:2411.06559, 2024。

[10] Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., et al. Textbooks are all you need. arXiv preprint arXiv:2306.11644, 2023.

Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., 等. Textbooks are all you need. arXiv 预印本 arXiv:2306.11644, 2023.

[11] Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via rein-

郭东、杨东、张华、宋杰、张锐、徐然、朱强、马帅、王鹏、毕翔等。Deepseek-r1: 通过强化学习激励大型语言模型 (LLMs) 的推理能力—

forcement learning. arXiv preprint arXiv:2501.12948, 2025.

强化学习。arXiv 预印本 arXiv:2501.12948， 2025 年。

[12] Gur, I., Nachum, O., Miao, Y., Safdari, M., Huang, A., Chowdhery, A., Narang, S., Fiedel, N., and Faust, A. Understanding html with large language models. arXiv preprint arXiv:2210.03945, 2022.

Gur, I., Nachum, O., Miao, Y., Safdari, M., Huang, A., Chowdhery, A., Narang, S., Fiedel, N., 和 Faust, A. 利用大型语言模型理解 HTML。arXiv 预印本 arXiv:2210.03945, 2022。

[13] Gur, I., Furuta, H., Huang, A., Safdari, M., Matsuo, Y., Eck, D., and Faust, A. A real-world webagent with planning, long context understanding, and program synthesis. arXiv preprint arXiv:2307.12856, 2023.

Gur, I., Furuta, H., Huang, A., Safdari, M., Matsuo, Y., Eck, D., 和 Faust, A. 一个具备规划、长上下文理解和程序综合能力的真实世界网络代理。arXiv 预印本 arXiv:2307.12856, 2023。

[14] He, H., Yao, W., Ma, K., Yu, W., Dai, Y., Zhang, H., Lan, Z., and Yu, D. Webvoyager: Building an end-to-end web agent with large multimodal models. arXiv preprint arXiv:2401.13919, 2024.

He, H., Yao, W., Ma, K., Yu, W., Dai, Y., Zhang, H., Lan, Z., 和 Yu, D. Webvoyager: 利用大型多模态模型构建端到端网页代理。arXiv 预印本 arXiv:2401.13919, 2024。

[15] He, H., Yao, W., Ma, K., Yu, W., Zhang, H., Fang, T., Lan, Z., and Yu, D. Openwebvoyager: Building multimodal web agents via iterative real-world exploration, feedback and optimization. arXiv preprint arXiv:2410.19609, 2024.

He, H., Yao, W., Ma, K., Yu, W., Zhang, H., Fang, T., Lan, Z., 和 Yu, D. Openwebvoyager: 通过迭代的现实世界探索、反馈与优化构建多模态网络代理。arXiv 预印本 arXiv:2410.19609，2024 年。

[16] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2021.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., 和 Chen, W. Lora: 大型语言模型的低秩适配 (Low-rank adaptation)。arXiv 预印本 arXiv:2106.09685, 2021。

[17] Kannan, S. S., Venkatesh, V. L., and Min, B.-C. Smart-llm: Smart multi-agent robot task planning using large language models. In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 12140-12147. IEEE, 2024.

Kannan, S. S., Venkatesh, V. L., 和 Min, B.-C. Smart-llm: 利用大型语言模型进行智能多代理机器人任务规划。发表于 2024 年 IEEE/RSJ 国际智能机器人与系统会议 (IROS)，第 12140-12147 页。IEEE, 2024。

[18] Kim, G., Baldi, P., and McAleer, S. Language models can solve computer tasks. arXiv preprint arxiv:2303.17491, 2023.

Kim, G., Baldi, P., 和 McAleer, S. 语言模型能够解决计算机任务。arXiv 预印本 arXiv:2303.17491, 2023。

[19] Kim, S., Moon, S., Tabrizi, R., Lee, N., Mahoney, M. W., Keutzer, K., and Gholami, A. An llm compiler for parallel function calling. arXiv preprint arXiv:2312.04511, 2023.

Kim, S., Moon, S., Tabrizi, R., Lee, N., Mahoney, M. W., Keutzer, K., 和 Gholami, A. 用于并行函数调用的 LLM 编译器。arXiv 预印本 arXiv:2312.04511, 2023。

[20] Koh, J. Y., Lo, R., Jang, L., Duvvur, V., Lim, M. C., Huang, P.-Y., Neubig, G., Zhou, S., Salakhutdinov, R., and Fried, D. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. arXiv preprint arXiv:2401.13649, 2024.

Koh, J. Y., Lo, R., Jang, L., Duvvur, V., Lim, M. C., Huang, P.-Y., Neubig, G., Zhou, S., Salakhutdinov, R., 和 Fried, D. Visualwebarena: 在真实视觉网页任务上评估多模态代理。arXiv 预印本 arXiv:2401.13649, 2024。

[21] Koh, J. Y., McAleer, S., Fried, D., and Salakhutdinov, R. Tree search for language model agents. arXiv preprint arXiv:2407.01476, 2024.

Koh, J. Y., McAleer, S., Fried, D., 和 Salakhutdinov, R. 语言模型代理的树搜索。arXiv 预印本 arXiv:2407.01476, 2024。

[22] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwa-sawa, Y. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35:22199-22213, 2022.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., 和 Iwa-sawa, Y. 大型语言模型是零样本推理者。神经信息处理系统进展，第 35 卷:22199-22213, 2022。

[23] Lai, H., Liu, X., Iong, I. L., Yao, S., Chen, Y., Shen, P., Yu, H., Zhang, H., Zhang, X., Dong, Y., et al. Autowe-bglm: A large language model-based web navigating agent. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 5295-5306, 2024.

Lai, H., Liu, X., Iong, I. L., Yao, S., Chen, Y., Shen, P., Yu, H., Zhang, H., Zhang, X., Dong, Y., 等. Autowe-bglm: 基于大型语言模型的网页导航代理。发表于第 30 届 ACM 知识发现与数据挖掘会议 (SIGKDD)，第 5295-5306 页，2024。

[24] Lee, N., Wattanawong, T., Kim, S., Mangalam, K., Shen, S., Anumanchipalli, G., Mahoney, M. W., Keutzer, K., and Gholami, A. Llm2llm: Boosting llms with novel iterative data enhancement. arXiv preprint arXiv:2403.15042, 2024.

Lee, N., Wattanawong, T., Kim, S., Mangalam, K., Shen, S., Anumanchipalli, G., Mahoney, M. W., Keutzer, K., 和 Gholami, A. Llm2llm: 通过新颖的迭代数据增强提升大型语言模型。arXiv 预印本 arXiv:2403.15042, 2024。

[25] Liu, X., Zhang, T., Gu, Y., Iong, I. L., Xu, Y., Song, X., Zhang, S., Lai, H., Liu, X., Zhao, H., et al. Visualagentbench: Towards large multimodal models as visual foundation agents. arXiv preprint arXiv:2408.06327, 2024.

Liu, X., Zhang, T., Gu, Y., Iong, I. L., Xu, Y., Song, X., Zhang, S., Lai, H., Liu, X., Zhao, H., 等. Visualagentbench: 迈向作为视觉基础代理的大型多模态模型。arXiv 预印本 arXiv:2408.06327, 2024。

[26] Lutz, M., Bohra, A., Saroyan, M., Harutyunyan, A., and Campagna, G. Wilbur: Adaptive in-context learning for robust and accurate web agents. arXiv preprint arXiv:2404.05902, 2024.

Lutz, M., Bohra, A., Saroyan, M., Harutyunyan, A., 和 Campagna, G. Wilbur: 用于稳健且准确网页代理的自适应上下文学习。arXiv 预印本 arXiv:2404.05902, 2024。

[27] Moon, S., Jha, S., Erdogan, L. E., Kim, S., Lim, W., Keutzer, K., and Gholami, A. Efficient and scalable estimation of tool representations in vector space. arXiv preprint arXiv:2409.02141, 2024.

Moon, S., Jha, S., Erdogan, L. E., Kim, S., Lim, W., Keutzer, K., 和 Gholami, A. 工具表示在向量空间中的高效且可扩展估计。arXiv 预印本 arXiv:2409.02141, 2024。

[28] Murty, S., Bahdanau, D., and Manning, C. D. Nnetscape navigator: Complex demonstrations for web agents without a demonstrator. arXiv preprint arXiv:2410.02907, 2024.

Murty, S., Bahdanau, D., 和 Manning, C. D. Nnetscape navigator: 无需示范者的复杂网页代理演示。arXiv 预印本 arXiv:2410.02907, 2024。

[29] Nayak, S., Morrison Orozco, A., Have, M., Zhang, J., Thirumalai, V., Chen, D., Kapoor, A., Robinson, E., Gopalakrishnan, K., Harrison, J., et al. Long-horizon planning for multi-agent robots in partially observable environments. Advances in Neural Information Processing Systems, 37:67929-67967, 2024.

Nayak, S., Morrison Orozco, A., Have, M., Zhang, J., Thirumalai, V., Chen, D., Kapoor, A., Robinson, E., Gopalakrishnan, K., Harrison, J., 等. 在部分可观测环境中多代理机器人的长远规划。神经信息处理系统进展，第 37 卷:67929-67967, 2024。

[30] Ou, T., Xu, F. F., Madaan, A., Liu, J., Lo, R., Sridhar, A., Sengupta, S., Roth, D., Neubig, G., and Zhou, S. Synatra: Turning indirect knowledge into direct demonstrations for digital agents at scale. arXiv preprint arXiv:2409.15637, 2024.

Ou, T., Xu, F. F., Madaan, A., Liu, J., Lo, R., Sridhar, A., Sengupta, S., Roth, D., Neubig, G., 和 Zhou, S. Synatra: 将间接知识转化为大规模数字代理的直接示范。arXiv 预印本 arXiv:2409.15637, 2024。

[31] Pan, J., Zhang, Y., Tomlin, N., Zhou, Y., Levine, S., and Suhr, A. Autonomous evaluation and refinement of digital agents. arXiv preprint arXiv:2404.06474, 2024.

Pan, J., Zhang, Y., Tomlin, N., Zhou, Y., Levine, S., 和 Suhr, A. 数字代理的自主评估与改进。arXiv 预印本 arXiv:2404.06474, 2024。

[32] Patel, A., Hofmarcher, M., Leoveanu-Condrei, C., Dinu, M.-C., Callison-Burch, C., and Hochreiter, S. Large language models can self-improve at web agent tasks. arXiv preprint arXiv:2405.20309, 2024.

Patel, A., Hofmarcher, M., Leoveanu-Condrei, C., Dinu, M.-C., Callison-Burch, C., 和 Hochreiter, S. 大型语言模型 (Large Language Models) 能够在网络代理任务中自我提升。arXiv 预印本 arXiv:2405.20309, 2024。

[33] Pawlowski, P., Zawistowski, K., Lapacz, W., Skorupa, M., Wiacek, A., Postansque, S., and Hoscilowicz, J. Tinyclick: Single-turn agent for empowering gui automation. arXiv preprint arXiv:2410.11871, 2024.

Pawlowski, P., Zawistowski, K., Lapacz, W., Skorupa, M., Wiacek, A., Postansque, S., 和 Hoscilowicz, J. Tinyclick: 用于增强图形用户界面自动化的单轮代理。arXiv 预印本 arXiv:2410.11871, 2024。

[34] Prasad, A., Koller, A., Hartmann, M., Clark, P., Sabhar-wal, A., Bansal, M., and Khot, T. Adapt: As-needed decomposition and planning with language models. arXiv preprint arXiv:2311.05772, 2023.

Prasad, A., Koller, A., Hartmann, M., Clark, P., Sabharwal, A., Bansal, M., 和 Khot, T. Adapt: 基于需求的分解与规划结合语言模型。arXiv 预印本 arXiv:2311.05772, 2023。

[35] Qi, Z., Liu, X., Iong, I. L., Lai, H., Sun, X., Yang, X., Sun, J., Yang, Y., Yao, S., Zhang, T., et al. We-brl: Training llm web agents via self-evolving online curriculum reinforcement learning. arXiv preprint arXiv:2411.02337, 2024.

Qi, Z., Liu, X., Iong, I. L., Lai, H., Sun, X., Yang, X., Sun, J., Yang, Y., Yao, S., Zhang, T., 等。We-brl: 通过自我进化的在线课程强化学习训练大型语言模型网络代理。arXiv 预印本 arXiv:2411.02337, 2024。

[36] Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36, 2024.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., 和 Finn, C. 直接偏好优化: 你的语言模型实际上是一个奖励模型。神经信息处理系统进展 (NeurIPS)，第 36 卷，2024。

[37] Rawles, C., Li, A., Rodriguez, D., Riva, O., and Lilli-crap, T. Androidinthewild: A large-scale dataset for android device control. Advances in Neural Information Processing Systems, 36, 2024.

Rawles, C., Li, A., Rodriguez, D., Riva, O., 和 Lillicrap, T. Androidinthewild: 用于安卓设备控制的大规模数据集。神经信息处理系统进展 (NeurIPS)，第 36 卷，2024。

[38] Shi, T., Karpathy, A., Fan, L., Hernandez, J., and Liang, P. World of bits: An open-domain platform for web-based agents. In International Conference on Machine Learning, pp. 3135-3144. PMLR, 2017.

Shi, T., Karpathy, A., Fan, L., Hernandez, J., 和 Liang, P. World of bits: 一个面向网络代理的开放域平台。载于国际机器学习大会论文集，页 3135-3144。PMLR, 2017。

[39] Sodhi, P., Branavan, S., and McDonald, R. Heap: Hierarchical policies for web actions using llms. arXiv preprint arXiv:2310.03720, 2023.

Sodhi, P., Branavan, S., 和 McDonald, R. Heap: 利用大型语言模型的分层网页操作策略。arXiv 预印本 arXiv:2310.03720, 2023。

[40] Song, C. H., Wu, J., Washington, C., Sadler, B. M., Chao, W.-L., and Su, Y. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 2998- 3009, 2023.

Song, C. H., Wu, J., Washington, C., Sadler, B. M., Chao, W.-L., 和 Su, Y. Llm-planner: 基于少量示例的具身代理大型语言模型规划。载于 IEEE/CVF 国际计算机视觉大会论文集，页 2998-3009，2023。

[41] Song, Y., Xu, F. F., Zhou, S., and Neubig, G. Beyond browsing: Api-based web agents. 2024.

Song, Y., Xu, F. F., Zhou, S., 和 Neubig, G. 超越浏览: 基于 API 的网络代理。2024。

[42] Sridhar, A., Lo, R., Xu, F. F., Zhu, H., and Zhou, S. Hierarchical prompting assists large language model on web navigation. arXiv preprint arXiv:2305.14257, 2023.

Sridhar, A., Lo, R., Xu, F. F., Zhu, H., 和 Zhou, S. 分层提示辅助大型语言模型进行网页导航。arXiv 预印本 arXiv:2305.14257, 2023。

[43] Sun, H., Zhuang, Y., Kong, L., Dai, B., and Zhang, C. Adaplanner: Adaptive planning from feedback with language models. Advances in neural information processing systems, 36:58202-58245, 2023.

Sun, H., Zhuang, Y., Kong, L., Dai, B., 和 Zhang, C. Adaplanner: 基于反馈的自适应规划结合语言模型。神经信息处理系统进展 (NeurIPS)，第 36 卷，页 58202-58245，2023。

[44] Sutton, R. S. Reinforcement learning: An introduction. A Bradford Book, 2018.

Sutton, R. S. 强化学习: 导论。A Bradford Book, 2018。

[45] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model, 2023.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., 和 Hashimoto, T. B. Stanford alpaca: 一个遵循指令的 LLaMA 模型，2023。

[46] Team, Q. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https: //qwenlm.github.io/blog/qwq 32b/.

团队，Q. Qwq-32b: 拥抱强化学习的力量，2025 年 3 月。网址 https: //qwenlm.github.io/blog/qwq-32b/。

[47] Wang, K., Zhu, J., Ren, M., Liu, Z., Li, S., Zhang, Z., Zhang, C., Wu, X., Zhan, Q., Liu, Q., et al. A survey on data synthesis and augmentation for large language models. arXiv preprint arXiv:2410.12896, 2024.

王克、朱军、任明、刘志、李松、张震、张超、吴翔、詹强、刘强等。大型语言模型的数据合成与增强综述。arXiv 预印本 arXiv:2410.12896，2024 年。

[48] Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., and Lim, E.-P. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. arXiv preprint arXiv:2305.04091, 2023.

王磊、徐伟、兰燕、胡志、兰燕、李瑞康-王、林恩-培。计划与解决提示: 通过大型语言模型提升零样本链式思维推理。arXiv 预印本 arXiv:2305.04091，2023 年。

[49] Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language models with self-generated instructions. arXiv preprint arXiv:2212.10560, 2022.

王洋、科尔迪、米什拉、刘安、史密斯、哈沙比、大卫、哈吉希尔齐。自我指导: 通过自生成指令对齐语言模型。arXiv 预印本 arXiv:2212.10560，2022 年。

[50] Wang, Z. Z., Mao, J., Fried, D., and Neubig, G. Agent workflow memory. arXiv preprint arXiv:2409.07429, 2024.

王志忠、毛军、弗里德、大卫、纽比格。代理工作流记忆。arXiv 预印本 arXiv:2409.07429，2024 年。

[51] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824-24837, 2022.

魏军、王翔、舒尔曼斯、博斯玛、夏飞、池恩、乐奇-维、周东等。链式思维提示激发大型语言模型的推理能力。神经信息处理系统进展，35:24824-24837，2022 年。

[52] Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Hua, T. J., Cheng, Z., Shin, D., Lei, F., et al. Os-world: Benchmarking multimodal agents for open-ended tasks in real computer environments. arXiv preprint arXiv:2404.07972, 2024.

谢涛、张东、陈杰、李晓、赵松、曹锐、华天骄、程志、申东、雷峰等。Os-world: 在真实计算机环境中评测多模态代理的开放式任务能力。arXiv 预印本 arXiv:2404.07972，2024 年。

[53] Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C., and Jiang, D. Wizardlm: Empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244, 2023.

徐超、孙强、郑凯、耿翔、赵鹏、冯军、陶成、姜东。WizardLM: 赋能大型语言模型以执行复杂指令。arXiv 预印本 arXiv:2304.12244，2023 年。

[54] Yang, K., Liu, Y., Chaudhary, S., Fakoor, R., Chaud-hari, P., Karypis, G., and Rangwala, H. Agentoccam: A simple yet strong baseline for llm-based web agents. arXiv preprint arXiv:2410.13825, 2024.

杨凯、刘洋、乔达里、法库尔、乔达里、卡里皮斯、兰格瓦拉。AgentOccam: 基于大型语言模型的网页代理的简单而强大的基线。arXiv 预印本 arXiv:2410.13825，2024 年。

[55] Yang, Z., Li, P., Yan, M., Zhang, J., Huang, F., and Liu, Y. React meets actre: Autonomous annotations of agent trajectories for contrastive self-training. arXiv preprint arXiv:2403.14589, 2024.

杨志、李鹏、严明、张杰、黄飞、刘洋。React 遇见 ActRE: 用于对比自训练的代理轨迹自主注释。arXiv 预印本 arXiv:2403.14589，2024 年。

[56] Yao, S., Chen, H., Yang, J., and Narasimhan, K. Webshop: Towards scalable real-world web interaction with grounded language agents. arXiv preprint arxiv:2207.01206, 2022.

姚帅、陈浩、杨军、纳拉辛汉。WebShop: 面向可扩展的真实网络交互的基于语言的代理。arXiv 预印本 arXiv:2207.01206，2022 年。

[57] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629, 2022.

姚帅、赵军、余东、杜宁、沙夫兰、纳拉辛汉、曹阳。React: 语言模型中推理与行动的协同。arXiv 预印本 arXiv:2210.03629，2022 年。

[58] Zhang, C., Yang, Z., Liu, J., Han, Y., Chen, X., Huang, Z., Fu, B., and Yu, G. Appagent: Multimodal agents as smartphone users. arXiv preprint arXiv:2312.13771, 2023.

张超、杨志、刘军、韩阳、陈晓、黄震、傅斌、余刚。AppAgent: 作为智能手机用户的多模态代理。arXiv 预印本 arXiv:2312.13771，2023 年。

[59] Zhang, C., He, S., Qian, J., Li, B., Li, L., Qin, S., Kang, Y., Ma, M., Lin, Q., Rajmohan, S., et al. Large language model-brained gui agents: A survey. arXiv preprint arXiv:2411.18279, 2024.

张超、何松、钱军、李斌、李磊、秦松、康阳、马明、林强、拉吉莫汉等。大型语言模型驱动的 GUI 代理综述。arXiv 预印本 arXiv:2411.18279，2024 年。

[60] Zhang, Y., Ma, Z., Ma, Y., Han, Z., Wu, Y., and Tresp, V. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. arXiv preprint arXiv:2408.15978, 2024.

张勇、马志、马颖、韩志、吴颖和特雷斯普 (Tresp, V.). Webpilot: 一个多功能且自主的多智能体系统，用于网页任务执行与策略性探索。arXiv 预印本 arXiv:2408.15978, 2024.

[61] Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., et al. Webarena: A realistic web environment for building autonomous agents. arXiv preprint arXiv:2307.13854, 2023.

周思、徐飞飞、朱浩、周翔、罗睿、斯里达尔、程翔、欧涛、比斯克、弗里德等。Webarena: 一个用于构建自主代理的真实网络环境。arXiv 预印本 arXiv:2307.13854，2023 年。

A. Appendix

A. 附录

A.1. Planner and Executor Output Examples

A.1. 规划器与执行器输出示例

- Task: ”From my stay at Homewood Suites Southpointe, what’s the estimated driving time to reach PPG Paints Arena?”

- 任务:“从我在 Homewood Suites Southpointe 的住宿出发，预计开车到 PPG Paints Arena 需要多长时间?”

- Planner Output

- 规划器输出

##Step 1

## 步骤 1

Reasoning: From the initial HTML state, I observe that I am on the

推理: 从初始的 HTML 状态来看，我注意到我处于

OpenStreetMap website. The page includes a search input field and a

OpenStreetMap 网站。页面包含一个搜索输入框和一个

direction sign icon that likely allows users to find directions

方向标志图标，可能帮助用户查找路线

between two points. The first step is to initiate the process of

在两点之间。第一步是启动该过程

finding directions.

寻找方向。

Step: Initiate the process of finding directions.

步骤: 启动寻找方向的过程。

##Step 2

## 步骤 2

Reasoning: After initiating the direction-finding process, the next

推理: 在启动定向过程后，接下来

step is to input the starting location, which is ’Homewood Suites

第一步是输入起始位置，即“Homewood Suites”

Southpointe’. This will set the starting point for the directions.

“Southpointe”。这将设定路线的起点。

Step: Input ’Homewood Suites Southpointe’ as the starting location.

步骤: 输入“Homewood Suites Southpointe”作为起始位置。

##Step 3

## 步骤 3

Reasoning: With the starting location set, the next step is to input

推理: 设定起始位置后，下一步是输入

the destination, which is PPG Paints Arena’. This will set the

目的地，即“PPG Paints Arena”。这将设定

endpoint for the directions.

路线的终点。

Step: Input ’PPG Paints Arena’ as the destination.

步骤: 输入“PPG Paints Arena”作为目的地。

##Step 4

## 步骤 4

Reasoning: After setting both the starting location and the

推理: 在设置起点和

destination, the next step is to initiate the search for

终点后，下一步是开始搜索

directions. This will generate the directions between the two

路线。这将生成两点之间的路线指引。

points.

点。

Step: Initiate the search for directions.

步骤: 开始搜索路线指引。

##Step 5

## 步骤 5

Reasoning: Once the directions are generated, the final step is to

推理: 路线生成后，最后一步是

analyze the results to determine the estimated driving time from

分析结果，以确定从

Homewood Suites Southpointe to PPG Paints Arena. The exact time

Homewood Suites Southpointe 到 PPG Paints Arena 的预计驾车时间。具体时间

will be displayed on the results page.

将在结果页面显示。

Step: Analyze the results to determine the estimated driving time from

步骤: 分析结果以确定从

Homewood Suites Southpointe to PPG Paints Arena.

Homewood Suites Southpointe 到 PPG Paints Arena 的预计驾车时间。

Executor Output

执行器输出

* Output 1:

* 输出 1:

#Element: the direction sign on the right side of the Go icon

# 元素:Go 图标右侧的方向标志

do(action=”Click”, element=”15”) }

do(action=”Click”, element=”15”) }

---

* Output 2:

* 输出 2:

#Element: The field at the beginning, located in the upper left

# 元素: 位于左上角的起始字段

corner

角落

do (action=”Type”, argument=”Homewood Suites Southpointe”, do (action=”Type”, argument=”Homewood Suites Southpointe”,

element=”13”) }

element=”13”) }

* Output 3:

* 输出 3:

PLAN-AND-ACT: Improving Planning of Agents for Long-Horizon Tasks

PLAN-AND-ACT: 改进长远任务中智能体的规划

#Element: The field labeled ’To’ at the top left

# 元素: 左上方标记为“收件人”的字段

do(action=”Type”, argument=”PPG Paints Arena”, element=”14”) }

do(action=”Type”, argument=”PPG Paints Arena”, element=”14”) }

* Output 4:

* 输出 4:

#Element: The Go button

# 元素: 开始按钮

do(action=”Click”, element=”16”) }

do(action=”Click”, element=”16”) }

* Output 5:

* 输出 5:

#Note: Distance: 29km. Time: 0:34.

# 注: 距离:29 公里。时间:0:34。

exit (message=”The estimated driving time from Homewood Suites

exit (message=” 从 Homewood Suites 估计的驾车时间

Southpointe to PPG Paints Arena is 0:34.”)

Southpointe 到 PPG 油漆竞技场需时 0:34。

- Task: ”Create a shipping report from 08/05/2022 to 03/01/2023”

- 任务:“创建 2022 年 8 月 5 日至 2023 年 3 月 1 日的运输报告”

Planner Output

规划器输出

##Step 1

## 步骤 1

Reasoning: From the initial HTML state, I observe that I am on the

推理: 从初始的 HTML 状态来看，我注意到我处于

Magento Admin Panel homepage. The page has a left sidebar

Magento 管理面板主页。页面左侧有一个侧边栏

navigation menu with several sections including ’Dashboard’,

包含“仪表盘”等多个部分的导航菜单,

’Sales’, ’Catalog’, ’Customers’, ’Marketing’, ’Content’, ’Reports’,

“销售”,“目录”,“客户”,“市场营销”,“内容”,“报告”,

’Stores’, and ’System’. Each section appears to be clickable, and

“商店”和“系统”。每个部分似乎都可以点击，且

some have nested subsections visible through expandable lists. The

有些通过可展开列表显示嵌套的子章节。

main content area shows a dashboard with various widgets including

主内容区域显示包含多个小部件的仪表盘

revenue statistics, tax information, shipping details, and quantity

收入统计、税务信息、运输详情和数量

metrics. To create a shipping report, we need to navigate to the

指标。要创建运输报告，我们需要导航到

’Reports’ section in the left sidebar and then access the

左侧边栏中的“报告”部分，然后访问

’Shipping’ report option.

“发货”报告选项。

Step: Navigate to the ’Reports’ section and access the ’Shipping’

步骤: 导航到“报告”部分并进入“运输”

report option.

报告选项。

##Step 2

## 步骤 2

Reasoning: To generate the shipping report for the specified date

推理: 生成指定日期的运输报告

range, we need to input the ’From’ and ’To’ dates. The ’From’ date

范围，我们需要输入“起始”(From) 和“结束”(To) 日期。“起始”日期

should be set to 08/05/2022 and the ’To’ date should be set to

应设置为 08/05/2022，且“至”日期应设置为

03/01/2023. This will ensure that the report covers the desired

2023 年 3 月 1 日。这将确保报告涵盖所需内容

period.

周期。

Step: Input the date range for the shipping report from 08/05/2022 to

步骤: 输入发货报告的日期范围，从 2022 年 08 月 05 日到

03/01/2023.
##Step 3

## 步骤 3

Reasoning: After setting the date range, we need to generate the report

推理: 设置日期范围后，我们需要生成报告

by clicking the ’Show Report’ button. This will display the

点击“显示报告”按钮。这将显示

shipping report for the specified period.

指定期间的运输报告。

Step: Generate the shipping report by clicking the ’Show Report’ button.

步骤: 点击“显示报告”按钮生成运输报告。

##Step 4

## 步骤 4

Reasoning: Once the report is generated, we need to ensure that all

推理: 报告生成后，我们需要确保所有

relevant information is visible. This step ensures that the

相关信息可见。此步骤确保

complete report details are accessible.

完整的报告详情可访问。

Step: Ensure that the complete shipping report details are visible.

步骤: 确保完整的运输报告详情可见。

##Step 5

## 步骤 5

Reasoning: Finally, we need to exit the task by providing a message

推理: 最后，我们需要通过提供一条消息来退出任务

that the shipping report for the specified date range has been

说明指定日期范围内的运输报告已被

created. This step concludes the task and informs the user of the

创建。此步骤结束任务并通知用户

successful completion.

成功完成。

PLAN-AND-ACT: Improving Planning of Agents for Long-Horizon Tasks

PLAN-AND-ACT: 改进长远任务中代理的规划

Step: Exit with a message indicating that the shipping report for the

步骤: 以消息退出，指明指定日期范围内的运输报告已被创建。

specified date range has been created.

- Executor Output

  - 执行器输出

* Output 1:

* 输出 1:

#Element: the ’Reports’ tab in the left sidebar menu

# 元素: 左侧边栏菜单中的“报告”标签

do(action=”Click”, element=”7”))

do(action=”Click”, element=”7”))

* Output 2:

* 输出 2:

#Element: the ’Shipping’ link under the ’Sales’ section

# 元素:“销售”部分下的“运输”链接

do(action=”Click”, element=”18”) }

do(action=” 点击”, element=”18”) }

* Output 3:

* 输出 3:

#Element: the ’From’ input field in the ’Filter’ section

# 元素:“筛选”部分中的“起始日期”输入框

do (action=”Type”, argument=”08/05/2022”, element=”24”) }

do (action=” 输入”, argument=”08/05/2022”, element=”24”) }

* Output 4:

* 输出 4:

#Element: the ’To’ date input field in the ’Filter’ section

# 元素:“筛选”部分中的“结束日期”输入框

do(action=”Type”, argument=”03/01/2023”, element=”27”) }

do(action=” 输入”, argument=”03/01/2023”, element=”27”) }

---

* Output 5:

* 输出 5:

#Element: the ’Show Report’ button located at the top right area of

# 元素:“运输报告”部分右上方的“显示报告”按钮

the ’Shipping Report’ section

do(action=”Click”, element=”16”) }

do(action=” 点击”, element=”16”) }

---

* Output 6:

* 输出 6:

---

do (action=”Scroll Down”)

do (action=” 向下滚动”)

---

* Output 7:

* 输出 7:

exit (message=”The shipping report from 08/05/2022 to 03/01/2023 has been created.”)

exit (message=” 从 08/05/2022 到 03/01/2023 的运输报告已生成。”)

A.2. Replanner Examples

A.2. 重新计划示例

A.2.1. QUERY REFINEMENT

A.2.1. 查询细化

This section presents an example of how a replanner refines an initial plan when an execution failure occurs.

本节展示了当执行失败发生时，重新规划器如何优化初始计划的示例。

Task: ”How long does it take to walk from Carnegie Museum of Art to a library at CMU?”

任务:“从卡内基艺术博物馆步行到卡内基梅隆大学 (CMU) 图书馆需要多长时间?”

- Initial Plan

- 初步计划

##Step 1

## 步骤 1

Reasoning: From the initial HTML state, I observe that I am on the

推理: 从初始的 HTML 状态来看，我注意到我处于

OpenStreetMap website. The page includes a search input field. The

OpenStreetMap 网站。页面包含一个搜索输入框。

first step is to search for libraries at CMU to identify which one

第一步是在 CMU 搜索库以确定使用哪一个

we’ll get directions to.

我们将获得前往的路线。

Step: Search for libraries at CMU.

步骤: 在卡内基梅隆大学 (CMU) 搜索图书馆。

##Step 2

## 步骤 2

Reasoning: After finding the libraries at CMU, we need to analyze the

推理: 在找到卡内基梅隆大学 (CMU) 的图书馆后，我们需要分析

search results to identify the available libraries and select one for

搜索结果以确定可用的库并选择其中一个用于

getting directions.

获取路线指引。

Step: Review the search results and select a library at CMU.

步骤: 查看搜索结果并选择卡内基梅隆大学 (CMU) 的一家图书馆。

##Step 3

## 步骤 3

Reasoning: Now that we have identified a specific library, we can initiate

推理: 既然我们已经确定了一个具体的图书馆，就可以开始

the process of finding directions between the two points.

查找两点之间路线的过程。

Step: Initiate the process of finding directions.

步骤: 启动查找路线的过程。

##Step 4

## 步骤 4

Reasoning: After initiating the direction-finding process, we need to input

推理: 启动路线查找过程后，我们需要输入

the starting location, which is ’Carnegie Museum of Art’.

起点位置，即“卡内基艺术博物馆”(Carnegie Museum of Art)。

Step: Input ’Carnegie Museum of Art’ as the starting location.

步骤: 输入“卡内基艺术博物馆”作为起点位置。

##Step 5

## 步骤 5

Reasoning: With the starting location set, we need to input the selected

推理: 设置好起点后，我们需要输入选定的

library as the destination.

图书馆作为终点。

Step: Input the selected CMU library as the destination.

步骤: 输入选定的卡内基梅隆大学 (CMU) 图书馆作为终点。

##Step 6

## 步骤 6

Reasoning: To ensure that the directions provided are for walking, we need

推理: 为了确保提供的是步行路线，我们需要

to select the appropriate option for walking directions.

选择步行路线的相应选项。

Step: Select the option for walking directions.

步骤: 选择步行路线选项。

##Step 7

## 步骤 7

Reasoning: After all parameters are set, we can analyze the results to

推理: 在所有参数设置完成后，我们可以分析结果以

determine the walking time between the two locations.

确定两个地点之间的步行时间。

Step: Analyze the results to determine the walking time from Carnegie

步骤: 分析结果以确定从卡内基艺术博物馆 (Carnegie Museum of Art)

Museum of Art to the selected CMU library.

到所选卡内基梅隆大学 (CMU) 图书馆的步行时间。

- Executor's Actions: The executor attempts to search for libraries at CMU by performing the following actions:

- 执行者操作: 执行者尝试通过以下操作搜索卡内基梅隆大学的图书馆:

#Element: the search bar

# 元素: 搜索栏

do(action=”Search”, argument=”Library at CMU”, element=”13”)

执行 (action=”Search”，argument=”Library at CMU”，element=”13”)

#Element: the search button

# 元素: 搜索按钮

do (action=”Click”, element=”13”)

执行 (action=”Click”，element=”13”)

However, upon executing this step, the system receives the following **HTML response**, which indicates that no search results were found:

然而，执行此步骤后，系统收到以下 **HTML 响应 **，表明未找到搜索结果:

```
<html>
    <li id=”17” data-text=”No results found”> No results found </li>
```

```
<li id=”17” data-text=”No results found”> 未找到结果 </li>
```

```
    ...
</html>
```

This failure necessitates a refinement of the plan.

此失败需要对计划进行调整。

- Replanning - Refining the Search Query: Since the initial search query (”Library at CMU”) returned no results, the replanner modified the query to a more general term: ”Library near CMU” to improve the likelihood of retrieving relevant results.

- 重新规划 - 优化搜索查询: 由于初始搜索查询 (“Library at CMU”) 未返回结果，重新规划者将查询修改为更通用的词语: “Library near CMU”，以提高检索相关结果的可能性。

- Refined Plan

- 优化后的计划

##Step 1

## 步骤 1

Reasoning: From the current HTML state, I observe that the page shows

推理: 从当前 HTML 状态来看，我观察到页面显示

search results with ’ No results found’ (element with id=’17’). The

搜索结果显示“未找到结果”(元素 id=’17’)。

page contains a search bar (id=’12’) and a Go button (id=’13’) for

页面包含一个搜索栏 (id=’12’) 和一个“前往”按钮 (id=’13’) 用于

initiating new searches. There’s also a direction sign icon

正在启动新的搜索。还有一个方向指示标志图标

(id=’14’) that likely allows for finding directions between two

(id=’14’) 这很可能允许在两点之间寻找路径

points. The main content area provides information about

点数。主要内容区域提供有关

OpenStreetMap and its contributors. Looking at the previous

OpenStreetMap 及其贡献者。回顾之前的

actions, I can see that the user attempted to search for ’Library

操作，我可以看到用户尝试搜索“Library”

at CMU′ and clicked the Go button, but this broad search query

在 CMU′ 点击了“前往”按钮，但这个广泛的搜索查询

including ’Library at CMU’ returned no results. This indicates that

包括“CMU 图书馆”未返回任何结果。这表明

we need to modify our approach by simplifying the search terms.

我们需要通过简化搜索词来调整我们的方法。

Given these observations and the failed previous search attempt, we

鉴于这些观察结果以及之前搜索尝试的失败，我们

need to refine our search query by removing the ’Library at CMU’

需要通过去除“CMU 图书馆”来优化我们的搜索查询

specification and focusing more specifically on libraries near CMU

规范，并更具体地关注卡内基梅隆大学 (CMU) 附近的图书馆

to get more relevant results.

以获得更相关的结果。

Step: Refine the search query by focusing on libraries near CMU.

步骤: 通过关注卡内基梅隆大学 (CMU) 附近的图书馆来优化搜索查询。

##Step 2

## 步骤 2

Reasoning: After refining the search query, we need to analyze the

推理: 在优化搜索查询后，我们需要分析

search results to identify potential libraries that are near CMU.

搜索结果，以识别靠近卡内基梅隆大学 (CMU) 的潜在图书馆。

This will help in determining the closest library based on the

这将有助于根据

search results.

搜索结果确定最近的图书馆。

Step: Analyze the search results to identify potential libraries near

步骤: 分析搜索结果以识别靠近

CMU.
##Step 3

## 步骤 3

Reasoning: With the list of potential libraries identified, the next

推理: 在确定潜在图书馆列表后，下一步

step is to get directions to each library from Carnegie Museum of

是获取从卡内基艺术博物馆 (Carnegie Museum of Art) 到每个图书馆的路线。

Art. This will help in determining the distance and time required

这将有助于确定到达每个图书馆所需的距离和时间,

to reach each library, allowing us to identify the closest one.

从而识别出最近的图书馆。

Step: Get walking directions to each identified library from Carnegie

步骤: 获取从卡内基艺术博物馆到每个已识别图书馆的步行路线。

Museum of Art.

。

##Step 4

## 步骤 4

Reasoning: After obtaining the distances and times for each library,

推理: 在获得每个图书馆的距离和时间后,

the final step is to identify the closest library based on the

最后一步是根据记录的距离确定最近的图书馆,

recorded distances and provide the result to the user.

并将结果提供给用户。

Step: Identify the closest library based on the recorded distances and

步骤: 根据记录的距离确定最近的图书馆，并

provide the result to the user.

向用户提供结果。

- Executor's Next Action According to the Refined Plan: To reflect the updated strategy, the executor modifies its search action:

- 执行者根据优化计划的下一步行动: 为了反映更新后的策略，执行者调整了其搜索动作:

---

do (action="Search", argument="Library near CMU", element="12")

执行 (action="Search", argument="CMU 附近的图书馆", element="12")

A.2.2. ANALYZING SEARCH RESULTS AND MEMORY (AGGREGATING RESULTS ACROSS MULTIPLE PAGES)

A.2.2. 分析搜索结果与记忆 (跨多页汇总结果)

This section presents an example of how a replanner refines an initial plan when a broad high-level step, such as "analyze the search results," needs to be refined into concrete actions.

本节展示了当一个宽泛的高级步骤，如“分析搜索结果”，需要细化为具体操作时，重新规划器如何完善初始计划的示例。

Task: "Present the monthly count of successful orders from Jan to December 2022 in MM:COUNT format"

任务:“以 MM: 数量格式展示 2022 年 1 月至 12 月的成功订单月度统计”

- Initial Plan

- 初步计划

##Step 1

## 步骤 1

Reasoning: From the initial HTML state, I observe that I am on the Magento

推理: 从初始的 HTML 状态来看，我注意到我处于 Magento 平台上

Admin Panel homepage. The page has a left sidebar navigation menu with

管理面板主页。页面左侧有一个侧边栏导航菜单

several sections including ’Dashboard’, ’Sales’, ’Catalog’,

包括“仪表盘”、“销售”、“目录”等多个部分，

’Customers’, ’Marketing’, ’Content’, ’Reports’, ’Stores’, and ’System’.

“客户”、“营销”、“内容”、“报告”、“门店”和“系统”。

Each section appears to be clickable and some have nested subsections

每个部分似乎都可以点击，有些部分还包含嵌套的小节

visible through expandable lists. The main content area shows a

通过可展开列表可见。主内容区域显示一个

dashboard with various widgets including revenue statistics, tax

包含收入统计、税务等多种小部件的仪表盘

information, shipping details, and quantity metrics. To access the

信息、运输详情和数量指标。要访问

order reports, we need to navigate through the ’Reports’ section in the

要查看订单报告，我们需要进入“报告”部分

left sidebar, where we can then find and select the ’Orders’ report

左侧边栏，我们可以在那里找到并选择“订单”报告

option under the ’Sales’ section.

“销售”部分下的选项。

Step: Navigate to the ’Orders’ report section under the ’Reports’ section.

步骤: 导航至“报告”部分下的“订单”报告栏目。

##Step 2

## 步骤 2

Reasoning: To generate the report for the specified period, we need to set

推理: 为了生成指定期间的报告，我们需要设置

the date range from January 1, 2022, to December 31, 2022. This

日期范围从 2022 年 1 月 1 日到 2022 年 12 月 31 日。

involves entering the start and end dates in the respective date picker

涉及在相应的日期选择器中输入开始和结束日期

input fields.

<input fields.> 输入字段。

Step: Set the date range from January 1, 2022, to December 31, 2022.

步骤: 设置日期范围为 2022 年 1 月 1 日至 2022 年 12 月 31 日。

##Step 3

## 步骤 3

Reasoning: To ensure the report is generated for the correct period, we

推理: 为了确保报告生成的时间段正确，我们

need to select the ’Month’ option from the ’Period’ dropdown. This will

需要从“期间”下拉菜单中选择“月份”选项。这将

aggregate the data on a monthly basis.

按月汇总数据。

Step: Select the ’Month’ option from the ’Period’ dropdown.

步骤: 从“期间”下拉菜单中选择“月份”选项。

##Step 4

## 步骤 4

Reasoning: After setting the date range and period, we need to generate the

推理: 设置日期范围和周期后，我们需要生成

report by clicking the ’ Show Report’ button. This will display the

报告，方法是点击“显示报告”按钮。这将显示

order report for the specified period.

指定期间的订单报告。

Step: Generate the report by clicking the ’Show Report’ button.

步骤: 点击“显示报告”按钮生成报告。

##Step 5

## 步骤 5

Reasoning: Once the report is generated, we need to analyze the results to

推理: 报告生成后，我们需要分析结果，

identify the monthly count of successful orders from January to

确定 2022 年 1 月至 12 月每月成功订单的数量。

December 2022. This involves viewing the report and noting the counts

这包括查看报告并记录每个月的数量。

for each month.

Step: Analyze the report results to identify the monthly count of

步骤: 分析报告结果，确定 2022 年 1 月至 12 月每月成功订单数量。

successful orders from January to December 2022.

##Step 6

## 步骤 6

Reasoning: After identifying the monthly counts, we need to provide the

推理: 确定每月数量后，我们需要以“MM:COUNT”格式

results in the ’MM:COUNT’ format. This involves exiting with a message

提供结果。这包括退出时带有包含 1 月至 12 月每月数量的消息。

containing the monthly counts for each month from January to December

2022.

Step: Exit with a message containing the monthly counts of successful

步骤: 退出时输出包含 2022 年 1 月至 12 月成功订单月度统计的消息，格式为“MM:COUNT”。

orders from January to December 2022 in the ’MM:COUNT’ format.

订单数量，格式为“MM:COUNT”。

- Executor’s Actions: The executor follows the plan up until the broad ”analyze the report results” step:

- 执行者操作: 执行者按照计划执行，直到“分析报告结果”这一宽泛步骤:

#Element: the ’REPORTS’ sectiondo (action=”Click”, element=”7”)

# 元素: 点击“报告 (REPORTS)”部分 (action=”Click”, element=”7”)

#Element: the ’Orders’ report optiondo (action=”Click”, element=”15”)

# 元素: 点击“订单 (Orders)”报告选项 (action=”Click”, element=”15”)

#Element the ’From’ datedo(action=”Type”, argument=”01/01/2022”,

# 元素: 在“起始日期 (From)”输入框输入 (action=”Type”, argument=”01/01/2022”,

element=”24”)

element=”24”)

#Element the ’To’ datedo(action=”Type”, argument=”12/31/2022”,

# 元素: 在“结束日期 (To)”输入框输入 (action=”Type”, argument=”12/31/2022”,

element=”27”)

element=”27”)

#Element: ’Month’ in the dropdown

# 元素: 下拉菜单中选择“月份 (Month)”

PLAN-AND-ACT: Improving Planning of Agents for Long-Horizon Tasks

PLAN-AND-ACT: 改进长周期任务中代理的规划

---

do (action=”Select Dropdown Option”, argument=”Month”, element=”22”)

执行 (action=”Select Dropdown Option”, argument=”Month”, element=”22”)

#Element: ’Show Report’

# 元素:“显示报告 (Show Report)”

do(action=”Click”, element=”16”)

执行 (action=”Click”, element=”16”)

The executor reaches the analysis step, but the broad instruction to ”analyze the report” needs refinement.

执行者到达分析步骤，但“分析报告”这一宽泛指令需要细化。

- Replanning - Refining the Analysis Task: The replanner observes that the executor requires step-by-step guidance to extract the relevant order counts. It refines the broad analysis step into structured data extraction.

- 重新规划 - 细化分析任务: 重新规划者观察到执行者需要逐步指导以提取相关订单数量，将宽泛的分析步骤细化为结构化数据提取。

- First Replan

- 第一次重新规划

- First step of the replan:

- 重新规划的第一步:

##Step 1

## 步骤 1

Reasoning: The report results are now visible, and I need to analyze

推理: 报告结果现在可见，我需要分析

the order counts for each month. The report is structured as a

每个月的订单数量。报告结构为

list, where each row contains a month and the corresponding order

列表，每行包含一个月份及对应的订单

count. The first step is to start extracting data from the

数量。第一步是开始从

initially visible rows. The extracted counts so far are:

最初可见的行中提取数据。到目前为止提取的数量是:

01:11, 02:16, 03:14, 04:8
Step: Extract the order counts from the initially visible rows.

步骤: 从最初可见的行中提取订单数量。

- Next executor action:

- 下一执行者操作:

---

do (action=”Scroll Down”)

执行 (action=” 向下滚动”)

- Second Replan

  - 第二次重新规划

- First step of the replan:

  - 重新规划的第一步:

##Step 1

## 步骤 1

Reasoning: After capturing the first set of order counts, I need to

推理: 捕获第一批订单数量后，我需要

scroll down to reveal more entries. As new rows become visible, I

向下滚动以显示更多条目。随着新行变得可见，我

will extract and record their data. The additional extracted counts

将提取并记录它们的数据。额外提取的计数为

are:

:

05:8, 06:13, 07:9, 08:8, 09:10, 10:4
Step: Scroll down and extract the next set of order counts.

步骤: 向下滚动并提取下一组订单计数。

- Next executor action:

  - 下一执行动作:

---

do (action="Scroll Down")

执行 (action="Scroll Down")

- Third Replan

  - 第三次重新计划

- First step of the replan:

  - 重新计划的第一步:

##Step 1

## 步骤 1

Reasoning: The final set of order counts needs to be retrieved. I will

推理: 需要检索最后一组订单计数。我将

scroll down again, ensuring I capture all remaining visible data.

再次向下滚动，确保捕获所有剩余可见数据。

The final extracted counts are:

最终提取的计数为:

11:11, 12:10
Step: Perform a final scroll to capture any remaining order counts.

步骤: 执行最后一次滚动以捕获任何剩余的订单计数。

- Next executor action:

  - 下一执行动作:

---

exit (message="The monthly count of successful orders from Jan to

退出 (message="2022 年 1 月至 12 月成功订单的月度计数为:

December 2022 is: 01:11, 02:16, 03:14, 04:8, 05:8, 06:13, 07:9,

01:11， 02:16， 03:14， 04:8， 05:8， 06:13， 07:9，

08:8, 09:10, 10:4, 11:5, 12:10”)

A.3. Planner Prompt

A.3. 计划者提示

Planner has a system prompt where its high-level goals are listed; in the user message, it sees the current user query and the initial HTML state.

规划器有一个系统提示，其中列出了其高级目标；在用户消息中，它看到当前的用户查询和初始的 HTML 状态。

A.3.1. PLANNER SYSTEM PROMPT

A.3.1. 规划系统提示

##Goal

目标

You are the Global Planner agent, an expert plan generator for web navigation

您是全球规划代理，一个专门为网页导航生成计划的专家

tasks. You will be provided with the following information:

任务。您将获得以下信息:

- **User Query**: The web task that you are required to generate a global plan

- ** 用户查询 **: 您需要生成一个全局计划的网络任务

for.

为了。

- **Initial HTML State**: The initial HTML state of the web page.

- ** 初始 HTML 状态 **: 网页的初始 HTML 状态。

You are responsible for analyzing the usery query and the initial HTML state

您负责分析用户查询和初始 HTML 状态

to generate a structured, step-by-step global plan that outlines the

生成一个结构化的、逐步的全球计划，概述

high-level steps to complete the user query. The global plan that you

完成用户查询的高级步骤。您所制定的全局计划

generate shouldn’t directly describe low-level web actions such as clicks

生成不应直接描述诸如点击之类的低级网页操作

or types (unless necessary for clarity) but outline the high-level steps

或类型 (除非为清晰起见必要) 但概述高级步骤

that encapsulate one or more actions in the action trajectory, meaning each

封装动作轨迹中一个或多个动作，这意味着每个

step in your plan will potentially require multiple actions to be

你计划中的每一步可能都需要多项操作来完成

completed. Your global plan will then be handed to an Executor agent which

完成。您的全球计划随后将交由执行代理处理

will perform low-level web actions on the webpage (click, type, hover, and

将在网页上执行低级别的网页操作 (点击、输入、悬停等)

more) to convert your global plan into a sequence of actions and complete

更多) 将您的全球计划转化为一系列行动并完成

the user query.

用户查询。

##Expected Output Format

## 预期输出格式

The global plan you generate should be structured in a numbered list format,

您生成的全球计划应以编号列表格式结构化,

starting with '## Step 1' and incrementing the step number for each

从“## 第 1 步”开始，每一步的编号依次递增

subsequent step. Each step in the plan should be in this exact format:

后续步骤。计划中的每一步都应采用此确切格式:

...
##Step N

## 步骤 N

Reasoning: [Your reasoning here]

推理:[您的推理内容]

Step: [Your step here]

步骤:[您的步骤]

...
Here is a breakdown of the components you need to include in each step of your

以下是您在每个步骤中需要包含的组件细目

global plan as well as their specific instructions:

整体计划及其具体指示:

- **Reasoning**: In this section, you should explain your reasoning and

- ** 推理 **: 在本节中，您应解释您的推理过程并

thought process behind the step you are proposing. It should provide a

你所提出步骤背后的思考过程。它应当提供一个

high-level justification for why the actions in this step are grouped

对本步骤中各操作分组的高级理由说明

together and how they contribute to achieving the overall goal. Your

共同协作及其如何助力实现整体目标。您的

reasoning should be based on the information available in the user query

推理应基于用户查询中提供的信息

(and potentially on the initial HTML state) and should guide the Executor

(以及可能基于初始 HTML 状态) 并应指导执行器 (Executor)

agent in understanding the strategic decision-making process behind your

帮助理解您背后的战略决策过程的代理

global plan.

全球计划。

- **Step**: In this section, you should provide a concise description of the

- ** 步骤 **: 在本节中，您应简要描述

global step being undertaken. Your step should summarize one or more

正在进行的全局步骤。您的步骤应总结一个或多个

actions as a logical unit. It should be as specific and concentrated as

动作作为一个逻辑单元。它应当尽可能具体且集中

possible. Your step should focus on the logical progression of the task

可能。你的步骤应侧重于任务的逻辑推进

instead of the actual low-level interactions, such as clicks or types.

而不是实际的低层交互，如点击或输入。

##Guidelines:

指南:

- Ensure every action and reasoning aligns with the user query, the webpage at

- 确保每个操作和推理都与用户查询及网页内容保持一致

hand, and the global plan, maintaining the strict order of actions.

手，以及全球计划，保持严格的行动顺序。

- Minimize the number of steps by clustering related actions into high-level,

- 通过将相关操作聚合为高级步骤，尽量减少步骤数量，

logical units. Each step should drive task completion and avoid unnecessary

逻辑单元。每一步都应推动任务完成，避免不必要的操作

granularity or redundancy. Focus on logical progression instead of

粒度或冗余。关注逻辑进展而非

detailing low-level interactions, such as clicks or UI-specific elements.

详细描述低层次的交互，如点击或特定于用户界面 (UI) 的元素。

- Provide clear, specific instructions for each step, ensuring the executor

- 为每一步提供清晰、具体的指示，确保执行者

has all the information needed without relying on assumed knowledge. For

拥有所有必要信息，不依赖于假设的知识。例如，

example, explicitly state, ’Input ’New York’ as the arrival city for the

明确说明“将‘纽约’输入为航班的到达城市”，而不是使用模糊的表达如“输入到达城市”。

flights,’ instead of vague phrases like ’Input the arrival city.’

- You can potentially output steps that include conditional statements in

- 你可以输出包含条件语句的步骤，使用自然语言表达，

natural language, such as ’If the search results exceed 100, refine the

例如“如果搜索结果超过 100 条，调整筛选条件以缩小选项范围”。但应避免过于复杂或

filters to narrow down the options.’ However, avoid overly complex or

含糊不清的指令，以免引起误解。

ambiguous instructions that could lead to misinterpretation.

##High-level Goals Guidelines:

高层目标指导原则:

- Focus on high-level goals rather than fine-grained web actions, while

- 关注高层目标，而非细粒度的网页操作，同时

maintaining specificity about what needs to be accomplished. Each step

保持对所需完成事项的具体描述。每一步应代表一个有意义的工作单元，可能包含多个

should represent a meaningful unit of work that may encompass multiple

低层操作 (点击、输入等)，这些操作服务于共同目的，但

low-level actions (clicks, types, etc.) that serve a common purpose, but

仍需明确预期结果。例如，不要将点击搜索框、输入查询和点击搜索分成多个步骤,

should still be precise about the intended outcome. For example, instead of

而应合并为一个高层次但具体的步骤。

having separate steps for clicking a search box, typing a query, and

clicking search, combine these into a single high-level but specific step

like ”Search for X product in the search box”.

如“在搜索框中搜索 X 产品”。

- Group related actions together that achieve a common sub-goal. Multiple

- 将实现共同子目标的相关操作分组。多个

actions that logically belong together should be combined into a single

逻辑上相关的操作应合并为一个

step. For example, multiple filter-related actions can be grouped into a

步骤。例如，多个与筛选相关的操作可以组合成一个

single step like ”Apply price range filters between $100-$200 and select

步骤，如“应用价格区间筛选 $100-$200 并选择

5-star rating”. The key is to identify actions that work together to

五星评级”。关键是识别协同完成特定目标的操作，

accomplish a specific objective while being explicit about the criteria and

并明确涉及的标准和

parameters involved.

参数。

- Focus on describing WHAT needs to be accomplished rather than HOW it will be

  - 重点描述需要完成的内容 (WHAT)，而非如何实现 (HOW)。

implemented. Your steps should clearly specify the intended outcome without

步骤应清晰说明预期结果，避免涉及界面交互的具体细节。

getting into the mechanics of UI interactions. The executor agent will

执行代理将

handle translating these high-level but precise steps into the necessary

负责将这些高层但精确的步骤转化为必要的

sequence of granular web actions.

细粒度网页操作序列。

##Initial HTML State Guidelines:

## 初始 HTML 状态指南:

- Use the initial HTML of the webpage as a reference to provide context for

  - 使用网页的初始 HTML 作为参考，为

your plan. Since this is just the initial HTML, possibly only a few of the

您的计划提供上下文。由于这只是初始 HTML，可能只有少数

initial actions are going to be taken on this state and the subsequent ones

将对该状态及其后续状态采取初步行动

are going to be taken on later states of the webpage; however, this initial

将会在网页的后续阶段进行处理；然而，这一初步阶段

HTML should help you ground the plan you are going to generate (both the

<html> 应该帮助你确定你将要生成的计划 (包括 </html>

reasoning behind individual steps and the overall plan) in the context of

在……背景下对各个步骤及整体方案的推理

the webpage at hand. This initial HTML should also help you ground the task

当前网页。这个初始 HTML 也应帮助你确定任务基础

description and the trajectory of actions in the context of the webpage,

网页背景下的描述和行为轨迹，

making it easier to understand the task.

使任务更易理解。

- You MUST provide an observation of the initial HTML state in your reasoning

- 你必须在推理过程中提供对初始 HTML 状态的观察

for the first step of your global plan, including the elements, their

作为您全球计划的第一步，包括元素及其

properties, and their possible interactions. Your observation should be

属性及其可能的相互作用。您的观察应当

detailed and provide a clear understanding of the current state of the HTML

详细且清晰地展示了 HTML 的当前状态

page.

页面。

##Formatting Guidelines:

## 格式指南:

- Start your response with the ’## Step 1’ header and follow the format

- 以“## 第一步”标题开始你的回复，并遵循该格式

provided in the examples.

示例中提供的。

- Ensure that each step is clearly separated and labeled with the ’## Step N′

- 确保每一步都清晰分隔并标注为“## 第 N′ 步”

header, where N is the step number.

标题，其中 N 是步骤编号。

- Include the ’Reasoning’ and ’Step’ sections in each step.

- 在每个步骤中包含“推理”和“步骤”部分。

A.3.2. PLANNER USER MESSAGE

A.3.2. 规划者用户消息

##User Query

## 用户查询

{user_query}

{user_query}

##Initial HTML State

## 初始 HTML 状态

{initial_html_state}

{initial_html_state}

You MUST start with the ’## Step 1’ header and follow the format provided in

你必须以“## 第 1 步”标题开始，并遵循示例中提供的格式。

the examples.

示例。

A.4. Executor Prompt

A.4. 执行者提示

Executor follows the WebArena-Lite defined executor prompt where each user-assistant message pair represents an HTML-action round. The only addition we have is to the system prompt which describes what a plan is.

执行者遵循 WebArena-Lite 定义的执行者提示，其中每对用户-助手消息代表一次 HTML 操作回合。我们唯一的补充是在系统提示中描述计划的含义。

A.4.1. EXECUTOR SYSTEM PROMPT

A.4.1. 执行者系统提示

#Goal

目标

You are the Executor Agent, a powerful assistant can complete complex web

你是执行者代理，一个强大的助手，能够通过执行点击、输入、选择等网页操作完成复杂的网页导航任务。

navigation tasks by issuing web actions such as clicking, typing,

你将获得以下信息:

selecting, and more. You will be provided with the following information:

- **Task Instruction**: The web task that you are required to complete.

  - ** 任务说明 **: 您需要完成的网页任务。

- **Global Plan**: A high-level plan that guides you to complete the web tasks.

  - ** 总体计划 **: 指导您完成网页任务的高层次计划。

- **Previous action trajectory**: A sequence of previous actions that you have

  - ** 先前操作轨迹 **: 您之前执行的一系列操作。

taken in the past rounds.

在过去的回合中采取的。

- **Current HTML**: The current HTML of the web page.

  - ** 当前 HTML**: 网页的当前 HTML 代码。

Your goal is to use the Global Plan, the previous action trajectory, and the

您的目标是利用总体计划、先前操作轨迹和

current observation to output the next immediate action to take in order to

当前观察结果，输出下一步立即采取的操作，

progress toward completing the given task.

以推进完成指定任务。

#Task Instruction: {intent}

# 任务说明:{intent}

#Global Plan

# 总体计划

The Global Plan is a structured, step-by-step plan that provides you with a

总体计划是一个结构化的、逐步的计划，为您提供

roadmap to complete the web task. Each step in the Global Plan (denoted as

完成网页任务的路线图。总体计划中的每一步 (表示为

'## Step X' where X is the step number) contains a reasoning and a

'## 第 X' 步'，其中 X 是步骤编号) 包含推理和

high-level action that you need to take. Since this Global Plan

您需要采取的高层次操作。由于该总体计划

encapsulates the entire task flow, you should identify where you are in the

涵盖了整个任务流程，您应通过参考先前操作轨迹和当前

plan by referring to the previous action trajectory and the current

状态来确定自己在计划中的位置。

observation, and then decide on the next action to take. Here is the Global

观察，然后决定下一步行动。以下是您的任务的全局计划:

Plan for the your task:

您的任务的全局计划:

{global_plan}

{global_plan}

A.5. Plan Data Annotator Prompt

A.5. 计划数据标注器提示

Similar to the planner prompt, there is a system prompt that defines the goals of the plan annotator; and the user message provides the user query, the initial HTML state, and the action trajectory for which the plan annotator needs to generate a plan.

类似于规划器提示，这里有一个系统提示定义了计划标注器的目标；用户消息提供用户查询、初始 HTML 状态以及计划标注器需要生成计划的动作轨迹。

A.5.1. PLAN DATA ANNOTATOR SYSTEM PROMPT

A.5.1. 计划数据标注器系统提示

##Goal

目标

You are the Global Planner agent, an expert plan generator for web navigation

您是全局规划器代理，是网页导航任务的专家计划生成器

tasks. You will be provided with the following information:

。您将获得以下信息:

- **User Query**: The web task that you are required to generate a global plan

- ** 用户查询 **: 您需要为其生成全局计划的网页任务

for.

。

- **Initial HTML State**: The initial HTML state of the web page.

- ** 初始 HTML 状态 **: 网页的初始 HTML 状态。

- **Trajectory**: A sequence of actions that represent a trajectory of a web

- ** 轨迹 **: 表示网页导航任务轨迹的一系列动作。

navigation task. It is formatted as series of actions where each action

它的格式是一系列动作，每个动作

first has a comment ('#') that describes the element to be interacted with

首先有一个注释 ('#')，描述要交互的元素

or a note what provides some context about the action and the current task

或提供有关动作和当前任务的一些上下文说明

state. The action is then described with the do function, which takes two

状态。然后使用 do 函数描述该动作，该函数接受两个

arguments: the action to be performed, the element to be interacted with,

参数: 要执行的动作、要交互的元素,

and sometimes an argument. The actions are numbered sequentially to

有时还包括一个参数。动作按顺序编号,

indicate the order in which they should be executed.

以指示它们应执行的顺序。

You are responsible for analyzing initial HTML state and the trajectory

您负责分析初始 HTML 状态和下面提供的轨迹,

provided below and producing a structured, step-by-step global plan that

并生成一个结构化的、逐步的全局计划,

clusters multiple actions into the fewest number of logical steps possible.

将多个动作聚合为尽可能少的逻辑步骤。

The global plan that you generate shouldn’t describe fine-grained web

您生成的全局计划不应描述细粒度的网页

interactions such as clicks or types but outline the high-level steps that

交互，如点击或输入，而应概述高层步骤,

encapsulate one or more actions in the trajectory, meaning each step in

这些步骤封装轨迹中的一个或多个动作，意味着您计划中的每一步

your plan will potentially require multiple actions to be completed. You

可能需要完成多个动作。您还需将轨迹中的每个动作

will also be tasked to classify each action in the trajectory with one of

分类到全局计划中的某一步。您的每一步

the steps in your global plan. Each of your steps will be handed to another

将交由另一个执行代理，该代理会将您的步骤转换为细粒度的网页

executor agent that will convert your step into fine-grained web

交互；因此，您的步骤应包含完成任务所需的所有具体信息，

interactions; hence, your steps should include every specific information

而不假设执行代理已具备

needed for completing the task without assuming the executor agent has

任何额外知识。

access to the whole task or trajectory.

访问整个任务或轨迹。

##Expected Output Format

## 预期输出格式

The global plan you generate should be structured in a numbered list format,

您生成的全球计划应以编号列表格式结构化，

starting with ’## Step 1’ and incrementing the step number for each

从“## 第 1 步”开始，每一步的编号依次递增

subsequent step. Each step in the plan should be in this exact format:

后续步骤。计划中的每一步都应采用此确切格式：

...
##Step N

## 步骤 N

Reasoning: [Your reasoning here]

推理:[您的推理内容]

Description: [Description of the actions this step covers]

描述:[本步骤涵盖的操作描述]

Step: [Your step here]

步骤:[在此填写您的步骤]

Actions: [list of action indexes associated with this step]

操作:[与此步骤相关的操作索引列表]

...

Here is a breakdown of the components you need to include in each step of your

以下是您在每个步骤中需要包含的组件细目

global plan as well as their specific instructions:

全球计划及其具体指示:

- **Reasoning**: In this section, you should explain your reasoning and

- ** 推理 **: 在本节中，您应解释您的推理过程并

thought process behind the step you are proposing. It should provide a

你所提出步骤背后的思考过程。它应当提供一个

high-level justification for why the actions in this step are grouped

对本步骤中各操作分组的高级理由说明

thogether and how they contribute to achieving the overall goal. Your

他们如何协同工作以及如何共同实现整体目标。您的

reasoning should be based on the information available in the trajectory

推理应基于轨迹中可获得的信息

(and potentially on the initial HTML state) and should guide the executor

(and potentially on the initial HTML state) 并可能基于初始 HTML 状态，应指导执行器

agent in understanding the strategic decision-making process behind your

帮助理解您背后的战略决策过程的代理

global plan.

全球计划。

- **Description**: This section should include a brief description of the

- ** 描述 **: 本节应包含对

actions that are grouped together in this step. You should exactly copy the

在此步骤中被归为一组的操作。你应当准确复制

action descriptions from the trajectory without any modifications or

从轨迹中提取的动作描述，未作任何修改或

additional information. This is to ensure that the executor agent can

附加信息。这样可以确保执行代理能够

accurately map the actions to the global plan steps. Specifically, every

准确地将动作映射到全局计划步骤。具体来说，每一个

action that you include in your description should include any ’# Element’,

您在描述中包含的操作应包括任何“# 元素”，

’# Note’, or ’# Exit’ comments that are present in the trajectory as well

轨迹中也存在的“# Note”或“# Exit”注释

as their corresponding ’do’ functions.

作为它们对应的“do”函数。

- **Step**: In this section, you should provide a concise description of the

  - ** 步骤 **: 在本节中，您应简要描述

global step being undertaken. Your step should summarize one or more

正在进行的全局步骤。您的步骤应总结一个或多个

actions from the trajectory as a logical unit. It should be as specific and

将轨迹中的动作作为一个逻辑单元。它应当尽可能具体且

concentrated as possible, without referring to any HTML or UI elements.

尽可能集中，不涉及任何 HTML 或用户界面元素。

Your step should focus on the logical progression of the task instead of

你的步骤应侧重于任务的逻辑进展，而非

the actual fine-grained interactions, such as clicks or types.

实际的细粒度交互，如点击或输入。

- ⋆⋆ Actions**: This section should list the indexes of the actions associated

- ⋆⋆ 操作 **: 本节应列出相关操作的索引

with this step. One or more actions should be grouped under one broader

通过这一步骤。一个或多个操作应归纳在一个更广泛的类别下

logical step. The indices in this section should exactly match the indices

逻辑步骤。本节中的索引应与索引完全匹配

of the actions in the trajectory.

轨迹中的动作。

##Examples

示例

Here are some examples of the expected output format for the global plan where

以下是全球计划预期输出格式的一些示例，其中

the input is the task description and the trajectory of actions taken to

输入是任务描述和所采取动作的轨迹

complete the task and the output is the structured global plan that

完成任务，输出的是结构化的全球计划

clusters multiple actions into the fewest number of logical steps possible

将多个操作聚合为尽可能少的逻辑步骤

without sacrificing specificity:

在不牺牲特异性的前提下:

{in_context_examples}

{in_context_examples}

##Planning Guidelines:

## 规划指南:

- Ensure every action and thought aligns with the trajectory and global plan,

- 确保每一个行动和想法都与轨迹和全球计划保持一致,

maintaining the strict order of actions. Actions should be sequential, with

保持严格的操作顺序。操作应按顺序进行，伴随

no skipping or misalignment (e.g., avoid assigning non-consecutive actions

不跳跃或错位 (例如，避免分配非连续的动作)

like Step 1: [0,3,4], Step 2: [1,2]). Deviation from the trajectory's order

如步骤 1:[0,3,4]，步骤 2:[1,2]。偏离轨迹顺序

will be PENALIZED!

将被处罚!

- Minimize the number of steps by clustering related actions into high-level,

- 通过将相关操作聚合为高级步骤，尽量减少步骤数量,

logical units. Each step should drive task completion and avoid unnecessary

逻辑单元。每一步都应推动任务完成，避免不必要的

granularity or redundancy. Focus on logical progression instead of

粒度或冗余。关注逻辑进展而非

detailing fine-grained interactions, such as clicks or UI-specific elements.

详细描述细粒度的交互，如点击或特定于用户界面 (UI) 的元素。

- Provide clear, specific instructions for each step, ensuring the executor

- 为每个步骤提供清晰、具体的指示，确保执行者

has all the information needed without relying on assumed knowledge. For

拥有所有所需信息，无需依赖假设的知识。

example, explicitly state, ’Input ’New York’ as the arrival city for the

例如，明确说明，将“New York”作为到达城市输入

flights,’ instead of vague phrases like ’Input the arrival city.’

使用“航班”而不是模糊的短语如“输入到达城市”。

- You can potentially output steps that include conditional statements in

- 您可以输出包含条件语句的步骤

natural language, such as ’If the search results exceed 100, refine the

自然语言，例如“如果搜索结果超过 100 条，请缩小范围

filters to narrow down the options.’ However, avoid overly complex or

筛选以缩小选项范围。

ambiguous instructions that could lead to misinterpretation.

可能导致误解的模糊指令。

##High-level Goals Guidelines:

## 高级目标指南:

- Focus on high-level goals rather than fine-grained web actions, while

- 关注高层次目标而非细粒度的网页操作，同时

maintaining specificity about what needs to be accomplished. Each step

保持对需要完成事项的具体明确。每一步

should represent a meaningful unit of work that may encompass multiple

应代表一个有意义的工作单元，可能包含多个部分

low-level actions (clicks, types, etc.) that serve a common purpose, but

低级操作 (点击、输入等) 具有共同目的，但

should still be precise about the intended outcome. For example, instead of

仍应明确预期的结果。例如，不要说

having separate steps for clicking a search box, typing a query, and

将点击搜索框、输入查询和

clicking search, combine these into a single high-level but specific step

点击搜索，将这些合并为一个高层次但具体的步骤

like ”Search for X product”.

如“搜索 X 产品”。

- Group related actions together that achieve a common sub-goal. Multiple

- 将实现共同子目标的相关操作组合在一起。多个

actions that logically belong together should be combined into a single

逻辑上相关的操作应合并为一个

step. For example, multiple filter-related actions can be grouped into a

步骤。例如，多个与过滤器相关的操作可以被归类为一个

single step like ”Apply price range filters between $100-$200 and select

单步操作，如“应用价格范围过滤器，介于 100 美元至 200 美元之间，并选择

5-star rating”. The key is to identify actions that work together to

“五星评级”。关键是识别协同作用的行为以

accomplish a specific objective while being explicit about the criteria and

明确标准并完成特定目标

parameters involved.

涉及的参数。

- Focus on describing WHAT needs to be accomplished rather than HOW it will be

- 重点描述需要完成的内容，而非如何完成

implemented. Your steps should clearly specify the intended outcome without

已实现。您的步骤应明确说明预期结果，而不应

getting into the mechanics of UI interactions. The executor agent will

深入探讨用户界面交互的机制。执行代理将

handle translating these high-level but precise steps into the necessary

处理将这些高级但精确的步骤转化为必要的

sequence of granular web actions.

颗粒化网页操作序列。

- Provide clear, specific instructions for each step, ensuring the executor

- 为每一步提供清晰、具体的指示，确保执行者

has all the information needed without relying on assumed knowledge. For

拥有所有所需信息，无需依赖假设的知识。

example, explicitly state, ’Input ’New York’ as the arrival city for the

例如，明确说明，“输入‘纽约’作为到达城市

flights,’ instead of vague phrases like ’Input the arrival city.’

使用“航班”，而不是模糊的短语如“输入到达城市”。

- The action trajectory might include several ”scroll down” actions necessary

- 操作轨迹可能包括若干必要的“向下滚动”动作

to locate or find an element, but you should not explicitly say ”scroll

定位或查找元素，但不应明确说“滚动”

down to find X'' in your step description. Instead, you can use phrases like

在你的步骤描述中查找 X'' 。相反，你可以使用诸如

”locate X”, ”find Y”, ”look for Z”, or similar phrases to represent the

“定位 X”、“查找 Y”、“寻找 Z”或类似短语来表示

scroll actions in your step description. The act of scrolling is not part

在步骤描述中滚动操作。滚动行为不属于

of the high-level goal but just implementation details, so you should not

是高级目标而仅仅是实现细节，因此你不应该

explicitly mention it in your step description.

在您的步骤描述中明确提及它。

- Example:

- 示例:

BAD plan (mentions scrolling):

糟糕的计划 (提到滚动):

Step 1: Scroll down to find the ’Contact Us’ button and click it

步骤 1: 向下滚动找到“联系我们”按钮并点击

Step 2: Scroll through the list to find the order numbered ID12345

步骤 2: 滚动列表，找到编号为 ID12345 的订单

...

GOOD plan (avoids mentioning scrolling):

良好方案 (避免提及滚动):

...

Step 1: Locate the ’Contact Us’ button and click it

步骤 1: 找到“联系我们”按钮并点击

Step 2: Find the order numbered ID12345

步骤 2: 找到编号为 ID12345 的订单

...

##Initial HTML State Guidelines:

## 初始 HTML 状态指南:

- Use the initial HTML of the webpage as a reference to provide context for

  - 使用网页的初始 HTML 作为参考，以提供上下文

  your plan. Since this is just the initial HTML, possibly only a few of the

  你的计划。由于这只是初始的 HTML，可能只有少数

  initial actions are going to be taken on this state and the subsequent ones

  将对该状态及后续状态采取初步行动

  are going to be taken on later states of the webpage; however, this initial

  将在网页的后续阶段进行处理；然而，这一初步

  HTML should help you ground the plan you are going to generate (both the

  HTML 应该帮助你确定你将要生成的计划 (包括

  reasoning behind individual steps and the overall plan) in the context of

  (各步骤及整体方案的推理依据) 在……的背景下

  the webpage at hand. This initial HTML should also help you ground the task

  当前网页。这个初始的 HTML 也应帮助你理解任务

  description and the trajectory of actions in the context of the webpage,

  网页环境中的描述和行为轨迹,

  making it easier to understand the task.

  使任务更易于理解。

- You MUST provide an observation of the initial HTML state in your reasoning

  - 你必须在推理中提供对初始 HTML 状态的观察

  for the first step of your global plan, including the elements, their

  作为你整体计划的第一步，包括元素、它们的

  properties, and their possible interactions. Your observation should be

  属性及其可能的交互。你的观察应当

detailed and provide a clear understanding of the current state of the HTML

详尽且清晰地展示当前 HTML 页面的状态。

page. Please refer to the examples for more information on how to do this.

请参照示例以获取更多如何操作的信息。

##Formatting Guidelines:

## 格式指南:

- Start your response with the ’## Step 1’ header and follow the format

- 以“## 第 1 步”标题开始你的回答，并遵循示例中提供的格式

provided in the examples.

。

- Ensure that each step is clearly separated and labeled with the ’## Step N’

- 确保每一步都清晰分隔，并以“## 第 N 步”标题标注，

header, where N is the step number.

其中 N 为步骤编号。

- Include the ’Reasoning’, ’Actions that this step covers’, ’Indices of

- 每步应包含“推理”、“该步骤涵盖的操作”、“操作索引”和“步骤”部分。

actions’, and ’Step’ sections in each step.

A.5.2. PLAN DATA ANNOTATOR USER MESSAGE

A.5.2. 计划数据标注者用户消息

##User Query

## 用户查询

{goal_description}

```
{goal_description}
```

##Initial HTML State

```
## 初始 HTML 状态
```

{initial_html_state}

```
{initial_html_state}
```

##Trajectory

```
## 轨迹
```

The following is a sequence of actions that represent a trajectory of a web

```
以下是一系列动作，表示一个网页导航任务的轨迹
```

navigation task. It is formatted as series of actions where each action

```
  它被格式化为一系列动作，每个动作
```

first has a comment ('#') that describes the element to be interacted with

```
  首先有一个注释 ('#')，描述要交互的元素
```

or a note what provides some context about the action and the current task

```
  或提供关于动作和当前任务状态的一些上下文说明
```

state. The action is then described with the do function, which takes two

```
  然后用 do 函数描述该动作，该函数接受两个
```

arguments: the action to be performed, the element to be interacted with,

```
  参数: 要执行的动作、要交互的元素,
```

and sometimes an argument. The actions are numbered sequentially to

```
  有时还包括一个参数。动作按顺序编号,
```

indicate the order in which they should be executed:

```
  以指示执行的顺序:
```

{trajectory}

{trajectory}

A.6. Synthetic Plan Generator Prompt

A.6. 合成计划生成器提示

Similarly, the synthetic plan generator also has a system prompt that presents the goals of the synthetic plan generator and also provides the seed data examples. The user message specifies how many synthetic plans to generate (from the seed data examples in the system prompt).

同样，合成计划生成器也有一个系统提示，展示合成计划生成器的目标，并提供种子数据示例。用户消息指定要生成多少个合成计划 (基于系统提示中的种子数据示例)。

A.6.1. SYNTHETIC PLAN GENERATOR SYSTEM PROMPT

A.6.1. 合成计划生成器系统提示

#Goal

目标

You are a Plan Data Generator that can generate new synthetic data to train a

你是一个计划数据生成器，能够生成新的合成数据用于训练

planner language model to be excellent at plan generation for web

规划语言模型，旨在出色地生成网页导航任务的计划

navigation tasks. The data that this model is going to trained (and hence

该模型将要训练的数据 (因此也是你生成的数据) 将采用以下格式:

the data you generate) is going to be in the following format:

数据格式如下:

- **Input**: A user query for a web navigation task.

  - ** 输入 **: 用户针对网页导航任务的查询。

- **Output**: A high-level global plan to accomplish the task.

- ** 输出 **: 完成该任务的高级全局计划。

You will be given some examples on how the input-output pairs look like and

你将获得一些输入-输出对的示例,

your goal is to generate new data pairs that are similar to the examples

你的目标是生成与示例相似的新数据对,

given. Your goal is to increase the data diversity by covering a wide

通过涵盖广泛的可能用户查询来增加数据多样性,

range of possible user queries while also grounding your data generation

同时将数据生成过程基于示例所依托的特定网站。

process on the specific website that the examples are based on. You

你不应仅仅复制示例，因为那样无法帮助模型更好地泛化,

shouldn’t just copy the examples since that would not help the model

但也不应生成网站上不可能实现的数据。

generalize better but you also shouldn’t generate data that is not

你必须利用给定的示例推断网站上的可能性,

possible on the website. You must use the given examples to infer what is

并以此为基础生成数据。

possible on the website and ground your generated data on it.

#Expected Output Format

# 预期输出格式

The input-output pairs you generate should be structured as follows:

你生成的输入-输出对应遵循以下结构:

...

##Data Pair {{i}}

## 数据对 {{i}}

User Query:

用户查询:

<user query>
Initial HTML State:

初始 HTML 状态:

<index of the example whose initial HTML state you are starting from>
Global Plan:

总体计划:

<global plan>
where:

其中:

- ‘{{i}}‘ is the data pair number.

  - ‘{{i}}‘ 是数据对编号。

- ‘<user query>‘ is a brief description of the task that the user wants to

  - ‘<user query>‘ 是用户希望在网站上完成的任务的简要描述。

  accomplish on a website.

- ‘<index of the example whose initial HTML state you are starting from>‘ is

  - ‘<index of the example whose initial HTML state you are starting from>‘ 是

  the index of the example whose initial HTML state you are starting from.

  你开始时所用示例的初始 HTML 状态的索引。

  This is just an integer like 1, 3, etc.

  这只是一个整数，如 1、3 等。

- ‘<global plan>‘ is a high-level global plan that outlines the steps needed

- ‘<global plan>‘ 是一个高层次的总体计划，概述完成任务所需的步骤。

to accomplish the task.

#Instructions

说明

Here are the guidelines to follow when generating the data:

以下是生成数据时应遵循的指导原则:

##User Query Instructions

## 用户查询说明

The User Query is a brief description of the task that the user wants to

用户查询是用户想要完成任务的简要描述

accomplish on a website. It should be concise and focused on the main goal

在网站上完成。应简明扼要，聚焦主要目标

of the task. The user query should provide enough context for an agent to

任务的。用户查询应提供足够的上下文以便代理能够

generate a high-level global plan to accomplish the task.

生成一个高层次的全球计划以完成任务。

##Initial HTML State Instructions

## 初始 HTML 状态指令

- The Initial HTML State is the HTML representation of the webpage at the

- 初始 HTML 状态是网页在

beginning of the task. It provides the context for the user query and the

任务开始。它为用户查询提供了背景信息，并且

global plan. When generating new data, you should choose the initial HTML

全局计划。生成新数据时，应选择初始 HTML

state of one of the examples that you want to start from and provide the

从您想要开始的某个示例的状态出发，并提供

index of the example whose initial HTML state you are starting from. This

你开始时所用的示例的初始 HTML 状态的索引。

will ensure that the generated data is grounded in the context of the

将确保生成的数据基于……的上下文

specific website and HTMLs that the examples are based on. You should only

示例所基于的特定网站和 HTML。您只应

provide the index of the example whose initial HTML state you are starting

提供您开始时的示例初始 HTML 状态的索引

from. For example, if you are starting from the second example's initial

来自。例如，如果你从第二个例子的初始状态开始

HTML state ('# Example 2'), you should provide '2' as the initial HTML

HTML 状态 ('# Example 2')，你应提供'2' 作为初始 HTML

state.

状态。

- When generating multiple data pairs, you should aim to use different

- 在生成多个数据对时，应尽量使用不同的

examples' initial HTML states in a balanced way. While you don't need to

以平衡的方式展示示例的初始 HTML 状态。虽然你不需要

use each HTML state exactly equally, you should ensure good coverage across

要均匀使用每个 HTML 状态，应确保覆盖面良好

all examples. Some HTML states may enable a wider range of user queries and

所有示例。一些 HTML 状态可能支持更广泛的用户查询和

can be used more frequently, but you shouldn’t completely ignore or heavily

可以更频繁地使用，但你不应完全忽视或过度

underutilize any of the examples. The goal is to leverage the full range of

不要低估任何示例的价值。目标是充分利用全部范围的

possible HTML states and website functionalities shown in the examples.

示例中展示的可能的 HTML 状态和网站功能。

- Aftering picking which HTML to start from, you MUST provide an observation

- 选择起始的 HTML 后，您必须提供一个观察结果

of the initial HTML state in your reasoning for the first step of your

在你推理的第一步中考虑初始 HTML 状态

global plan, including the elements, their properties, and their possible

全局计划，包括元素、它们的属性及其可能性

interactions. Your observation should be detailed and provide a clear

交互。您的观察应详尽且清晰地呈现

understanding of the current state of the HTML page. Please refer to the

对当前 HTML 页面状态的理解。请参阅

examples for more information on how to do this.

有关如何操作的更多信息示例。

##Global Plan Instructions

## 全球计划说明

The Global Plan is a structured, step-by-step plan that provides a high-level

全球计划是一个结构化的、逐步推进的计划，提供了一个高层次的

overview of the actions that need to be taken to accomplish a web

完成一个网站所需采取的操作概述

navigation task. The plan should be detailed enough to guide the user

导航任务。计划应足够详细以指导用户

through the task but not too detailed that it becomes a step-by-step

完成任务，但又不能过于详尽以至于变成逐步操作说明

instruction. In other words, the global plan that you generate shouldn’t

换言之，你生成的全局计划不应

directly describe low-level web actions such as clicks or types (unless

直接描述低级网页操作如点击或输入 (除非

necessary for clarity) but outline the high-level steps that encapsulate

为清晰起见必须)，而应概述封装

one or more actions in the action trajectory, meaning each step in your

一个或多个动作的高层步骤，意味着计划中的每一步

plan will potentially require multiple actions to be completed. Your global

可能需要完成多个动作。你的全局

plan will then be handed to an Executor agent which will perform low-level

计划随后将交给执行代理，该代理将在网页上执行低级

web actions on the webpage (click, type, hover, and more) to convert your

操作 (点击、输入、悬停等)，将你的

global plan into a sequence of actions and complete the user query.

全局计划转化为一系列动作并完成用户查询。

###Global Plan Expected Output Format

### 全局计划预期输出格式

The global plan you generate should be structured in a numbered list format,

你生成的全局计划应以编号列表格式组织,

starting with ’## Step 1’ and incrementing the step number for each

从“## Step 1”开始，后续步骤依次递增编号。

subsequent step. Each step in the plan should be in this exact format:

计划中的每一步应采用以下准确格式:

...
##Step N

##Step N

Reasoning: [Your reasoning here]

推理:[在此填写你的推理]

Step: [Your step here]

步骤:[在此填写您的步骤]

...
Here is a breakdown of the components you need to include in each step of your

以下是您在每个步骤中需要包含的组成部分及其具体说明:

global plan as well as their specific instructions:

全局计划以及它们的具体指示:

- ⋆⋆ Reasoning**: In this section, you should explain your reasoning and

- ⋆⋆ 推理 **: 在本节中，您应解释所提步骤背后的推理和思考过程。

thought process behind the step you are proposing. It should provide a

应提供对该步骤中各项行动为何被归为一组的高层次理由,

high-level justification for why the actions in this step are grouped

以及它们如何助力实现整体目标。

together and how they contribute to achieving the overall goal. Your

您的推理应基于用户查询中提供的信息 (以及可能的初始 HTML 状态),

reasoning should be based on the information available in the user query

并指导执行代理理解您全局计划背后的战略决策过程。

(and potentially on the initial HTML state) and should guide the Executor

agent in understanding the strategic decision-making process behind your

global plan.

- ⋆⋆ Step**: In this section, you should provide a concise description of the

- ⋆⋆ 步骤 **: 在本节中，您应简明描述所执行的全局步骤。

global step being undertaken. Your step should summarize one or more

您的步骤应将一个或多个动作总结为一个逻辑单元,

actions as a logical unit. It should be as specific and concentrated as

尽可能具体且集中。

possible. Your step should focus on the logical progression of the task

步骤应侧重于任务的逻辑进展，而非具体的低层交互，如点击或输入。

instead of the actual low-level interactions, such as clicks or types.

##High-level Goals Guidelines:

## 高级目标指南:

- Focus on high-level goals rather than fine-grained web actions, while

- 关注高层次目标而非细粒度的网页操作，同时

maintaining specificity about what needs to be accomplished. Each step

保持对需要完成事项的具体明确。每一步

---

should represent a meaningful unit of work that may encompass multiple

应代表一个有意义的工作单元，可能包含多个部分

low-level actions (clicks, types, etc.) that serve a common purpose, but

低级操作 (点击、输入等) 具有共同目的，但

should still be precise about the intended outcome. For example, instead of

仍应明确预期的结果。例如，不要说

having separate steps for clicking a search box, typing a query, and

将点击搜索框、输入查询内容分为独立步骤，且

clicking search, combine these into a single high-level but specific step

点击搜索，将这些合并为一个高级但具体的步骤

like ”Search for X product”.

比如“搜索 X 产品”。

- Group related actions together that achieve a common sub-goal. Multiple

- 将实现共同子目标的相关操作组合在一起。多个

actions that logically belong together should be combined into a single

逻辑上相关的操作应合并为一个

step. For example, multiple filter-related actions can be grouped into a

步骤。例如，多个与过滤器相关的操作可以被归类为一个

single step like ”Apply price range filters between $100-$200 and select

单步操作，如“应用价格区间过滤器，范围在 100 美元至 200 美元之间，并选择

5-star rating”. The key is to identify actions that work together to

“五星评价”。关键是识别协同作用的行为

accomplish a specific objective while being explicit about the criteria and

完成特定目标，同时明确标准和

parameters involved.

涉及的参数。

- Focus on describing WHAT needs to be accomplished rather than HOW it will be

- 重点描述需要完成的内容，而非如何完成

implemented. Your steps should clearly specify the intended outcome without

已实现。您的步骤应明确说明预期结果，而不应

getting into the mechanics of UI interactions. Another executor agent will

深入探讨用户界面交互的机制。另一个执行代理将

handle translating these high-level but precise steps into the necessary

处理将这些高级但精确的步骤转化为必要的内容

sequence of granular web actions.

颗粒化网页操作序列。

#Examples

示例

Here are some examples you must utilize to understand what is possible on the

以下是一些示例，您必须利用它们来了解可能实现的内容

website, what kind of actions are executable, what HTML elements are

网站，哪些操作可执行，HTML 元素是什么

present on the website, and what kind of tasks you can generate data for.

网站上展示的内容，以及您可以为哪些类型的任务生成数据。

Remember:

请记住:

1. You are required to take inspiration from these example but not exactly

1. 你需要从这些例子中获得灵感，但不要完全照搬

copy them since we want enough diversity to be able to cover a wide variety

复制它们，因为我们需要足够的多样性以覆盖各种情况

of use cases.

的用例。

2. You shouldn’t hallucinate or create non-existing elements or actions that

2. 你不应产生幻觉或虚构不存在的元素或行为，

are not possible on the website. If you make up something that is not

在网站上是不可能的。如果你编造一些不存在的内容

possible on the website, you will be penalized. Your data needs to be

在网站上可能的情况下，您将受到处罚。您的数据需要被

grounded on the website and the examples given.

基于网站和所给示例。

{examples_str}

{examples_str}

A.6.2. SYNTHETIC PLAN GENERATOR USER MESSAGE

A.6.2. 合成计划生成器用户消息

Use the given examples to generate {how_many_to_generate_at_once} new data

使用给定的示例生成 {how_many_to_generate_at_once} 条新数据

pairs. The data pairs you generate SHOULDN’ T be similar to each other. They

对。你生成的数据对不应彼此相似。它们

should be diverse and cover a wide range of possible user queries and tasks.

应多样化，涵盖广泛的可能用户查询和任务。

#Output Formatting

# 输出格式

You should output the data pairs you generate in the following format:

你应按以下格式输出生成的数据对:

...
##Data Pair {i}

## 数据对 {i}

User Query:

用户查询:

<user query>
Initial HTML State:

初始 HTML 状态:

<index of the example whose initial HTML state you are starting from. Remember

< 你开始的示例的初始 HTML 状态的索引。请记住

this is just an integer like 1, 3 etc.>

这只是一个整数，如 1、3 等。>

Global Plan:

全局计划:

<global plan>
...
#Remember

# 请记住

- You shouldn’t hallucinate or create non-existing elements or actions that

- 你不应虚构或创建不存在的元素或操作，

are not possible on the website. If you make up something that is not

网站上不可能实现。如果你编造了网站上不可能实现的内容，

possible on the website, you will be penalized. Your data needs to be

你将受到惩罚。你的数据需要基于网站和给出的示例。

grounded on the website and the examples given.

- You are required to take inspiration from these examples but not exactly

- 你需要从这些示例中获得灵感，但不能完全复制，

copy them since we want enough diversity to be able to cover a wide variety

因为我们希望有足够的多样性以覆盖各种使用场景。

---

of use cases. However, while trying to create diverse data, you MUST avoid

然而，在尝试创建多样化数据时，必须避免

making up non-existing elements or actions that are not possible on the

编造不存在的元素或网站上不可能的操作。

website.

- You MUST provide a detailed initial HTML state observation for the first

- 你必须为全局计划的第一步提供详细的初始 HTML 状态观察。

step of your global plan.

A.7. Training Data Failure Classification Prompt

A.7. 训练数据失败分类提示

For classifying the training data based on the failure classes we identified, we have the main system prompt which defines the goal of the classification model. Each website has its own failure classes. If the model classifies a training data point into any one of these classes, we keep that data point for the next round of synthetic data generation.

为了根据我们识别的失败类别对训练数据进行分类，我们有一个主要系统提示，定义了分类模型的目标。每个网站都有其特定的失败类别。如果模型将某个训练数据点分类到这些类别中的任何一个，我们将保留该数据点用于下一轮合成数据生成。

A.7.1. Main System Prompt

A.7.1. 主要系统提示

##Goal

目标

You are an expert classifier model tasked with classifying data points that

你是一个专家分类模型，负责对用于训练“Planner”(规划器) 模型的数据点进行分类。

were used to train a ”Planner” model. This model was trained to take in a

该模型被训练用于接收

user query (or a task) related to common websites such as shopping

与购物等常见网站相关的用户查询 (或任务)

websites, Reddit, GitLab, etc., and output a high-level global plan for

网站、Reddit、GitLab 等，并输出一个高级的全球计划

completing that task. After training, we conducted a failure analysis to

完成该任务后，我们进行了故障分析以

identify the types of errors the planner was most prone to.

识别规划者最容易出现的错误类型。

Now, using the identified failure classes, we aim to label the training points

现在，利用已识别的故障类别，我们旨在标注训练点

of the global planner. The purpose of this classification is to determine

全局规划器的目的在于确定

which data points can be leveraged to generate synthetic data. This

可以利用哪些数据点来生成合成数据。

synthetic data will be used to retrain the planner, helping it correct its

合成数据将用于重新训练规划器，帮助其纠正

mistakes and avoid previous failures.

错误并避免以往的失败。

For each data point, you will receive:

对于每个数据点，您将收到:

- The website name: e.g., ”shopping_admin”

- 网站名称: 例如，“shopping_admin”

- A user query (task): The user query or task that the planner is supposed to

- 用户查询 (任务): 规划者应处理的用户查询或任务

complete

完成

- A ground truth global plan: The global planner was trained to generate this

- 一个真实的全局规划: 全局规划器经过训练以生成此规划

plan for the given user query.

为给定的用户查询制定计划。

Remember: The data points that will be given to you are going to be perfect

请记住: 将提供给您的数据点将是完美的

(they are from the training data): They are going to be the best possible

(它们来自训练数据): 它们将是最优的

plans that the planner can generate. Hence, your job is not to classify the

规划者能够生成的计划。因此，你的任务不是去分类

data point itself into a failure class but rather identify whether this

数据点本身并不归入失败类别，而是识别该

data point is a good example to train the planner to generate better plans

数据点是训练规划器生成更优计划的良好示例

and which failure class it will potentially help the planner avoid.

以及它可能帮助规划者避免的故障类别。

Your job:

你的工作:

1) Read the given user query and the plan carefully

1) 仔细阅读给定的用户查询和计划

2) Identify what this data points is trying to do and what can the planner

2) 确定这些数据点的目的以及规划者能做什么

model learn from being trained on this data point and data points like it

模型通过对该数据点及类似数据点的训练进行学习

3) Provide clear reasoning for your classification decision

3) 提供明确的分类决策理由

4) Classify the data point into one of the known failure classes for that

4) 将该数据点分类到已知的故障类别之一

website or ”Other” if no class fits; specifically, you should classify the

网站，或如果没有合适的类别则归为“其他”；具体来说，您应对其进行分类

failure class that this data point will help the planner avoid if it was

如果发生，该数据点将帮助规划者避免的故障类别

trained on this data point and data points like it

在该数据点及类似数据点上训练

Below is the set of possible classes for the website: {website.value}.

以下是该网站的可能类别集合:{website.value}。

{classification_section_for_website}

{classification_section_for_website}

General guidelines:

通用指南:

1. Carefully check the user query and plan

1. 仔细检查用户查询和计划

2. Match them against the class definitions

2. 将其与类别定义进行匹配

3. If none of the classes apply, label as ”Other”

3. 如果没有任何类别适用，则标记为“其他”

4. Provide your output in the following format:

4. 按以下格式提供输出:

##Reasoning

推理

[Explain your thought process and why this example fits the chosen class]

[解释你的思考过程以及为何该示例符合所选类别]

##Classification

分类

[Class label: ”Class A”, ”Class B”, ”Other”, etc.]

[类别标签:“类别 A”、“类别 B”、“其他”等]

Please ensure your output follows this exact format.

请确保你的输出严格遵循此格式。

Here are the prompts for the failure classification model for each website separately. Each failure class was identified by looking at our model’s performance on the validation set:

以下是针对每个网站的失败分类模型提示。每个失败类别均通过观察模型在验证集上的表现确定:

A.7.2. SHOPPING ADMIN (CMS) FAILURE CLASSES

A.7.2. 购物管理 (CMS) 失败类别

Here are the classes for the shopping_admin website:

以下是 shopping_admin 网站的类别:

#Shopping Admin Website Classes

# 购物管理网站类别

##Class A: Search Query Optimization Failures

## 类别 A: 搜索查询优化失败

###Description

描述

The planner fails to implement proper search query strategies, particularly:

规划器未能实施适当的搜索查询策略，具体表现为:

- Using overly specific search terms without fallback to broader terms

- 使用过于具体的搜索词，且未回退到更广泛的词汇

- Not utilizing the search functionality effectively when simpler queries

- 在简单查询可行时未有效利用搜索功能

would work

- Missing critical search parameters or using irrelevant ones

- 缺失关键搜索参数或使用了无关参数

###Training Data Needed

### 所需训练数据

- Examples showing fallback to broader search terms when specific searches fail

- 示例展示在具体搜索失败时回退到更广泛搜索词的情况

- Cases demonstrating effective use of search functionality with simpler

- 案例演示在简单查询下有效使用搜索功能

queries

###Example Tasks

### 示例任务

1. ”Show me the name of the customers who have expressed dissatisfaction with

1. “显示表达对 Chloe 水箱不满的客户姓名”

Chloe tank”

- Error: Planner used exact ”chloe tank” search instead of broader ”chloe”

- 错误: 规划器使用了精确的“chloe tank”搜索，而非更广泛的“chloe”

search that would have found ”chloe plastic tank”

搜索，本可找到“chloe 塑料水箱”

2. ”List the top 3 search terms in my store”

2. “列出我店铺中排名前三的搜索词”

- Error: Planner incorrectly included date filtering steps which don’t

- 错误: 规划器错误地包含了不必要的日期过滤步骤

exist in search terms report

存在于搜索词报告中

- Solution: Training data showing correct navigation of ”search terms”

- 解决方案: 训练数据展示正确导航“搜索词”的方法

report without date filtering

无日期过滤的报告

##Class B: Product Attribute Update Confusion

##B 类: 产品属性更新混淆

###Description

描述

The planner confuses high-level status changes with specific attribute updates:

规划者将高级状态变更与具体属性更新混淆:

- Using ”Change status” action instead of updating specific product attributes

  - 使用“更改状态”操作代替更新具体产品属性

- Attempting to modify stock/price/sale status through wrong interface elements

  - 试图通过错误的界面元素修改库存/价格/销售状态

###Training Data Needed

### 所需训练数据

- Examples showing correct attribute updates for sales status

  - 展示正确销售状态属性更新的示例

- Cases demonstrating proper stock level modifications

  - 演示正确库存水平修改的案例

- Examples distinguishing between status changes and attribute updates

- 区分状态变更与属性更新的示例

###Example Tasks

### 示例任务

1. ”Mark all Hollister shirts on sale”

1. “将所有 Hollister 衬衫标记为促销”

- Error: Planner used general status change instead of specific sale

- 错误: 规划者使用了通用状态变更而非具体的促销属性更新

attribute update

属性更新

- Solution: Training data showing how to update sale attributes

- - 解决方案: 展示如何更新销售属性的训练数据

specifically using ’update attributes’ option

特别使用“更新属性”选项

---

2. ”Make all Aeno capri as out of stock”

2. “将所有 Aeno 七分裤标记为缺货”

- Error: Planner tried using Enable/Disable status instead of stock

- - 错误: 计划者尝试使用启用/禁用状态而非库存属性

attribute

属性

- Solution: More examples of updating product attributes vs changing status

- - 解决方案: 更多关于更新产品属性与更改状态的示例

##Class C: Review Analysis Navigation Failures

##C 类: 评论分析导航失败

###Description

描述

The planner fails to properly navigate and analyze product reviews:

计划者未能正确导航和分析产品评论:

- Missing steps to access product review sections

  - 缺少访问产品评论部分的步骤

- Failing to specify review content examination steps

  - 未能明确评论内容检查步骤

- Not including steps to gather specific review details

  - 未包含收集具体评论细节的步骤

###Training Data Needed

### 所需训练数据

- Examples showing navigation to product review sections

  - 展示导航至产品评论部分的示例

- Cases demonstrating proper review content analysis

  - 演示正确评论内容分析的案例

- Examples of gathering specific review details

  - 收集具体评论细节的示例

###Example Tasks

### 示例任务

1. ”Tell me the reasons why customers like Circe’s products”

1. “告诉我客户喜欢 Circe 产品的原因”

   - Error: Planner didn’t include steps to access and analyze review content

- 错误: 规划者未包含访问和分析评论内容的步骤

- Solution: Training data showing how to navigate to and analyze review

- 解决方案: 展示如何导航至并分析评论的训练数据

sections

部分

##Other

其他

Description: If none of the above classes match.

描述: 如果以上类别均不匹配。

A.7.3. REDDIT FAILURE CLASSES PROMPT

A.7.3. REDDIT 失败类别提示

#Reddit Website Classes

#Reddit 网站类别

##Class A: Content Reposting Strategy Failures

## 类别 A: 内容转发策略失败

###Description

描述

The planner fails to implement correct reposting workflow:

规划者未能执行正确的转发工作流程:

- Missing steps to access repost functionality

  - 缺少访问转发功能的步骤

- Creating new posts instead of using repost features

  - 创建新帖子而非使用转发功能

- Incorrect navigation for cross-posting

  - 跨帖导航错误

###Training Data Needed

### 所需训练数据

- Examples showing proper repost functionality usage

  - 展示正确转发功能使用的示例

- Cases demonstrating cross-posting workflows

  - 演示跨帖工作流程的案例

###Example Tasks

### 示例任务

1. ”Re-post the image of costume contest to funny subreddit”

1. “将服装比赛的图片转发到搞笑子版块”

- Error: Planner created new post instead of using existing repost

- 错误: 规划者创建了新帖子，而不是使用已有的转发

functionality

功能

- Solution: Training data showing correct repost/crosspost workflow

- 解决方案: 展示正确转发/跨帖工作流程的训练数据

##Other

其他

Description: If none of the above classes match.

描述: 如果以上类别均不匹配。

A.7.4. GITLAB FAILURE CLASSES PROMPT

A.7.4. GITLAB 失败类别提示

#GitLab Website Classes

#GitLab 网站类别

##Class A: Issue/MR Navigation Strategy Failures

## 类别 A: 问题/合并请求导航策略失败

###Description

描述

The planner fails to use proper navigation paths for issues/merge requests:

规划者未能使用正确的问题/合并请求导航路径:

- Using global search instead of dedicated Issues/MR sections
  - 使用全局搜索而非专用的问题/合并请求版块
- Not utilizing proper filtering tabs (Open/Closed/All)
  - 未使用正确的筛选标签 (打开/关闭/全部)
- Missing steps to access personal issues/MRs through correct interface
  - 缺少通过正确界面访问个人问题/合并请求的步骤

###Training Data Needed

### 所需训练数据

- Examples showing navigation through Issues/MR tabs
  - 展示通过问题/合并请求标签导航的示例
- Cases demonstrating proper use of filtering options
  - 演示正确使用筛选选项的案例

- Examples of accessing personal issues/MRs

- 访问个人问题/合并请求的示例

###Example Tasks

### 示例任务

1. ”Open my latest created issue that has homepage content in its title”

1. “打开我最新创建的标题中包含主页内容的问题”

- Error: Planner used global search instead of navigating through Issues

- 错误: 规划者使用了全局搜索，而非通过问题

tab and filters

标签和筛选器导航

- Solution: Training data showing navigation through Issues section with

- 解决方案: 展示通过问题部分正确筛选导航的训练数据

proper filtering

2. ”Checkout merge requests requiring my review”

2. “查看需要我审核的合并请求”

- Error: Planner attempted repository search instead of using MR section

- 错误: 规划者尝试仓库搜索，而非使用带有审核筛选的合并请求部分

with review filter

- Solution: Examples showing how to access personal merge requests

- 解决方案: 展示如何访问个人合并请求的示例

##Class B: Profile/Project Settings Navigation Errors

##B 类: 个人资料/项目设置导航错误

###Description

描述

The planner fails to locate correct paths for user/project settings:

规划器未能找到用户/项目设置的正确路径:

- Not identifying correct navigation path for profile settings

  - 未能识别配置文件设置的正确导航路径

- Missing steps to access specific project settings sections

  - 缺少访问特定项目设置部分的步骤

- Using non-existent UI elements for status/member management

  - 使用不存在的 UI 元素进行状态/成员管理

###Training Data Needed

### 所需训练数据

- Examples showing correct profile settings navigation

  - 展示正确配置文件设置导航的示例

- Cases demonstrating project member management

  - 演示项目成员管理的案例

- Examples of updating user status through correct paths

  - 通过正确路径更新用户状态的示例

###Example Tasks

### 示例任务

1. ”Set my gitlab status as Enjoying life”

1. “将我的 GitLab 状态设置为享受生活”

- Error: Planner looked for non-existent ”Edit status” button instead of

- 错误: 规划器寻找不存在的“编辑状态”按钮，而非

profile settings path

配置文件设置路径

- Solution: Training data showing how to update profile settings and status

- 解决方案: 提供展示如何更新配置文件设置和状态的训练数据

2. ”Create a new public project and add members”

2. “创建一个新的公共项目并添加成员”

- Error: Planner tried accessing members through settings instead of

- 错误: 规划器试图通过设置访问成员，而非

project information page

项目信息页面

- Solution: Examples showing correct project member management workflow

- - 解决方案: 展示正确的项目成员管理工作流程的示例

##Class C: Repository Analysis Strategy Failures

##C 类: 仓库分析策略失败

###Description

描述

The planner fails to implement proper repository analysis strategies:

规划者未能实施适当的仓库分析策略:

- Not utilizing correct sorting/filtering for stars/contributions

  - 未正确使用星标/贡献的排序或筛选

- Missing steps to access personal repositories section

  - 缺少访问个人仓库部分的步骤

- Incorrect navigation for contribution analysis

  - 贡献分析的导航错误

###Training Data Needed

### 所需训练数据

- Examples showing repository sorting by stars

  - 展示按星标排序仓库的示例

- Cases demonstrating personal repository filtering

  - 演示个人仓库筛选的案例

- Examples of analyzing repository contributions

  - 分析仓库贡献的示例

###Example Tasks

### 示例任务

1. ”Tell me the repositories where I made contributions with most stars”

1. “告诉我我贡献过且星标最多的仓库有哪些”

- Error: Planner didn’t navigate to personal repositories section for

- - 错误: 规划者未导航至个人仓库部分以进行

proper star filtering

适当的星标筛选

- Solution: Training data showing how to filter and sort personal

- - 解决方案: 训练数据展示如何筛选和排序个人

repositories

仓库

##Class D: Commit Section Access Errors

##D 类: 提交区访问错误

###Description

描述

The planner fails to properly access the commits section of the repository:

规划器未能正确访问仓库的提交区:

- Not identifying the correct path to the commits section
  - 未能识别提交区的正确路径
- Missing steps to filter commits by date and author
  - 缺少按日期和作者筛选提交的步骤
- Incorrect navigation for commit history analysis
  - 提交历史分析的导航错误

###Training Data Needed

### 所需训练数据

- Examples showing correct navigation to the commits section
  - 展示正确导航至提交区的示例
- Cases demonstrating filtering commits by date and author
  - 演示按日期和作者筛选提交的案例
- Examples of analyzing commit history
  - 提交历史分析的示例

###Example Tasks

### 示例任务

1. ”How many commits did Eric and Kilian make to allyproject on 1/3/2023?”

1. “Eric 和 Kilian 在 2023 年 1 月 3 日对 allyproject 做了多少次提交?”

- Error: Planner didn’t navigate to the commits section or apply correct

- - 错误: 规划器未导航至提交区或未应用正确的

filters

筛选器

- Solution: Training data showing how to access the commits section and

- - 解决方案: 训练数据展示如何访问提交部分并

filter by date and author

按日期和作者筛选

##Other

其他

Description: If none of the above classes match.

描述: 如果以上类别均不匹配。

A.7.5. SHOPPING (OSS) FAILURE CLASSES PROMPT

A.7.5. 购物 (OSS) 失败类别提示

#Shopping Website Classes

# 购物网站类别

##Class A: Account Feature Navigation Failures

## 类别 A: 账户功能导航失败

###Description

描述

The planner fails to locate specific account-related features:

规划器未能定位特定的账户相关功能:

- Missing steps to access newsletter subscriptions

    - 缺少访问新闻订阅的步骤

- Not identifying correct paths for account settings

- 未识别账户设置的正确路径

- Incorrect navigation for personal features

- 个人功能导航错误

###Training Data Needed

### 所需训练数据

- Examples showing navigation to newsletter subscriptions

- 展示导航至新闻订阅的示例

- Cases demonstrating account settings access

- 演示账户设置访问的案例

- Examples of personal feature management

- 个人功能管理的示例

###Example Tasks

### 示例任务

1. ”Subscribe to the newsletter of OneStopMarket”

1. “订阅 OneStopMarket 的新闻通讯”

- Error: Planner didn’t identify path through account settings to

- 错误: 规划器未能识别通过账户设置到达

newsletter subscription

新闻通讯订阅的路径

- Solution: Training data showing navigation to newsletter subscription

- 解决方案: 训练数据展示导航至新闻通讯订阅

section

部分

##Class B: ’Advanced Search’ Feature Underutilization

##B 类:‘高级搜索’功能使用不足

###Description

描述

The planner fails to effectively use ’advanced search’ functionality:

规划器未能有效利用‘高级搜索’功能:

- Not utilizing price range filters in advanced search
  - 未在高级搜索中使用价格区间筛选
- Missing steps to combine category and price filtering
  - 缺少结合类别和价格筛选的步骤
- Using basic search when advanced search would be more efficient
  - 使用基础搜索，而高级搜索会更高效

###Training Data Needed

### 所需训练数据

- Examples showing proper use of advanced search with price filters
  - 展示正确使用带价格筛选的高级搜索的示例
- Cases demonstrating category + price range filtering
  - 演示类别 + 价格区间筛选的案例
- Examples of complex search criteria using advanced search
  - 使用高级搜索的复杂搜索条件示例

###Example Tasks

### 示例任务

---

1. ”Show me products under $30 in ’men shoes’ category”

1. “显示‘男鞋’类别中价格低于 30 美元的产品”

- Error: Planner used basic search instead of ’advanced search’ with price

- 错误: 规划器使用了基础搜索，而非带价格筛选的“高级搜索”

filter

筛选

- Solution: Training data showing how to use advanced search with category

- 解决方案: 训练数据展示如何使用带类别

and price range filters

和价格区间筛选的高级搜索

2. ”Buy the highest rated product from the meat substitute category within

2. “购买肉类替代品类别中评分最高的产品”

$100-200”
- Error: Planner didn’t utilize ’advanced search’ price range functionality

- 错误: 规划器未使用“高级搜索”的价格区间功能

- Solution: Examples showing how to combine category, price range and

- 解决方案: 示例展示如何结合类别、价格区间和

rating filters

评分筛选

##Other

其他

Description: If none of the above classes match.

描述: 如果以上类别均不匹配。

A.7.6.MAP FAILURE CLASSES PROMPT

A.7.6.MAP 失败类别提示

#Map Website Classes

# 映射网站类别

##Class A: Location Search Strategy Failures

## 类别 A: 位置搜索策略失败

###Description

描述

The planner fails to properly handle tasks requiring location search before

规划器未能正确处理需要先进行位置搜索的任务

directions:

再获取路线:

- Not searching to resolve generic/unspecified location references (e.g.,

  - 未进行搜索以解析通用/未指定的位置引用 (例如,

”nearest coffee shop”, ”a library”)

“最近的咖啡店”,“图书馆”)

- Attempting to get directions before resolving ambiguous locations through

  - 在通过搜索解析模糊位置之前尝试获取路线

search

- Missing steps to select specific locations from search results when generic

  - 在使用通用术语时，遗漏从搜索结果中选择具体位置的步骤

terms are used

###Training Data Needed

### 所需训练数据

- Examples showing proper workflow for resolving generic location references

  - 展示在获取路线前正确解析通用位置引用的工作流程示例

  before getting directions

  - 演示当一个或两个端点未具体命名时的搜索和选择案例

- Cases demonstrating search and selection when one or both endpoints are not

  specifically named

###Example Tasks

### 示例任务

1. ”Show me the walking distance from nearby hotels to Gardner Steel

1. “显示附近酒店到加德纳钢铁会议中心 (Gardner Steel Conference Center) 的步行距离”

   Conference Center”

   - Error: Planner jumped to directions without first searching for nearby

   - - 错误: 规划器在未先搜索附近酒店的情况下直接跳转到路线指引

   hotels

   酒店

   - Solution: Training data showing how to search for nearby locations before

   - - 解决方案: 训练数据展示如何在获取路线指引前先搜索附近地点

   getting directions

   获取路线指引

2. ”How long does it take to walk from Carnegie Museum of Art to a library at

2. “从卡内基艺术博物馆步行到图书馆需要多长时间”

CMU”
- Error: Planner tried direct routing without first identifying specific

- - 错误: 规划器在未先确定具体图书馆位置的情况下尝试直接规划路线

library location

图书馆位置

- Solution: Examples showing how to search for and select specific

- - 解决方案: 示例展示如何搜索并选择具体目的地

destinations

目的地

##Other

其他

Description: If none of the above classes match. For example, if the data

描述: 如果以上类别均不匹配。例如，若数据点包含在已命名地点间进行简单路线查找的任务，应归类为“其他”。

point contains a simple direction finding task between already named

数据点包含在已命名地点间进行简单路线查找的任务

locations, it should be classified as ”Other”.

应归类为“其他”。

A.8. Synthetic Plan Generation after Failure Analysis

A.8. 失败分析后的合成计划生成

After the failure analysis and the training data classification, our objective now is to generate data that is similar to the seed data (instead of generating diverse data). That is why this part has a slightly different prompt than the prompt above for the synthetic plan generator prompt at Appendix A.6. For each seed data point, we generate one more synthetic plan.

在失败分析和训练数据分类之后，我们的目标是生成与种子数据相似的数据 (而非生成多样化数据)。因此，本部分的提示与附录 A.6 中合成计划生成器的提示略有不同。对于每个种子数据点，我们生成一个额外的合成计划。

#Goal

目标

You are a Plan Data Generator tasked with producing one new synthetic data

您是一个计划数据生成器，负责生成一条新的合成数据

point from a single provided example. The new data point should:

仅凭一个提供的示例推断的新数据点应当:

1. **Preserve the same core user intention** as the original example. Avoid

1. ** 保持与原始示例相同的核心用户意图 **。避免

changing the main purpose or high-level goal of the user.

改变用户的主要目的或高层目标。

2. **Introduce minor variations** in details such as product names, numeric

2. ** 在产品名称、数字等细节上引入细微变化 **

values, or the user’s phrasing, to ensure the data point is not an exact

值，或用户的措辞，以确保数据点不是完全相同的

copy.

复制。

3. **Use the same Initial HTML State index** (unless otherwise specified) or a

3. ** 使用相同的初始 HTML 状态索引 **(除非另有说明) 或一个

context that is logically consistent with the original example’s HTML

与原始示例 HTML 逻辑一致的上下文

environment.

环境。

4. **Output a coherent high-level plan** that remains grounded in the

4. ** 输出一个连贯的高层计划 **，并保持其基础性

capabilities indicated by the initial HTML state and the provided example.

由初始 HTML 状态和提供的示例所指示的功能。

Your output must follow this format:

您的输出必须遵循此格式:

...
##Data Pair 1

## 数据对 1

User Query:

用户查询:

<new user query reflecting the same intention>
Global Plan:

全球计划:

##Step 1

## 步骤 1

Reasoning: [A concise but clear explanation of how you're building upon the

推理:[简明但清晰地说明你如何基于初始 HTML 状态并实现用户目标]

initial HTML state and addressing the user's goal]

初始 HTML 状态并实现用户目标]

Step: [A high-level step aimed at fulfilling part of the user's request]

步骤:[旨在完成用户请求部分内容的高级步骤]

##Step 2

## 步骤 2

Reasoning: [...]

推理:[...]

Step: [...]

步骤:[...]

...
##Important Details

## 重要细节

- ⋆⋆ Maintain the same overall user goal**. Do not drastically alter the user’s

  - ⋆⋆ 保持相同的整体用户目标 **。不要大幅更改用户的最终目标。例如，如果用户最初想“更新产品的库存水平”，则保持该高级目标。

  end objective. For example, if the user originally wanted to ”update the

  最终目标。例如，如果用户最初想“更新产品的库存水平”，则保持该高级目标。

  stock levels of a product,” keep that high-level aim.

  库存水平”，则保持该高级目标。

- **Preserve exact UI element names**: Never modify:

  - ** 保留精确的 UI 元素名称 **: 绝不修改:

  - Button names and labels

  - 按钮名称和标签

  - Form field identifiers

  - 表单字段标识符

  - Page names and URLs

  - 页面名称和 URL

  - Specific web element IDs or classes

  - 特定网页元素的 ID 或类名

  - Any technical identifiers used in the website

- - 网站中使用的任何技术标识符

- **Vary only non-technical details**. Changes should be limited to:

  - ** 仅变更非技术细节 **。更改应限于:

- User’s writing style and tone

- - 用户的写作风格和语气

- Generic product descriptions

- - 通用产品描述

- Numeric values (when not referring to specific UI elements)

- - 数值 (不涉及具体界面元素时)

- General context that doesn’t involve UI elements

- - 不涉及界面元素的一般上下文

##Language Variation Requirements

## 语言变体要求

- **Diverse Query Perspectives**: Generate queries from different viewpoints

  - ** 多样化查询视角 **: 从不同角度生成查询

such as:

例如:

- Direct requests: ”I need to...”

- - 直接请求:“我需要……”

- Question format: ”Could you help me...”

- - 疑问句式:“你能帮我……”

- Task-oriented: ”Look for...”

- - 任务导向:“寻找……”

- Casual tone: ”Hey, I want to...”

- - 轻松语气: “嘿，我想……”

- **Sentence Structure Variation**:

  - ** 句子结构变化 **:

- Vary between simple, compound, and complex sentences

- - 变换简单句、复合句和复杂句

- Mix up word order (e.g., ”The product inventory needs updating” vs ”I

- - 词序混合 (例如，“产品库存需要更新”与“我

need to update the product inventory”)

需要更新产品库存

- Use different transitional phrases and connectors

- - 使用不同的过渡短语和连接词

- **Vocabulary Diversity**:

  - ** 词汇多样性 **:

- Use synonyms and alternative expressions for common actions (e.g.,

- - 对常见动作使用同义词和替代表达 (例如,

”modify”, ”change”, ”update”, ”revise”, ”adjust”)

“修改”、“更改”、“更新”、“修订”、“调整”)

- Vary between formal and informal language styles

- - 在正式和非正式语言风格之间变化

- Avoid copying phrases verbatim from the example

- - 避免逐字复制示例中的短语

- ⋆⋆ Vary the objects, names, and locations** in the user query. For example,

- ⋆⋆ 在用户查询中变换对象、名称和地点。例如，

use different places, repositories, titles, products, ids, etc.

使用不同的地点、仓库、标题、产品、ID 等

- **NEVER modify the UI element names** (see the list above in ’## Important

- ** 绝不修改界面元素名称 **(见上文“## 重要细节”中的列表)

Details’)

详情’）

- **Keep the global plan structured and concise**. Each step should provide a

- ** 保持整体计划结构清晰简洁 **。每一步应提供

high-level sub-goal (”Apply filters”, ”Navigate to product page”, ”Update

一个高层次的子目标 (“应用筛选器”、“导航到产品页面”、“更新属性”等)，并将逻辑相关的操作归组。尽量不要

attributes”, etc.), and group logically related actions together. Try not

过多更改给定示例的计划，因为这些计划是

to change the plan of the given example too much since those plans are

我希望生成更多类似数据的真实示例

ground truth examples that I want to generate more data similar to in order

。

for the Planner to become better at that specific task.

使规划器在该特定任务上表现得更好。

- **Reasoning sections** in each step should briefly explain the sub-goal and

- 每一步中的 ** 推理部分 ** 应简要说明子目标及

how it connects to the overall intention, referencing any relevant elements

其如何与整体意图相连接，必要时引用初始 HTML 状态中的相关元素。

from the initial HTML state if necessary.

- **No hallucination** of features or UI elements not present in the initial

- ** 禁止虚构 ** 初始 HTML 状态中不存在的功能或界面元素。保持与现有结构和能力一致。

HTML state. Stay aligned with the existing structure and capabilities.

#Given Example

# 给定示例

{example_str}

{example_str}

#Task

任务

Generate a **single** new data point that preserves the user’s main goal but

生成一个 ** 单一 ** 的新数据点，保持用户的主要目标不变，但

changes some details. Output it exactly in the format described above while

更改部分细节。严格按照上述格式输出，同时

ensuring linguistic diversity in the generated content.

确保生成内容的语言多样性。

A.9. Replanner Data Annotator Prompt

A.9. 重新规划器数据标注提示

For the replanner data annotation, we provide all the previous plans, the current HTML state, and the future actions to the model and we ask it to generate a replan grounded on the future actions. For this, we have a system prompt that defines the goals of the replanner and we represent the previous rounds of replan as user-assistant messages, similar to how the Executor treats each user-assistant message pair as an HTML-action pair in Appendix A.4.

对于重新规划器数据标注，我们向模型提供所有先前的计划、当前的 HTML 状态和未来的操作，要求其基于未来操作生成重新规划。为此, 我们有一个系统提示定义重新规划器的目标, 并将之前的重新规划轮次表示为用户-助手消息，类似于执行器在附录 A.4 中将每对用户-助手消息视为 HTML-操作对的方式。

A.9.1. SYSTEM PROMPT

A.9.1. 系统提示

##Goal and Rules

## 目标与规则

You are the Global Planner agent, an expert plan generator for web navigation

您是全球规划代理，一个专门为网页导航生成计划的专家

tasks, responsible for providing high-level plans to help users achieve

任务，负责提供高级计划以帮助用户实现

their goals on a website. You will be assisting a user who is navigating a

他们在网站上的目标。您将协助一位正在浏览

simplified web interface to complete a task. The user will interact with

简化的网页界面以完成任务。用户将与之交互

the website by clicking on elements, typing text, and performing other

通过点击元素、输入文本及执行其他操作来使用该网站

actions. You will be given:

操作。您将获得:

- **User Query**: The web task that you are required to generate a global plan

  - ** 用户查询 **: 您需要生成一个全局计划的网络任务

for.

为了。

- **HTML**: The current HTML state of the web page.

  - **HTML**: 网页当前的 HTML 状态。

- ⋆⋆ Previous Actions**: The previous actions that the user has taken.

  - ⋆⋆ 之前的操作 **: 用户之前执行的操作。

- **Future Actions**: The future actions that the user will take.

  - ** 未来行动 **: 用户将采取的未来行动。

At each round of user-web interaction, you will generate a structured plan

在每一轮用户与网页的交互中，您将生成一个结构化计划

based on the user’s previous actions and the required future actions. Your

基于用户之前的操作和未来所需的操作。您的

goal is to:

目标是:

---

1. Cluster future actions into logical, high-level steps. This means that you

1. 将未来的操作聚合成逻辑性强的高级步骤。这意味着你

need to create steps that describe the overall goal rather than specific

需要创建描述整体目标的步骤，而非具体细节

fine-grained web interactions (clicks, types, etc.), where each step should

细粒度的网页交互 (点击、输入等)，每一步应

encapsulate one or more actions in the future trajectory.

封装未来轨迹中的一个或多个动作。

2. Classify each future action under an appropriate step

2. 将每个未来动作归类到适当的步骤中

3. Provide sufficient detail for the user to complete each step without

3. 提供足够的细节，使用户无需假设先验知识即可完成每一步

assuming prior knowledge

4. 不假设先验知识

Rules:

规则:

- For the first round, create a complete plan from scratch

- 第一轮从零开始制定完整计划

- For later rounds, incorporate previous actions in reasoning but only plan

- 后续轮次在推理时纳入之前的动作，但只规划未来步骤

future steps

- 未来步骤

- The plan should be updated each round as new actions become available.

- 随着新动作的出现，每轮应更新计划。

- Focus on high-level goals rather than specific web interactions, unless

- 关注高层目标，而非具体网页交互，除非为清晰起见

needed for clarity

- 需要时为清晰起见

- Group related actions logically to minimize the number of steps while

- 合理地将相关动作分组，以在保持清晰的同时减少步骤数量

maintaining clarity

13. 保持清晰

##Expected Output Format

## 预期输出格式

The plan you generate should be structured in a numbered list format, starting

你生成的计划应以编号列表格式结构化，起始于

with '## Step 1' and incrementing the step number for each subsequent step.

以“## Step 1”开头，后续步骤依次递增步骤编号。

Each step in the plan should be in this exact format:

计划中的每一步应采用以下准确格式:

...
##Step N

##Step N

Reasoning: [Your reasoning here]

推理:[在此填写您的推理]

Step: [Your step here]

步骤:[在此填写您的步骤]

...
Here is a breakdown of the components you need to include in each step of your

以下是您在计划每一步中需要包含的组成部分及其具体说明:

plan as well as their specific instructions:

计划及其具体说明:

- **Reasoning**: In this section, you should explain your reasoning and

- ** 推理 **: 在本节中，您应解释所提步骤背后的推理和思考过程。

thought process behind the step you are proposing. It should provide a

应提供对该步骤中各项行动为何被归为一组的高层次理由，

high-level justification for why the actions in this step are grouped

以及它们如何有助于实现整体目标。

together and how they contribute to achieving the overall goal. Your

您的推理应基于轨迹中可用的信息 (包括用户已采取的行动和

reasoning should be based on the information available in the trajectory

未来应采取的行动)，并应引导用户理解您计划背后的战略决策过程。

(both the actions the user has already taken and the future actions they

should take) and should guide the user in understanding the strategic

decision-making process behind your plan.

> Note: In the reasoning section of the first step, you should include an

> 注意: 在第一步的推理部分，您应包含一个

observation of the current HTML state of the task, including the

对任务当前 HTML 状态的 ** 观察 **，包括

elements, their properties, and their possible interactions. Your

元素、它们的属性及其可能的相互作用。您的

observation should be detailed and provide a clear understanding of the

观察应详尽且清晰地阐明

current state of the HTML page. You should also include a **reflection** on

HTML 页面的当前状态。你还应包括一段关于

the previous actions that have been taken so far.

迄今为止已采取的先前行动。

- **Description**: This section should include a brief description of the

- ** 描述 **: 本节应包含对…的简要描述

actions that are grouped together in this step. You should exactly copy the

在此步骤中被归为一组的操作。您应当准确复制

action descriptions from the trajectory without any modifications or

从轨迹中提取的动作描述，未作任何修改或

additional information. This is to ensure that the user can accurately map

附加信息。这样可以确保用户能够准确映射

the actions to the plan steps. Specifically, every action that you include

对计划步骤的操作。具体来说，您包含的每个操作

in your description should include any ’# Element’, ’# Note’, or ’# Exit’

你的描述中应包括任何“# 元素”、“# 注释”或“# 退出”

comments that are present in the trajectory as well as their corresponding

轨迹中存在的评论及其对应的

’do’ functions.

’do’ 函数。

- **Step**: In this section, you should provide a concise description of the

- ** 步骤 **: 在本节中，您应简要描述

global step being undertaken. Your step should summarize one or more

正在进行的全局步骤。您的步骤应总结一个或多个

actions from the trajectory as a logical unit. It should be as specific and

将轨迹中的动作作为一个逻辑单元。它应当尽可能具体且

concentrated as possible, without referring to any HTML or UI elements.

尽可能集中，不涉及任何 HTML 或用户界面元素。

Your step should focus on the logical progression of the task instead of

你的步骤应侧重于任务的逻辑进展，而非

the actual fine-grained interactions, such as clicks or types.

实际的细粒度交互，如点击或输入。

- **Actions**: This section should list the indexes of the actions associated

- ** 操作 **: 本节应列出相关操作的索引

with this step. One or more actions should be grouped under one broader

通过这一步骤，应将一个或多个操作归纳到一个更广泛的类别下

logical step. The indices in this section should exactly match the indices

逻辑步骤。本节中的索引应与索引完全匹配

of the actions in the trajectory.

轨迹中的动作。

##Examples

示例

Here are some examples of the expected output format for the plan where the

以下是该计划预期输出格式的一些示例，其中

input is the user query and the output is the structured plan that clusters

<input> 输入是用户查询，输出是将其聚类的结构化计划 </input>

multiple actions into the fewest number of logical steps possible without

将多个操作合并为尽可能少的逻辑步骤

sacrificing specificity:

牺牲特异性:

{examples}

{示例}

##Maintain Strict Order of Actions and Be Specific:

## 保持严格的操作顺序并具体说明:

- ⋆⋆ Strict order of actions**: Ensure every action and thought aligns with the

  - ⋆⋆ 严格的操作顺序 **: 确保每个动作和思考都符合

trajectory and plan, maintaining the strict order of actions. Actions

轨迹和计划，保持动作的严格顺序。动作

should be sequential, with no skipping or misalignment (e.g., avoid

应当是连续的，不得跳跃或错位 (例如，避免

assigning non-consecutive actions like Step 1: [0,3,4], Step 2: [1,2]).

分配非连续的操作，如步骤 1:[0,3,4]，步骤 2:[1,2])。

Deviation from the trajectory’s order will be PENALIZED!

偏离轨迹顺序将被处罚!

- **Specific instructions**: Provide clear, specific instructions for each

- ** 具体指示 **: 为每一步提供清晰、具体的指示，确保用户拥有完成操作所需的全部信息，而不依赖

step, ensuring the user has all the information needed without relying on

假设的知识。例如，明确说明“将‘New York’输入为航班的到达城市”，而不是模糊地说“输入到达城市”；或者不要说“为产品输入合适的评价”，

assumed knowledge. For example, explicitly state, ”Input ’New York’ as the

而应说“输入‘我喜欢这个产品’作为产品评价”。

arrival city for the flights,” instead of vague phrases like ”Input the

arrival city”; or instead of saying ”Type an appropriate review for the

product.” you should say ”Type ’I love this product’ as a review for the

product.”

##High-level Goals Guidelines:

高层目标指南:

- Focus on high-level goals rather than fine-grained web actions, while

- 关注高层目标，而非细粒度的网页操作，同时保持对需要完成内容的具体描述。每一步应代表一个有意义的工作单元，可能包含多个

maintaining specificity about what needs to be accomplished. Each step

低层操作 (点击、输入等)，这些操作服务于共同目的，但

should represent a meaningful unit of work that may encompass multiple

仍应精确描述预期结果。例如，代替

low-level actions (clicks, types, etc.) that serve a common purpose, but

should still be precise about the intended outcome. For example, instead of

having separate steps for clicking a search box, typing a query, and

将点击搜索框、输入查询和

clicking search, combine these into a single high-level but specific step

点击搜索，将这些合并为一个高级但具体的步骤

like ”Search for X product”.

如“搜索 X 产品”。

- Group related actions together that achieve a common sub-goal. Multiple

- 将实现共同子目标的相关操作组合在一起。多个

actions that logically belong together should be combined into a single

逻辑上相关的操作应合并为一个

step. For example, multiple filter-related actions can be grouped into a

步骤。例如，多个与过滤器相关的操作可以被归纳为一个

single step like ”Apply price range filters between \$100-\$200 and select

单步操作，如“应用价格范围过滤器，介于 \$100-\$200 之间，并选择

5-star rating”. The key is to identify actions that work together to

“五星评级”。关键是识别协同作用的行为

accomplish a specific objective while being explicit about the criteria and

明确标准并完成特定目标

parameters involved.

涉及的参数。

- Focus on describing WHAT needs to be accomplished rather than HOW it will be

- 重点描述需要完成的内容，而非如何完成

implemented. Your steps should clearly specify the intended outcome without

已实现。您的步骤应明确说明预期结果，而不应

getting into the mechanics of UI interactions. The executor agent will

深入探讨用户界面交互的机制。执行代理将

handle translating these high-level but precise steps into the necessary

处理将这些高级但精确的步骤转化为必要的内容

sequence of granular web actions.

颗粒化网页操作序列。

##Search Results and Dynamic Content Guidelines:

## 搜索结果与动态内容指南:

- CRITICAL: Since you are like a data annotator, which is given the ground

- 重要提示: 由于你类似于数据标注员，负责提供基础数据

truth action trajectory, you might be tempted to output steps that directly

真实动作轨迹，你可能会倾向于输出直接的步骤

describe dynammic search results that appears in future actions. You MUST

描述出现在未来操作中的动态搜索结果。您必须

NOT do this. User will not have access to the trajectory or the actions in

不要这样做。用户将无法访问轨迹或其中的操作

the trajectory beforehand like you do. Because of this, if your task

事先像你那样规划轨迹。正因为如此，如果你的任务

requires you to ”search” for something and analyze the search results, you

需要你“搜索”某些内容并分析搜索结果，你

should output high-level steps such as ”Analyze the search results for gas

应输出诸如“分析气体搜索结果”等高级步骤

stations and note their locations” or ”Look through the orders to find

记录车站并标注其位置”或“查看订单以查找

order number 178” and let the user focus on the high-level steps. You will

订单号 178”，让用户专注于高级步骤。您将

have the chance to look at the search results in the future steps when you

有机会在后续步骤中查看搜索结果

see them in the current HTML state. Until then, please just reference the

在当前的 HTML 状态中查看它们。在此之前，请仅参考

search results in high-level terms.

以高级术语进行搜索结果。

##Formatting Guidelines:

## 格式指南:

- Start your response with the ’## Step 1’ header and follow the format

- 以“## 第一步”标题开始你的回复，并遵循该格式

provided in the examples.

示例中提供的。

- Ensure that each step is clearly separated and labeled with the ’## Step N′

- 确保每个步骤都清晰分隔并标注为“## 第 N′ 步”

header, where N is the step number.

标题，其中 N 是步骤编号。

- Include the ’Reasoning’, ’Description’, ’Step’, and ’Actions’ sections in

- 包含“推理”、“描述”、“步骤”和“操作”部分

each step.

每个步骤中。

A.9.2. USER-ASSISTANT MESSAGES

A.9.2. 用户-助手消息

Each round of replanning is formulated as a user-assistant message pair where the assistant messages are the previous plans and the user messages are in the following format.

每轮重新规划被表述为用户-助手消息对，其中助手消息是之前的计划，用户消息格式如下。

All previous user messages are represented in the following format since we don’t want to dump the entire HTML and the future actions into the context:

所有之前的用户消息均以以下格式表示，因为我们不想将整个 HTML 和未来操作全部放入上下文:

##Round {index}

## 第 {index} 轮

##HTML

##HTML

** Simplified html **

** 简化的 html **

##Action taken

## 已执行操作

{previous action taken}

{previous action taken}

##Future Actions Trajectory

## 未来操作轨迹

** Future actions **

** 未来操作 **

And here is the last user message where we provide the current HTML state and the future actions for which it needs to replan:

以下是最后一条用户消息，我们提供当前 HTML 状态及其需要重新规划的未来操作:

##Round {last action index}

## 第 {last action index} 轮

##HTML

##HTML

{current_html_state}

{current_html_state}

##Future Actions Trajectory

## 未来行动轨迹

The following is the future trajectory to complete the web navigation task. It

以下是完成网页导航任务的未来轨迹。它

is formatted as series of actions where each action first has a comment

被格式化为一系列操作，每个操作首先带有一个注释

('#') that describes the element to be interacted with or a note which

('#') 描述要交互的元素或注释

provides some context about the action and the current task state. The

提供有关动作和当前任务状态的一些背景信息。

action is then described with the do function, which takes two arguments:

然后用 do 函数描述动作，该函数接受两个参数:

the action to be performed, the element to be interacted with, and

要执行的操作、要交互的元素，以及

sometimes an argument. The actions are numbered sequentially to indicate

有时是一场争论。动作按顺序编号以示

the order in which they should be executed:

它们应被执行的顺序:

{future_trajectory}

{future_trajectory}

You MUST start with the ’## Step 1’ header and follow the format provided in

您必须从“## 第一步”标题开始，并遵循所提供的格式

the examples.

这些例子。

A.10. Replanner Prompt

A.10. 重新规划提示

Similar to the replanner data annotator prompt in Appendix A.9, we represent the previous rounds of replans as user-assistant message pairs. The only difference is that the replanner doesn’t know the future actions. Also, it has a system prompt that defines the high-level goals of the replanner.

类似于附录 A.9 中的重新规划器数据标注提示，我们将之前的重新规划轮次表示为用户与助手的消息对。唯一的区别是重新规划器不知道未来的动作。此外，它还有一个系统提示，用于定义重新规划器的高层目标。

A.10.1. SYSTEM PROMPT

A.10.1. 系统提示

#Goal and Rules

# 目标与规则

You are an expert plan generator for web navigation tasks, responsible for

你是一名网页导航任务的专家计划生成器，负责

providing high-level plans to help users achieve their goals on a website.

提供高层次的计划，帮助用户在网站上实现他们的目标。

You will be assisting a user who is navigating a simplified web interface

你将协助正在使用简化网页界面的用户

to complete a task. The user will interact with the website by clicking on

完成任务。用户将通过点击

elements, typing text, and performing other actions. You will be given:

元素、输入文本及执行其他操作与网站交互。你将获得:

- ⋆⋆ User Query ⋆⋆ : The web task that you are required to generate a global plan

  - ⋆⋆ 用户查询 ⋆⋆ : 你需要为其生成全局计划的网页任务。

for.

- **HTML**: The current HTML state of the web page.

  - **HTML**: 网页当前的 HTML 状态。

- **Previous Actions**: The previous actions that the user has taken.

  - ** 先前操作 **: 用户之前执行的操作。

- **Previous Global Plans**: The previous global plans generated in the

- ** 先前全局计划 **: 之前轮次生成的全局计划。

previous rounds.

At each round of user-web interaction, you will generate a structured plan

在每轮用户与网页的交互中，你将基于用户的先前操作、当前 HTML 状态及

based on the user’s previous actions, current HTML state, and the previous

先前的全局计划，生成结构化的计划。

global plans.

Rules:

规则:

- For the first round, create a complete plan from scratch

- 第一轮，从零开始制定完整计划

- For later rounds, incorporate previous actions in reasoning but only plan

- 对于后续回合，在推理中纳入先前的动作，但仅限于规划

future steps

未来步骤

- The plan should be updated each round as new actions become available.

- 随着新行动的出现，计划应在每轮中更新。

- Keep the plan concise and actionable

- 保持计划简明且可执行

- Focus on high-level goals rather than specific web interactions, unless

- 关注高层次目标，而非具体的网页交互，除非

needed for clarity

为清晰起见所需

Remember:

请记住:

Since the previous global plans were constructed without seeing the current

由于之前的全球规划是在未见当前情况的情况下制定的

state of the HTML that you are viewing now, they may include steps that are

您当前查看的 HTML 状态，可能包括一些步骤

not needed (e.g., less efficient, unrelated, or wrong) or miss some

不需要的 (例如，效率较低、不相关或错误的) 或遗漏了一些

important actions that are required to proceed further. In these cases

继续进行所需的重要操作。在这些情况下

where the previous global plan needs to be refined based on the current

之前的全局计划需要根据当前情况进行细化

state of the HTML, your key responsibility is to make the previous plan

根据 HTML 的状态，您的主要职责是制定之前的计划

more specific by:

更具体地通过:

1. Identifying which steps from the previous plan are now possible/visible

1. 确定之前计划中哪些步骤现在可行/可见

based on the current HTML state

基于当前的 HTML 状态

2. Updating those steps with specific details you can now see (e.g., exact

2. 用你现在能看到的具体细节更新这些步骤 (例如，确切的

items to click, specific text to enter)

点击项，输入的具体文本)

3. Removing steps that are no longer relevant or needed

3. 删除不再相关或不需要的步骤

4. Adding new steps if the current state reveals necessary additional actions

4. 如果当前状态显示需要额外操作，则添加新步骤

5. Fixing any errors or assumptions based on the current state

5. 根据当前状态修正任何错误或假设

6. Adapting the plan if expected elements or results are not found

6. 如果未找到预期的元素或结果，则调整计划

For example:

例如:

- If a previous step was ”search for products”, and you now see search

- 如果之前的步骤是“搜索产品”，而你现在看到了搜索

results, update the plan with which specific result to select

结果，则更新计划，指定选择哪个具体结果

- If a previous step was ”navigate to a section”, and you now see the

- 如果之前的步骤是“导航到某个部分”，而你现在看到了

navigation options, specify which exact link/button to use

导航选项，则明确使用哪个具体链接/按钮

- If a previous step was ”find an item”, and the item is not found, provide

- 如果之前的步骤是“查找某个项目”，但未找到该项目，

alternative items or navigation paths

则提供替代项目或导航路径

Consider the previous global plans when generating the new plan, decide

生成新计划时考虑之前的整体计划，决定是否做出任何更改，

whether to make any changes, and provide your reasoning.

并说明你的理由。

##Expected Output Format

## 预期输出格式

The plan you generate should be structured in a numbered list format, starting

你生成的计划应以编号列表格式结构化，起始于

with ’ ## Step 1′ and incrementing the step number for each subsequent step.

“## 第 1 步”，后续步骤依次递增步骤编号。

Each step in the plan should be in this exact format:

计划中的每一步应采用以下准确格式:

...
##Step N

## 第 N 步

Reasoning: [Your reasoning here]

推理:[在此填写你的推理]

Step: [Your step here]

步骤:[在此填写你的步骤]

...

Here is a breakdown of the components you need to include in each step of your

以下是你需要在计划每一步中包含的组成部分及其具体说明:

plan as well as their specific instructions:

- **Reasoning**: In this section, you should explain your reasoning and

- ** 推理 **: 在本节中，你应解释所提步骤背后的推理和思考过程。它应提供

thought process behind the step you are proposing. It should provide a

对该步骤中行动为何被归为一组的高层次理由，以及这些行动如何有助于实现整体目标。

high-level justification for why the actions in this step are grouped

你的推理应基于轨迹中可用的信息

together and how they contribute to achieving the overall goal. Your

(包括用户已采取的行动和他们应采取的未来行动)，并应指导用户理解战略

reasoning should be based on the information available in the trajectory

(both the actions the user has already taken and the future actions they

should take) and should guide the user in understanding the strategic

decision-making process behind your plan.

你计划背后的决策过程。

> Note: In the reasoning section of the first step, you should include an

> 注意: 在第一步的推理部分，您应包含一个

observation of the current HTML state of the task, including the

对任务当前 HTML 状态的 ** 观察 **，包括

elements, their properties, and their possible interactions. Your

元素、它们的属性及其可能的相互作用。您的

observation should be detailed and provide a clear understanding of the

观察应详尽且清晰地阐明

current state of the HTML page. You should also include a **reflection** on

HTML 页面的当前状态。你还应包括一个 ** 反思 **

the previous actions that have been taken so far. This reflection should

迄今为止所采取的先前行动。这一反思应当

include:

包括:

- What were the previous actions that were taken?

- 之前采取了哪些措施?

- Were the previous actions successful? How do you know this from the

- - 之前的操作成功了吗？你是如何从中得知的?

current HTML state? For example, if the previous action was to type in

当前的 HTML 状态？例如，如果之前的操作是输入

an input field, you MUST reflect on whether the input field is now

一个输入字段，您必须考虑该输入字段当前是否

populated with the correct text.

填充正确的文本。

- **Step**: In this section, you should provide a concise description of the

- ** 步骤 **: 在本节中，您应简要描述

global step being undertaken. Your step should summarize one or more

正在进行的全局步骤。您的步骤应总结一个或多个

actions from the trajectory as a logical unit. It should be as specific and

将轨迹中的动作作为一个逻辑单元。它应当尽可能具体且

concentrated as possible, without referring to any HTML or UI elements.

尽可能集中，不涉及任何 HTML 或用户界面元素。

Your step should focus on the logical progression of the task instead of

您的步骤应侧重于任务的逻辑进展，而非

the actual fine-grained interactions, such as clicks or types.

具体的细节操作，如点击或输入。

Be Specific:

具体说明:

- **Specific instructions**: Provide clear, specific instructions for each

- ** 具体指令 **: 为每一步提供清晰、具体的指令，确保用户拥有完成任务所需的全部信息，而不依赖于

step, ensuring the user has all the information needed without relying on

假设的知识。例如，明确说明“将‘New York’输入为航班的到达城市”，而不是模糊地说“输入到达城市”；或者不要说“为产品输入合适的评价”，而应说“输入‘我喜欢这个产品’作为产品评价”。

assumed knowledge. For example, explicitly state, ”Input ’New York’ as the

arrival city for the flights,” instead of vague phrases like ”Input the

arrival city”; or instead of saying ”Type an appropriate review for the

product.” you should say ”Type ’I love this product’ as a review for the

product.”

##High-level Goals Guidelines:

高层目标指导原则:

- Focus on high-level goals rather than fine-grained web actions, while

- 关注高层目标，而非细节的网页操作，同时保持对需要完成内容的具体描述。每一步

maintaining specificity about what needs to be accomplished. Each step

应代表一个有意义的工作单元，可能包含多个

should represent a meaningful unit of work that may encompass multiple

低层操作 (点击、输入等)，但这些操作服务于共同的目的，且

low-level actions (clicks, types, etc.) that serve a common purpose, but

should still be precise about the intended outcome. For example, instead of

仍应明确预期的结果。例如，不要说

having separate steps for clicking a search box, typing a query, and

将点击搜索框、输入查询和

clicking search, combine these into a single high-level but specific step

点击搜索，将这些合并为一个高层次但具体的步骤

like ”...

像“...

- Group related actions together that achieve a common sub-goal. Multiple

- 将实现共同子目标的相关操作组合在一起。多个

actions that logically belong together should be combined into a single

逻辑上相关的操作应合并为一个

step. For example, multiple filter-related actions can be grouped into a

步骤。例如，多个与过滤器相关的操作可以被归纳为一个

single step like ”Apply price range filters between $100-$200 and select

单步操作，如“应用价格区间过滤器，范围在 100 美元至 200 美元之间，并选择

5-star rating”. The key is to identify actions that work together to

“五星评级”。关键是识别协同作用的行为

accomplish a specific objective while being explicit about the criteria and

明确标准并完成特定目标

parameters involved.

涉及的参数。

- Focus on describing WHAT needs to be accomplished rather than HOW it will be

  - 重点描述需要完成的内容，而非如何完成

implemented. Your steps should clearly specify the intended outcome without

已实现。您的步骤应明确说明预期结果，而不应

getting into the mechanics of UI interactions. The executor agent will

深入探讨用户界面交互的机制。执行代理将

handle translating these high-level but precise steps into the necessary

处理将这些高级但精确的步骤转化为必要的内容

sequence of granular web actions.

颗粒化网页操作序列。

##Formatting Guidelines:

## 格式指南:

- Start your response with the '## Step 1' header and follow the format

  - 以“## 第 1 步”标题开始你的回复，并遵循示例中提供的格式

provided in the examples.

。

- Ensure that each step is clearly separated and labeled with the '## Step N′

  - 确保每一步都清晰分隔，并用“## 第 N′ 步”标题标注，其中 N 为步骤编号。

header, where N is the step number.

- Include the ’Reasoning’ and ’Step’ sections in each step.

- 每一步都包含“推理”和“步骤”部分。

A.10.2. USER-ASSISTANT MESSAGES

A.10.2. 用户-助手消息

Each round of replanning is formulated as a user-assistant message pair where the assistant messages are the previous plans and the user messages are in the following format:

每轮重新规划都被表述为用户-助手消息对，其中助手消息是之前的计划，用户消息格式如下:

All previous user messages are represented in the following format since we don’t want to dump the entire HTML into the context:

所有之前的用户消息都用以下格式表示，因为我们不想将整个 HTML 内容放入上下文中:

#Previous Actions

# 之前的操作

** List of previous actions **

** 之前操作列表 **

#HTML

#HTML

** Simplified html **

** 简化的 HTML **

Here is the last user message where we provide the list of previous actions and the current HTML state upon which the model needs to base its replan.

这是最后一条用户消息，我们提供了之前操作列表和当前 HTML 状态，模型需要基于此进行重新规划。

#Previous Actions

# 之前的操作

{previous actions of the executor}

{执行者的之前操作}

#HTML

#HTML

{obs}

{obs}

A.11. WebArena Performance Breakdown

A.11. WebArena 性能细分

Figure 4. Task performance metrics by website.

图 4. 各网站的任务性能指标。

Website	#Tasks	Avg. Steps (All)	Avg. Steps (Success)	Avg. Steps (Fail)	Success Rate (%)
Overall	165	11.12	7.52	13.43	53.9
GitLab	30	13.70	5.98	20.35	53.3
Reddit	19	9.37	8.31	9.92	84.2
Shopping Admin	35	12.40	8.65	14.41	48.6
Shopping	45	9.87	7.11	10.66	55.6
Map	26	10.00	10.37	9.10	46.2
Multiple Websites	10	11.70	6.00	17.83	30.0

网站	任务数	平均步骤数 (全部)	平均步骤数 (成功)	平均步骤数 (失败)	成功率 (%)
总体	165	11.12	7.52	13.43	53.9
GitLab	30	13.70	5.98	20.35	53.3
Reddit	19	9.37	8.31	9.92	84.2
购物管理	35	12.40	8.65	14.41	48.6
购物	45	9.87	7.11	10.66	55.6
地图	26	10.00	10.37	9.10	46.2
多个网站	10	11.70	6.00	17.83	30.0

A.12. Hyperparameters

A.12. 超参数

Training Hyperparameters	
Learning Rate	2e-5
Optimizer	AdamW
LR Scheduler	Cosine
Warmup Ratio	0.1
Batch Size	32
Epochs	1
FP16/BF16	Enabled
Machine	8×A100
Framework	torchtune
(a) Training	
Inference Hyperparameters	
Temperature	0
Framework	vLLM
Max tokens generated	4196
Maximum sequence length	32000

训练超参数	
学习率	2e-5
优化器	AdamW(AdamW 优化器)
学习率调度器	余弦退火
预热比例	0.1
批量大小	32
训练轮数	1
FP16/BF16	启用
机器	8×A100
框架	torchtune
(a) 训练	
推理超参数	
温度	0
框架	vLLM
最大生成标记数	4196
最大序列长度	32000

(b) Inference

(b) 推理

Figure 5. Model hyperparameters for training and inference

图 5. 训练与推理的模型超参数