# AppAgent: Multimodal Agents as Smartphone Users

# AppAgent: 作为智能手机用户的多模态代理

Chi Zhang* Zhao Yang* Jiaxuan Liu* Yucheng Han Xin Chen

张驰 * 杨钊 * 刘佳轩 * 韩宇成陈鑫

Zebiao Huang Bin Fu Gang Yu†

黄泽彪傅斌钢 Yu†

Tencent

腾讯

{johnczhang, jayzyang, jiaxuanliu, yuchenghan, shingxchen, zebiaohuang, brianfu, skicyyu}@tencent.com
https://appagent-official.github.io/

{johnczhang, jayzyang, jiaxuanliu, yuchenghan, shingxchen, zebiaohuang, brianfu, skicyyu}@tencent.com
https://appagent-official.github.io/

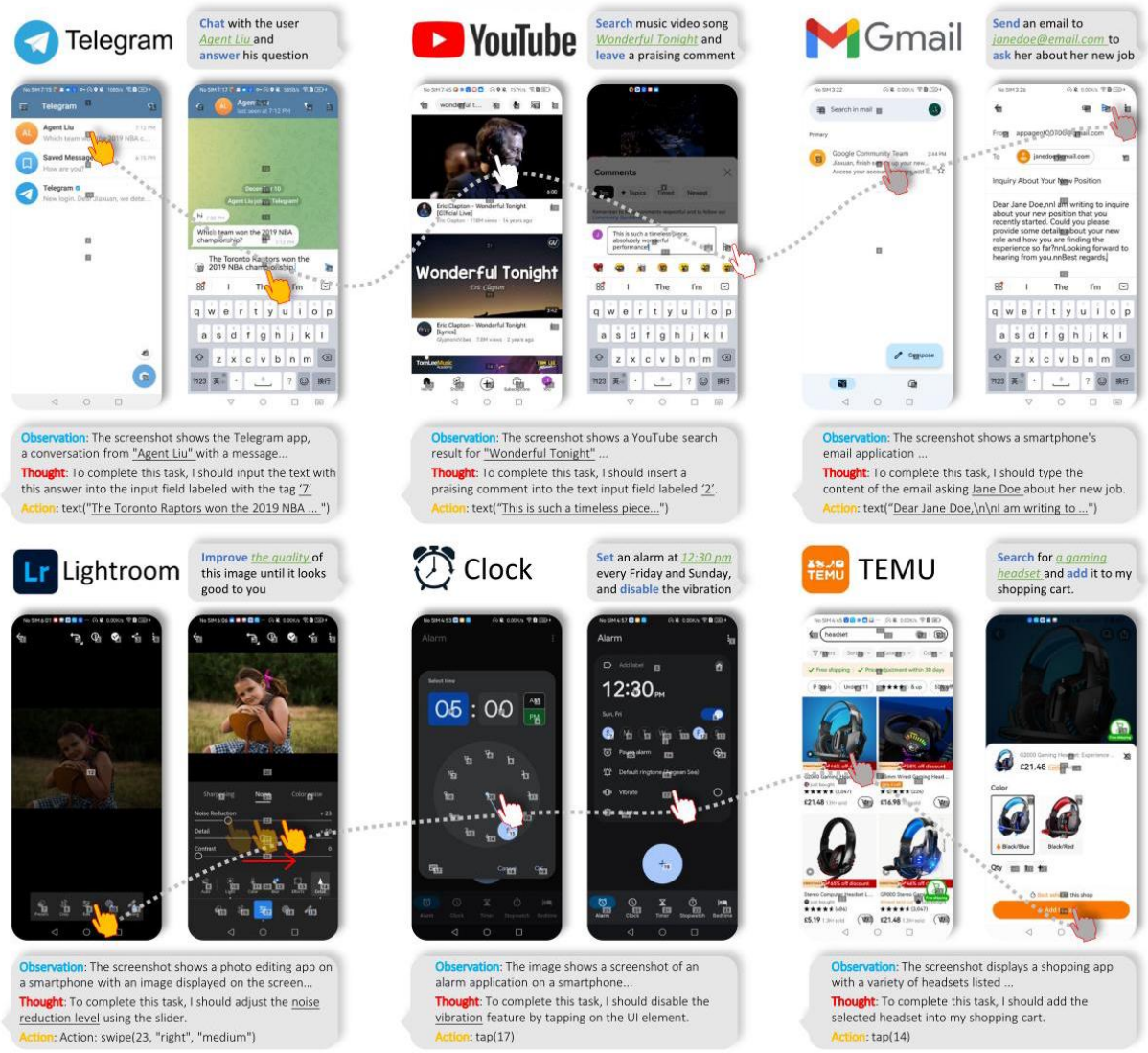


Figure 1: Diverse applications of our multimodal agent framework for smartphone App operation. We evaluate the effectiveness of our agent model on 50 tasks across 10 different Apps, highlighting its adaptability and effectiveness in a real-world context.

图 1: 我们多模态代理框架在智能手机应用操作中的多样化应用。我们在 10 个不同应用中评估了该代理模型在 50 个任务上的有效性，突出其在真实环境中的适应性和效果。

## Abstract


## 摘要

Recent advancements in large language models (LLMs) have led to the creation of intelligent agents capable of performing complex tasks. This paper introduces a novel LLM-based multimodal agent framework designed to operate smartphone applications. Our framework enables the agent to operate smartphone applica-

近年来大型语言模型 (LLMs) 的进步催生了能够执行复杂任务的智能代理。本文提出了一种基于 LLM 的新型多模态代理框架，旨在操作智能手机应用。我们的框架使代理能够通过简化的动作空间操作手机应用，模拟人类的点击和滑动等交互方式。

tions through a simplified action space, mimicking human-like interactions such as tapping and swiping. This novel approach bypasses the need for system back-end access, thereby broadening its applicability across diverse apps. Central to our agent's functionality is its innovative learning method. The agent learns to navigate and use new apps either through autonomous exploration or by observing human demonstrations. This process generates

这一创新方法绕过了系统后端访问的需求，从而拓宽了其在多样化应用中的适用性。代理功能的核心是其创新的学习方法。代理通过自主探索或观察人类示范学习导航和使用新应用。这一过程生成

---

*Equal contributions.

* 同等贡献。

† Corresponding Author.

† 通讯作者。

---

a knowledge base that the agent refers to for executing complex tasks across different applications. To demonstrate the practicality of our agent, we conducted extensive testing over 50 tasks in 10 different applications, including social media, email, maps, shopping, and sophisticated image editing tools. The results affirm our agent's proficiency in handling a diverse array of high-level tasks.

一个知识库，代理据此执行跨不同应用的复杂任务。为展示代理的实用性，我们在包括社交媒体、电子邮件、地图、购物及高级图像编辑工具等 10 个应用中进行了 50 个任务的广泛测试。结果证明了代理处理多样高阶任务的能力。

# 1 Introduction

# 1 引言

The emergence of large language models (LLMs), such as ChatGPT (OpenAI, 2021) and GPT-4 (Ope-nAI, 2023), marks a significant milestone in the field of artificial intelligence and natural language processing. These advanced models represent a fundamental change in how machines understand and generate human language, exhibiting a level of sophistication and versatility previously unattainable. One of the most exciting developments in this field is the capability of LLMs to function not just as language processors, but as agents capable of performing complex tasks. This evolution is evident in initiatives such as AutoGPT (Yang et al., 2023a) and MetaGPT (Hong et al., 2023), which showcase the practical applications of LLMs in tasks requiring advanced cognitive functions like reasoning, planning, and collaboration. The significance of these developments cannot be overstated, as they

extend the utility of LLMs beyond simple language tasks, revolutionizing various aspects of technology and daily life.

大型语言模型 (LLMs) 的出现，如 ChatGPT(OpenAI，2021) 和 GPT-4(OpenAI，2023)，标志着人工智能和自然语言处理领域的重要里程碑。这些先进模型代表了机器理解和生成自然语言方式的根本变革，展现出前所未有的复杂性和多功能性。该领域最令人振奋的发展之一是 LLMs 不仅作为语言处理器，还能作为执行复杂任务的代理。这一演进在 AutoGPT(Yang 等，2023a) 和 MetaGPT(Hong 等，2023) 等项目中得以体现，展示了 LLMs 在推理、规划和协作等高级认知功能任务中的实际应用。这些发展意义重大，扩展了 LLMs 在简单语言任务之外的应用，革新了技术和日常生活的多个方面。

However, a key limitation of these LLM-based agents has been their reliance solely on text-based information. This restriction has historically curtailed their perception and interaction with their environment. The introduction of models equipped with vision capabilities, such as the latest iteration of GPT-4, marks a pivotal breakthrough. By integrating the ability to process and interpret visual information, these models can now understand aspects of their surroundings that are difficult or impossible to convey through text alone. This extended capability enables LLMs to interpret context, recognize patterns, and respond to visual cues, thus providing a more holistic and interactive experience with the world.

然而，这些基于 LLM 的代理的一个关键限制是它们仅依赖文本信息。这一限制历来限制了它们对环境的感知和交互。配备视觉能力的模型的引入，如最新版本的 GPT-4，标志着一个关键突破。通过整合处理和解释视觉信息的能力，这些模型现在能够理解文本难以或无法传达的环境细节。这种扩展能力使 LLMs 能够解读上下文、识别模式并响应视觉线索，从而提供更全面和互动的世界体验。

In our work, we focus on building a multimodal agent leveraging the vision capabilities of multimodal large language models to undertake tasks previously unachievable by text-only agents. In particular, we explore an interesting but challenging application that builds an agent to operate any smartphone application (App) in the mobile operating system. Our approach differs significantly from existing intelligent phone assistants like Siri, which operate through system back-end access and function calls. Instead, our agent interacts with smartphone apps in a human-like manner, using low-level operations such as tapping and swiping on the graphical user interface (GUI). The proposed agent offers multiple advantages. Firstly, it eliminates the need for system back-end access, making our agent universally applicable across various applications. Additionally, this approach enhances security and privacy, as the agent does not require deep system integration. Furthermore, by operating on the GUI level, our agent can adapt to changes in app interfaces and updates, ensuring long-term applicability and flexibility.

在本研究中，我们专注于构建利用多模态大型语言模型视觉能力的多模态代理，以完成文本代理无法实现的任务。特别地，我们探索了一个有趣且具挑战性的应用——构建一个能够操作移动操作系统中任意智能手机应用 (App) 的代理。我们的方法与现有智能手机助手如 Siri 显著不同，后者通过系统后端访问和函数调用操作。相反，我们的代理以类人方式通过点击和滑动等低级操作与手机应用的图形用户界面 (GUI) 交互。所提代理具有多重优势。首先，它无需系统后端访问，使代理在各种应用中通用。此外，该方法提升了安全性和隐私性，因为代理不需深度系统集成。再者，通过在 GUI 层面操作，代理能适应应用界面和更新的变化，确保长期适用性和灵活性。

However, creating a multimodal agent capable of operating diverse smartphone apps presents significant challenges. Existing research indicates that adapting current models for embodied tasks necessitates extensive training data, and collecting a large dataset of app demonstrations for training is a formidable task. Moreover, different

apps have unique GUIs with varying icon meanings and operational logic, and it remains uncertain whether these adapted models can effectively generalize to unseen apps.

然而，创建一个能够操作多种智能手机应用的多模态代理面临重大挑战。现有研究表明，将现有模型适配于具身任务需要大量训练数据，而收集大量应用演示数据用于训练是一项艰巨的任务。此外，不同应用具有独特的图形用户界面 (GUI)，图标含义和操作逻辑各异，尚不确定这些适配后的模型能否有效泛化到未见过的应用。

In this paper, we introduce a multimodal agent framework aimed at operating any smartphone app like human users. The learning of our framework involves an exploration phase where the agent interacts autonomously with apps through a set of pre-defined actions and learns from their outcomes. These interactions are documented, which assists the agent in navigating and operating the apps. This learning process can be accelerated by observing a few human demonstrations. Following this exploratory phase, the agent can operate the app by consulting the constructed document based on its current state, eliminating the need to adapt the parameters of the LLMs or collect extensive training data for each app.

本文提出了一种多模态代理框架，旨在像人类用户一样操作任何智能手机应用。该框架的学习过程包括一个探索阶段，代理通过一组预定义动作自主与应用交互，并从交互结果中学习。这些交互被记录下来，帮助代理导航和操作应用。通过观察少量人类演示，可以加速该学习过程。探索阶段结束后，代理可根据其当前状态查阅构建的文档来操作应用，无需调整大型语言模型 (LLM) 的参数或为每个应用收集大量训练数据。

To validate its effectiveness, we tested our agent on 50 tasks across 10 different apps, ranging from social media and messaging to email, maps, shopping, and even complex image editing apps. Both quantitative results and user studies underscore the advantages of our design, particularly its adaptability, user-friendliness, and efficient learning and

为了验证其有效性，我们在 10 个不同应用上测试了代理的 50 个任务，涵盖社交媒体、消息、电子邮件、地图、购物，甚至复杂的图像编辑应用。定量结果和用户研究均强调了我们设计的优势，特别是其适应性、用户友好性以及高效的学习和

operating capabilities across a wide range of applications. This underlines the potential of our agent as a versatile and effective tool in the realm of smartphone app operation.

操作能力，适用于广泛的应用场景。这凸显了我们的代理作为智能手机应用操作领域多功能且高效工具的潜力。

In summary, this paper makes the following contributions:

总之，本文的主要贡献包括:

- We open-source a multimodal agent framework, focusing on operating smartphone applications with our developed action space.

  - 我们开源了一个多模态代理框架，重点是通过我们开发的动作空间操作智能手机应用。

- We propose an innovative exploration strategy, which enables the agent to learn to use novel apps.

- 我们提出了一种创新的探索策略，使代理能够学习使用新颖的应用。

- Through extensive experiments across multiple apps, we validate the advantages of our framework, demonstrating its potential in the realm of AI-assisted smartphone app operation.

- 通过在多个应用上的大量实验，我们验证了框架的优势，展示了其在 AI 辅助智能手机应用操作领域的潜力。

## 2 Related Work

## 2 相关工作

## 2.1 Large language models

## 2.1 大型语言模型

The development of ChatGPT (OpenAI, 2021) and GPT-4 (OpenAI, 2023) represents a crucial advancement in natural language processing. Unlike earlier large language models (LLMs), these new models (Touvron et al., 2023a, b; Zeng et al., 2022; Taori et al., 2023; Zheng et al., 2023) enable multi-round conversations and have the impressive ability to follow complex instructions. The integration of vision capabilities in GPT-4V (Yang et al., 2023b) is a further milestone, enabling the language model to process and interpret visual data. This addition has broadened the scope of potential AI applications, allowing GPT-4 to undertake diverse tasks such as problem-solving, logical reasoning, tool usage, API calls, and coding. Recent studies (Yang et al., 2023c; Yan et al., 2023) have shown that GPT-4V can understand various types of images, including simple user interfaces (UIs) in popular smartphone apps. However, challenges arise when the apps are new and their UIs are less typical, which highlights a major problem that our work aims to address. Among open-source efforts from the industry and research community, the LLaMA series (Touvron et al., 2023a, b) are the most popular equivalents and have been fine-tuned to acquire conversational abilities, employing a decoder-only architecture similar to ChatGPT (Taori et al., 2023; Zheng et al., 2023). Building upon LLaMA, many multimodal LLMs, such as LLaVA (Liu et al.,

ChatGPT(OpenAI，2021) 和 GPT-4(OpenAI，2023) 的发展标志着自然语言处理领域的重要进展。与早期大型语言模型 (LLMs) 不同，这些新模型 (Touvron 等，2023a, b；Zeng 等，2022；Taori 等，2023；Zheng 等，2023) 支持多轮对话，并具备遵循复杂指令的强大能力。GPT-4V(Yang 等，2023b) 中视觉能力的集成是又一里程碑，使语言模型能够处理和理解视觉数据。这一扩展拓宽了 AI 应用的范围，使 GPT-4 能够执行诸如问题解决、逻辑推理、工具使用、API 调用和编程等多样任务。近期研究 (Yang 等，2023c；Yan 等，2023) 表明，GPT-4V 能够理解多种类型的图像，包括流行智能手机应用中的简单用户界面 (UI)。然而，当应用较新且其 UI 不典型时，仍存在挑战，这正是我们工作旨在解决的主要问题。在业界和研究社区的开源努力中，LLaMA 系列 (Touvron 等，2023a, b) 是最受欢迎的同类模型，经过微调以获得对话能力，采用与 ChatGPT 类似的仅解码器架构 (Taori 等，2023；Zheng 等，2023)。基于 LLaMA，许多多模态大型语言模型，如 LLaVA(Liu 等，

2023b, a), ChartLlama (Han et al., 2023), and Sta-bleLLaVA (Li et al., 2023), also demonstrate vision understanding capabilities akin to those of GPT-4V. Nevertheless, a performance gap persists between these open-source

models and GPT-4V, suggesting potential areas for further development.

2023b, a)、ChartLlama(Han 等，2023) 和 StableLLaVA(Li 等，2023)，也展示了类似于 GPT-4V 的视觉理解能力。然而，这些开源模型与 GPT-4V 之间仍存在性能差距，表明未来仍有改进空间。

## 2.2 LLMs as agents

## 2.2 作为代理的大型语言模型

The use of LLMs as agents for executing complex tasks has gained increasing attention. Initiatives like AutoGPT (Yang et al., 2023a), Hug-gingGPT (Shen et al., 2023), and MetaGPT (Hong et al., 2023) illustrate this trend, and many projects demonstrate impressive capabilities, moving beyond basic language tasks to engaging in activities requiring higher cognitive functions, such as software development (Qian et al., 2023; Chen et al., 2021) and gaming (FAIR et al., 2022; Park et al., 2023; Xu et al., 2023). In this context, Yao et al. (Yao et al., 2023) introduce an innovative approach that synergizes reasoning and acting in LLMs, significantly enhancing their decision-making and interactive capabilities. LLM-based agents are designed to utilize the advanced language and reasoning skills of LLMs to interact with and manipulate their environment (Liu et al., 2023c; Gur et al., 2023; Xie et al., 2023). This includes performing tasks that require understanding context, making decisions, and learning from interactions (Xi et al., 2023; Hu and Shu, 2023). Such agents are pivotal in applications where human-like cognitive abilities are essential.

将大型语言模型 (LLMs) 作为执行复杂任务的代理的应用日益受到关注。诸如 AutoGPT(Yang 等，2023a)、HuggingGPT(Shen 等，2023) 和 MetaGPT(Hong 等，2023) 等项目体现了这一趋势，许多研究展示了令人印象深刻的能力，超越了基础语言任务，参与需要更高认知功能的活动，如软件开发 (Qian 等，2023；Chen 等，2021) 和游戏 (FAIR 等，2022；Park 等，2023；Xu 等，2023)。在此背景下，Yao 等 (Yao 等，2023) 提出了一种创新方法，将推理与行动在大型语言模型中协同，显著提升了其决策和交互能力。基于大型语言模型的代理旨在利用其先进的语言和推理技能与环境进行交互和操作 (Liu 等，2023c；Gur 等，2023；Xie 等，2023)，包括执行需要理解上下文、做出决策及从交互中学习的任务 (Xi 等，2023；Hu 和 Shu，2023)。此类代理在需要类人认知能力的应用中具有关键作用。

The emergence of multimodal LLM agents (Wang et al., 2023; Furuta et al., 2023; Brohan et al., 2022, 2023; Reed et al., 2022), capable of processing various inputs including text, images, audio, and video, has further broadened the scope of LLM applications. This versatility is particularly beneficial for LLM-based agents, enabling them to interact more effectively with their environment and complete more complex tasks, be it completing household tasks in a physical world (Ahn et al., 2022), generating 3D assets via procedural tool use (Sun et al., 2023), or mastering over 600 tasks across different domains at the same time (Reed et al., 2022). Our research contributes to this area by focusing on an agent designed to operate smartphone applications. This agent's ability to interpret screenshots from the operating system demonstrates its flexibility and adaptability, making it a valuable tool in a wide

多模态大型语言模型代理的出现 (Wang 等，2023；Furuta 等，2023；Brohan 等，2022，2023；Reed 等，2022)，能够处理包括文本、图像、音频和视频在内的多种输入，进一步拓宽了大型语言模型的应用范围。这种多样性对基于大型语言模型的代理尤为有利，使其能够更有效地与环境交互并完成更复杂的任务，无论是在物理世界中完成家务 (Ahn 等，2022)、通过程序化工具生成三维资产 (Sun 等，2023)，还是同时掌握 600 多项跨领域任务 (Reed 等，2022)。我们的研究聚焦于设计一个能够操作智能手机应用的代理。该代理解读操作系统截图的能力展示了其灵活性和适应性，使其成为广泛应用中的宝贵工具。

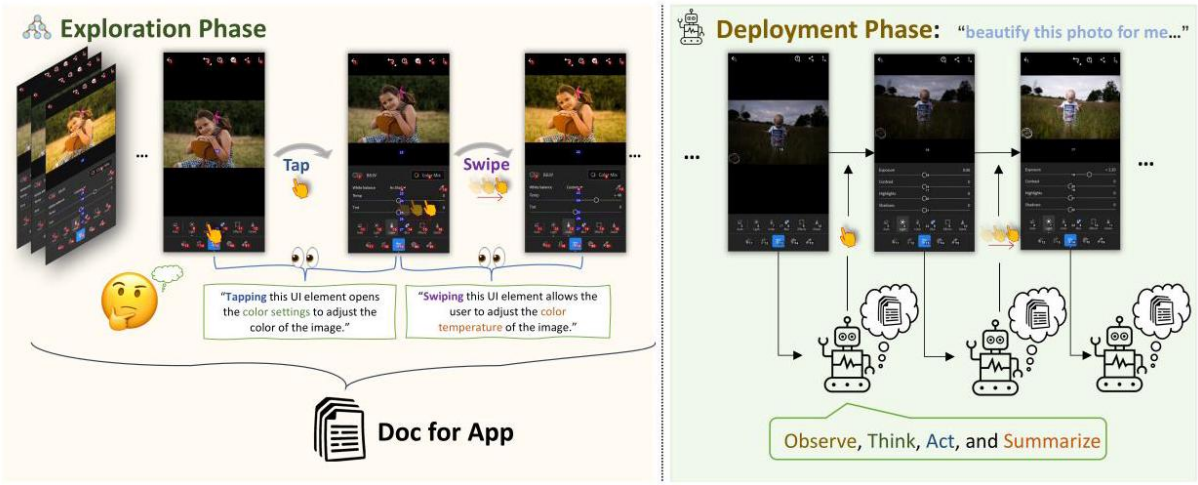


Figure 2: Overview of our multimodal agent framework designed to operate smartphone applications. The figure illustrates the two-phase approach of our framework. In the exploration phase, the agent interacts with a smartphone application and learns from their outcomes to create a comprehensive reference document. In the deployment phase, the agent utilizes the information compiled in this document to operate and navigate the apps effectively.

图 2: 我们设计的用于操作智能手机应用的多模态代理框架概览。图中展示了框架的两阶段方法。在探索阶段，代理与智能手机应用交互并从结果中学习，以创建详尽的参考文档。在部署阶段，代理利用该文档中汇总的信息，有效地操作和导航应用程序。

range of applications.

广泛的应用领域。

# 3 Method

# 3 方法

This section details the methodology behind our innovative multimodal agent framework. This framework enables an agent to interact with smartphone applications in a manner akin to human behavior. We first describe the experimental environment and action space, which are foundational elements of our system. Next, we discuss the exploration phase, where the agent learns app functionalities either through autonomous interactions or by

observing human demonstrations. Finally, we outline the deployment phase, explaining how the agent applies its acquired knowledge to execute high-level tasks.

本节详细介绍我们创新多模态代理框架的方法论。该框架使代理能够以类似人类的方式与智能手机应用交互。我们首先描述实验环境和动作空间，这些是系统的基础要素。接着讨论探索阶段，代理通过自主交互或观察人类示范学习应用功能。最后，概述部署阶段，说明代理如何应用所获知识执行高级任务。

## 3.1 Environment and Action Space

## 3.1 环境与动作空间

Experimental Environment: Our experimental environment is built on a command-line interface (CLI), allowing the agent to interact with smart-phone apps. We chose the Android operating system for our experiments. The agent receives two key inputs: a real-time screenshot showing the app's interface and an XML file detailing the interactive elements. To enhance the agent's ability to identify and interact with these elements seamlessly, we assign each element a unique identifier. These identifiers are derived either from the resource ID in the XML file (if provided) or are constructed by combining the class name, size, and content of the element. These elements are overlaid as semi-transparent numbers on the screenshot.

实验环境: 我们的实验环境基于命令行界面 (CLI), 允许代理与智能手机应用交互。实验选用 Android 操作系统。代理接收两个关键输入: 显示应用界面的实时截图和描述交互元素的 XML 文件。为增强代理识别和操作这些元素的能力，我们为每个元素分配唯一标识符。该标识符来源于 XML 文件中的资源 ID(若有)，否则通过组合元素的类名、大小和内容构建。这些元素以半透明数字形式覆盖在截图上。

This helps the agent to interact accurately without needing to specify exact positions on the screen and enhances the agent's precision in controlling the phone.

这帮助代理准确交互，无需指定屏幕上的精确位置，提升了代理对手机的控制精度。

Action Space: Our agent's action space mirrors common human interactions with smartphones: taps and swipes. We designed four basic functions:

动作空间: 代理的动作空间模拟人类与智能手机的常见交互: 点击和滑动。我们设计了四个基本功能:

- Tap(element : int) : This function simulates a tap on the UI element numbered on the screen. For example, tap(5) would tap the element labeled '5'.

    - Tap(element : int) : 该功能模拟对屏幕上编号的 UI 元素的点击。例如，tap(5) 表示点击标记为“5”的元素。

- Long_press(element : int) : This function emulates a long press (for 1 second) on a UI element.

- Long_press(element : int) : 该功能模拟对 UI 元素的长按 (持续 1 秒)。

- Swipe ( element : int, direction : str, dist : str): It allows the agent to swipe on an element in a specified direction (up, down, left, right) and distance (short, medium, long). For instance, swipe(21, ”up”, ”medium”) would swipe up on element ’21’ for a medium distance.

- Swipe(element : int, direction : str, dist : str): 允许代理在指定方向 (上、下、左、右) 和距离 (短、中、长) 上对元素进行滑动。例如，swipe(21, ”up”, ”medium”) 表示对编号为“21”的元素向上滑动中等距离。

- Text(text : str) : To bypass inefficient virtual keyboard typing, this function inputs text directly into an input field when a virtual keyboard is visible. For example, text(”Hello, world!”) inputs the string ”Hello, world!”.

- Text(text : str) : 为绕过低效的虚拟键盘输入，当虚拟键盘可见时，该函数直接将文本输入到输入框中。例如，text(”Hello, world!”) 会输入字符串“Hello, world!”。

- Back() : A system-level function that helps the agent return to the previous UI page, especially useful for exiting irrelevant pages.

- Back() : 一个系统级函数，帮助代理返回到上一个界面，特别适用于退出无关页面。

- Exit() : A specialized function is employed to conclude processes, typically invoked upon successful task completion.

- Exit() : 一个专用函数，用于结束进程，通常在任务成功完成时调用。

These predefined actions are designed to simplify the agent’s interactions, particularly by eliminating the need for precise screen coordinates, which can pose challenges for language models in accurately predicting.

这些预定义动作旨在简化代理的交互，尤其是避免对精确屏幕坐标的依赖，因为语言模型在准确预测坐标时存在挑战。

## 3.2 Exploration Phase

## 3.2 探索阶段

Exploring by autonomous interactions. The Exploration Phase is central to our framework. Here, the agent learns about the functionalities and features of smartphone apps through trial and error. In this phase, the agent is assigned a task and starts interacting autonomously with the UI elements. It uses different actions and observes the resulting changes in the app interface to understand how it works. The agent, driven by a large language model, attempts to figure out the functions of UI elements and the effects of specific actions by analyzing screenshots before and after each action. This information is compiled into a document that records the effects of actions applied to different elements. When a UI element is acted upon multiple times, the agent will update the document based on past documents and current observations to improve quality. To make exploration more efficient, the agent stops further exploring UI elements if the current UI page seems unrelated to the main tasks of the app, like advertisement

pages. In such cases, it uses the Android system's Back() function to return to the previous UI page. Compared with random exploration, such as Depth-First Search and Breadth-First Search, this goal-oriented exploration approach ensures that the agent focuses on elements crucial for the effective operation of the app. The agent also utilizes the LLM's existing knowledge about user interfaces to improve exploration efficiency. The exploration stops when the agent completes the assigned task.

通过自主交互进行探索。探索阶段是我们框架的核心。在此阶段，代理通过反复试验学习智能手机应用的功能和特性。代理被分配任务后，开始自主与界面元素交互，使用不同动作并观察应用界面的变化以理解其工作原理。由大型语言模型驱动的代理，通过分析每次动作前后的截图，尝试推断界面元素的功能及特定动作的效果。相关信息被整理成文档，记录不同元素上动作的影响。当某个界面元素被多次操作时，代理会基于历史文档和当前观察更新文档以提升质量。为提高探索效率，若当前界面页面与应用主要任务无关 (如广告页面)，代理会停止进一步探索，使用 Android 系统的 Back() 函数返回上一界面。相比随机探索 (如深度优先搜索和广度优先搜索)，这种目标导向的探索方法确保代理聚焦于应用有效运行的关键元素。代理还利用大型语言模型对用户界面的既有知识提升探索效率。探索在代理完成分配任务时结束。

Exploring by watching demos. An alternative and often more effective exploration method involves the agent observing human demonstrations. These demonstrations provide the agent with examples of efficient app usage, especially for understanding complex functionalities that might be challenging to discover through autonomous interactions. In this method, a human user operates the apps while the agent observes, recording only the elements and actions employed by the human.

通过观看演示进行探索。另一种通常更有效的探索方法是代理观察人类演示。这些演示为代理提供了高效使用应用的示例，尤其有助于理解通过自主交互难以发现的复杂功能。在此方法中，人类用户操作应用，代理观察并仅记录人类使用的元素和动作。

This strategy narrows down the exploration space and prevents the agent from engaging with irrelevant app pages, making it a more streamlined and efficient approach compared to autonomous interactions.

该策略缩小了探索空间，避免代理进入无关应用页面，使其比自主交互更简洁高效。

## 3.3 Deployment Phase

## 3.3 部署阶段

Following the exploration phase, the agent is well-equipped to execute complex tasks based on its accrued experience. The agent adheres to a step-by-step approach when given a task, with each step encompassing access to a screenshot of the current UI and a dynamically generated document detailing the functions of UI elements and the actions' effects on the current UI page. The prompts also provide detailed explanations of all available actions. In each step, the agent is first tasked with providing its observations of the current UI, followed by articulating its thought process concerning the task and current observations. Subsequently, the agent proceeds to execute actions by invoking available functions. After each action, the agent summarizes the interaction history and the actions taken during the current step. This information is incorporated into the next prompt, which provides the agent with a form of memory. This meticulous approach enhances the reliability and interpretability of the agent's actions,

thereby facilitating more informed decision-making. The deployment phase stops when the agent determines that the task has been accomplished, at which point it can exit the process by taking the Exit() action.

探索阶段结束后，代理积累了丰富经验，能够执行复杂任务。代理在执行任务时遵循逐步方法，每一步包括获取当前界面的截图和动态生成的文档，文档详细描述界面元素功能及动作对当前界面的影响。提示中还提供所有可用动作的详细说明。每一步，代理首先提供对当前界面的观察，然后阐述其对任务和当前观察的思考过程，随后通过调用可用函数执行动作。每次动作后，代理总结交互历史和当前步骤中采取的动作，将信息纳入下一次提示，为代理提供记忆支持。这种细致方法提升了代理动作的可靠性和可解释性，促进更明智的决策。部署阶段在代理判断任务完成时结束，此时可通过 Exit() 动作退出流程。

# 4 Experiments

## 4 实验

In this section, we will present our evaluation of the multimodal agent framework through a combination of quantitative and qualitative experiments. Our primary goal is to assess the agent's performance and its ability to operate a diverse set of smartphone applications effectively.

本节将通过定量和定性实验评估多模态代理框架。我们的主要目标是评估代理的性能及其有效操作多样智能手机应用的能力。

## 4.1 Experimental Setup

## 4.1 实验设置

To comprehensively evaluate our method, we construct a benchmark that includes 10 popular applications, each serving various purposes. These applications include Google Maps, Twitter, Telegram, YouTube, Spotify, Yelp, Gmail, TEMU, Clock, and Lightroom. We have intentionally chosen this diverse set of apps to test the agent's adaptability across various functions and interfaces. In particular, to gain a more comprehensive insight into the

为全面评估我们的方法，我们构建了包含 10 款流行应用的基准测试，这些应用涵盖多种用途。包括 Google Maps(谷歌地图)、Twitter、Telegram、YouTube、Spotify、Yelp、Gmail、TEMU、Clock(时钟) 和 Lightroom。我们有意选择这组多样化应用，以测试代理在不同功能和界面上的适应能力。特别是，为了获得更全面的洞察，

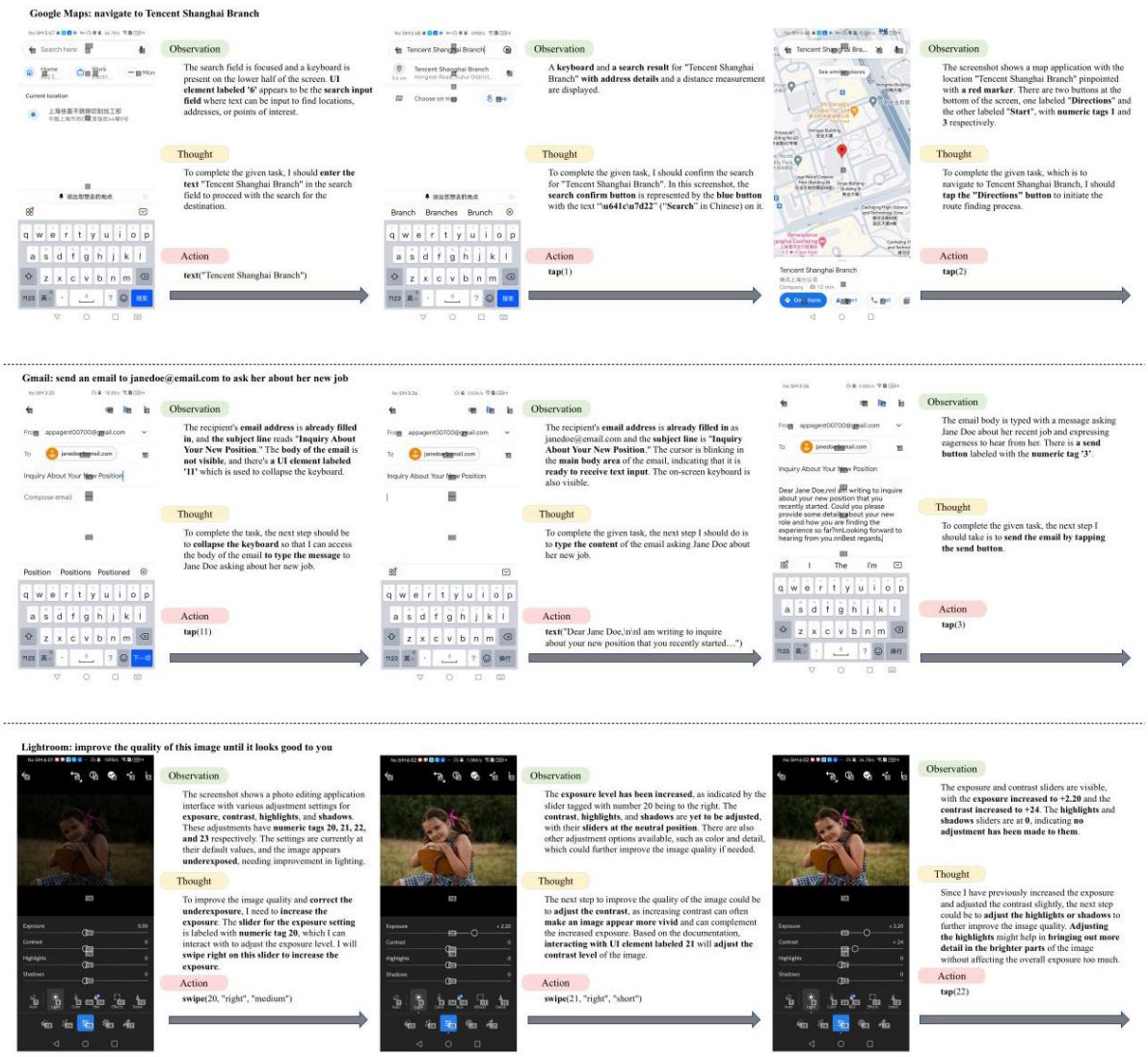


Figure 3: Qualitative Task Evaluation Across Three Apps. This figure presents qualitative results for three distinct tasks conducted on Google Maps, Gmail, and Lightroom. It showcases AppAgent's ability to accurately perceive, reason, and execute tasks, demonstrating its competence in various application contexts. Due to space constraints, some less critical details have been omitted from the description.

图 3: 三款应用的定性任务评估。该图展示了在 Google Maps、Gmail 和 Lightroom 上进行的三项不同任务的定性结果，体现了 AppAgent 准确感知、推理和执行任务的能力，展示其在多种应用场景中的胜任力。由于篇幅限制，描述中省略了一些次要细节。

vision capabilities of our agent, we conducted an in-depth case study using Adobe Lightroom, an image-editing application. This specific case study allowed us to evaluate the agent's proficiency in handling visual tasks and its ability to interpret and manipulate images within the app. For the exploration phase, we capped the maximum number of steps at 40. During testing, we limited the maximum number of steps to 10. For these experiments, we utilized the state-of-the-art multimodal large language model, GPT-4. GPT-4 is equipped to process interleaved image-and-text inputs effectively. This unique capability enables our agent to interpret and interact with both visual and textual information seamlessly within the applications.

为了评估我们代理的视觉能力，我们使用图像编辑应用 Adobe Lightroom 进行了深入的案例研究。该案例研究使我们能够评估代理处理视觉任务的熟练度及其在应用内解释和操作图像的能力。在探索阶段，我们将最大步骤数限制为 40 步。测试阶段则将最大步骤数限制为 10 步。实验中，我们采用了最先进的多模态大型语言模型 GPT-4。GPT-4 能够有效处理交错的图像与文本输入。这一独特能力使我们的代理能够在应用中无缝解读并交互视觉与文本信息。

## 4.2 Design and Analysis

## 4.2 设计与分析

Baselines. To comprehensively evaluate our multimodal agent framework, we considered various design choices and their impact on performance. We conducted experiments using different configurations to provide valuable insights into the agent's behavior. We started with GPT-4 without any reference documents during testing and examined its performance both with the raw action API and our simplified action space. Next, we explored different ways to generate guiding documents for the agent. These included documents generated through autonomous exploration, watching human demonstrations, and the manually crafted document as an oracle benchmark.

基线方法。为了全面评估我们的多模态代理框架，我们考虑了多种设计选择及其对性能的影响。通过不同配置的实验，我们获得了关于代理行为的宝贵见解。首先，我们在测试时使用不带任何参考文档的 GPT-4，考察其在原始动作 API 和我们简化动作空间下的表现。随后，我们探索了为代理生成指导文档的不同方式，包括通过自主探索生成的文档、观看人类示范生成的文档，以及作为基准的手工制作文档。

To effectively compare the performance of different methods, we employed three key metrics:

为了有效比较不同方法的性能，我们采用了三个关键指标:

| Method | Document | Action Space | SR↑ | Reward ↑ | Avg. Steps |
|---|---|---|---|---|---|
| GPT4 (Baseline) | None | Raw | 2.2% | 0.6 | 4.0 |
| | None | Ours | 48.9% | 3.5 | 6.9 |
| AppAgent | Auto. Exploration | Ours | 73.3% | 5.1 | 4.4 |
| | Watching Demos | Ours | 84.4% | 4.7 | 5.1 |
| | Manually Crafted | Ours | 95.6% | 5.5 | 5.5 |

| 方法 | 文档 | 动作空间 | 成功率 ↑ | 奖励 ↑ | 平均步数 |
|---|---|---|---|---|---|
| GPT4(基线) | 无 | 原始 | 2.2% | 0.6 | 4.0 |
| | 无 | 我们的 | 48.9% | 3.5 | 6.9 |
| 应用代理 | 自动探索 | 我们的 | 73.3% | 5.1 | 4.4 |
| | 观看演示 | 我们的 | 84.4% | 4.7 | 5.1 |
| | 手工设计 | 我们的 | 95.6% | 5.5 | 5.5 |

Table 1: Evaluating Design Choices in AppAgent Performance. This table contrasts different design elements within AppAgent. Key findings include: our custom-developed action space surpasses the raw action space in

efficiency; the exploration phase, incorporating both autonomous interaction and observation of human demonstrations, significantly enhances agent performance; and the auto-generated documentation yields outcomes on par with those derived from manually crafted documents.

表 1:AppAgent 性能设计选择评估。该表对 AppAgent 中的不同设计元素进行了对比。主要发现包括: 我们自定义开发的动作空间在效率上优于原始动作空间；探索阶段结合自主交互和人类示范观察，显著提升了代理性能；自动生成的文档效果与手工制作的文档相当。

| Method | Document | Action Space | Avg. Rank ↓ | Num. Tools |
|---|---|---|---|---|
| GPT4 (Baseline) | None | Ours | 2.30 | 2.4 |
| AppAgent | Watching Demos | Ours | 1.95 | 5.8 |
| | Manually Crafted | Ours | 1.75 | 4.0 |

| 方法 | 文档 | 动作空间 | 平均排名 ↓ | 工具数量 |
|---|---|---|---|---|
| GPT4(基线) | 无 | 我们的 | 2.30 | 2.4 |
| 应用代理 | 观看演示 | 我们的 | 1.95 | 5.8 |
| | 手工制作 | 我们的 | 1.75 | 4.0 |

Table 2: Case study on image editing tasks with Lightroom App. We conduct a user study to rank the image editing results of different methods. Our agents produce better results than the GPT-4 baseline.

表 2: 使用 Lightroom 应用进行图像编辑任务的案例研究。我们进行了用户研究，对不同方法的图像编辑结果进行排名。我们的代理产生的结果优于 GPT-4 基线。

Successful Rate (SR): This metric measures the average rate at which the agent successfully completes tasks within an app. If the agent fails to finish the task in 10 steps, it is considered a failure.

成功率 (SR): 该指标衡量代理在应用内成功完成任务的平均比例。如果代理在 10 步内未完成任务，则视为失败。

Reward: To provide a more fine-grained measurement, we developed a reward model to assess performance. For each task within an app, we scored different UI pages. The closer the UI page was to the objective, the higher the score received. This means that even if the agent failed to complete the task, it would still receive credit based on its final state.

奖励: 为了提供更细粒度的评估，我们开发了一个奖励模型来衡量性能。对于应用内的每个任务，我们对不同的 UI 页面进行了评分。UI 页面越接近目标，得分越高。这意味着即使代理未能完成任务，也会根据其最终状态获得相应的积分。

Average Steps: We also reported the average number of steps required to successfully finish tasks across the selected applications.

平均步骤数: 我们还报告了在所选应用中成功完成任务所需的平均步骤数。

Results. The comparison of our experimental results is presented in Table 1. We report the average performance of 45 tasks on 9 of the 10 previously described apps. Notably, we excluded Lightroom from this evaluation,

as assessing task completion in this application presented inherent ambiguities. As demonstrated, our simplified action space significantly improves the performance of the GPT-4 baseline. Our observations indicate that LLM struggles with producing accurate xy coordinates, while our simplified action space eliminates this challenging requirement. Additionally, documents generated through autonomous exploration and observ-

结果。我们的实验结果比较见表 1。我们报告了之前描述的 10 个应用中 9 个应用上 45 个任务的平均表现。值得注意的是，我们排除了 Lightroom 的评估，因为该应用中任务完成的评估存在固有的模糊性。如所示，我们简化的动作空间显著提升了 GPT-4 基线的性能。观察表明，大型语言模型 (LLM) 在生成准确的 xy 坐标方面存在困难，而我们简化的动作空间消除了这一挑战。此外，通过自主探索和观察人类演示生成的文档证明了其高效性。

ing human demonstrations proved to be highly effective. Their results consistently outperformed the GPT-4 baseline and are comparable to the results of human-written documents, which highlights the efficacy of our design in enhancing the agent's performance across a diverse set of applications.

其结果持续优于 GPT-4 基线，并且与人工编写的文档结果相当，凸显了我们设计在提升代理在多样化应用中表现的有效性。

Qualitative results. In Fig. 3, we provide examples showcasing the agent's execution process for various tasks. This qualitative analysis serves to demonstrate the agent's capacity to accurately perceive, reason, and act in response to given tasks. For a more comprehensive understanding of our agent's capabilities, please refer to our project page, which includes additional demonstration videos.

定性结果。在图 3 中，我们展示了代理执行各种任务的过程示例。该定性分析旨在展示代理准确感知、推理和响应任务的能力。欲全面了解代理能力，请参阅我们的项目页面，其中包含更多演示视频。

## 4.3 Case Study

## 4.3 案例研究

To gain deeper insights into the vision capabilities of our agent, we conducted an extensive case study using Adobe Lightroom, an image-editing application. This specific case study allowed us to evaluate the agent's proficiency in handling visual tasks, which was previously impossible for text-only agent models. Lightroom, as an image-editing app with various editing tools, demands a wide range of operations, such as selecting appropriate tools and manipulating image parameters. This case study provides a robust evaluation of the agent's overall capabilities. Additionally, the open-ended nature of image editing tasks allows us to

为了深入了解代理的视觉能力，我们使用图像编辑应用 Adobe Lightroom 进行了广泛的案例研究。该案例研究使我们能够评估代理处理视觉任务的熟练度，这在之前仅限文本的代理模型中是不可能的。Lightroom 作为一款具备多种编辑工具的图像编辑应用，要求执行诸如选择合适工具和调整图像参数等多样操作。该案例研究为代理整体能力提供了有力评估。此外，图像编辑任务的开放性使我们能够评估代理的问题解决能力。

assess the agent's problem-solving abilities. We prepared five images with visual issues, such as low contrast

and overexposure. Various variants of our model, as previously illustrated, were used to edit these images. A user study was conducted to rank the editing results produced by different methods. We also reported the average number of tools used for image editing, providing an additional reference to the editing process's complexity. All models were assigned the task of "fix this image until it looks good to you" without specifying the image's problems. The comparison of the results is presented in Table 2. As we can see, our agent model with documents yields consistently better results than the GPT-4 baseline, which emphasizes the influence of documents in our design. The generated documents by watching the demonstration produced comparable results with the results of manually crafted documents, which suggests the effectiveness of the exploration phase. We also find that with a document, the agent tends to use various tools to improve the image quality, while the GPT-4 baseline uses fewer tools.

我们准备了五张存在视觉问题的图像，如低对比度和过曝。使用之前展示的多种模型变体对这些图像进行编辑。我们进行了用户研究，对不同方法产生的编辑结果进行排名。我们还报告了图像编辑中使用工具的平均数量，为编辑过程的复杂性提供了额外参考。所有模型均被赋予“修复此图像直到你觉得满意”的任务，未具体说明图像问题。结果比较见表 2。正如所见，带文档的代理模型结果持续优于 GPT-4 基线，强调了文档在我们设计中的作用。通过观看演示生成的文档产生的结果与手工制作文档的结果相当，表明探索阶段的有效性。我们还发现，有文档时，代理倾向于使用多种工具提升图像质量，而 GPT-4 基线使用的工具较少。

# 5 Conclusion

# 5 结论

In this paper, we have introduced a novel multimodal agent framework that leverages the vision capabilities of large language models to operate smartphone applications in a human-like manner. Our approach eliminates the need for system back-end access and offers security, adaptability, and flexibility advantages. Our exploration-based learning strategy allows the agent to quickly adapt to new applications with unfamiliar user interfaces, making it a versatile tool for various tasks. Our extensive experiments across various apps highlight our agent's ability to handle diverse high-level tasks and underscore its adaptability and learning efficiency.

本文提出了一种新颖的多模态代理框架，利用大型语言模型 (LLM) 的视觉能力以类人方式操作智能手机应用。我们的方法无需系统后端访问，具备安全性、适应性和灵活性优势。基于探索的学习策略使代理能够快速适应具有陌生用户界面的新应用，成为多任务的通用工具。我们在多款应用上的广泛实验展示了代理处理多样高阶任务的能力，强调了其适应性和学习效率。

Limitation. We have adopted a simplified action space for smartphone operations, which means that advanced controls such as multi-touch and irregular gestures are not supported. This limitation may restrict the agent's applicability in some challenging scenarios. Nevertheless, we recognize this as an avenue for future research and development.

局限性。我们采用了简化的智能手机操作动作空间，因此不支持多点触控和不规则手势等高级控制。这一限制可能限制代理在某些复杂场景中的适用性。但我们认为这是未来研究和开发的方向。

## References

## 参考文献


Michael Ahn, Anthony Brohan, Noah Brown, Yev-gen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol

Michael Ahn, Anthony Brohan, Noah Brown, Yev-gen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol

Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jes-month, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do as i can and not as i say: Grounding language in robotic af-fordances. In arXiv preprint arXiv:2204.01691.

Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jes-month, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. 按我所能而非我所言: 将语言基础植入机器人可供性 (affordances)。发表于 arXiv 预印本 arXiv:2204.01691。

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, 等. 2023. RT-2: 视觉-语言-动作模型将网络知识迁移至机器人控制。arXiv 预印本 arXiv:2307.15818。

Anthony Brohan, Noah Brown, Justice Carbajal, Yev-gen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. 2022. Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817.

Anthony Brohan, Noah Brown, Justice Carbajal, Yev-gen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, 等. 2022. RT-1: 面向大规模现实控制的机器人变换器 (transformer)。arXiv 预印本 arXiv:2212.06817。

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, 等. 2021. 评估基于代码训练的大型语言模型。arXiv 预印本 arXiv:2107.03374。

Meta FAIR, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. 2022. Human-level play in the game of diplomacy by combining language models with strategic reasoning. Science, 378(6624):1067-1074.

Meta FAIR, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, 等. 2022. 通过结合语言模型与战略推理，实现外交游戏中的人类水平对弈。科学，378(6624):1067-1074。

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. 2023. Multimodal web navigation with instruction-finetuned foundation models.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, 和 Izzeddin Gur. 2023. 基于指令微调基础模型的多模态网页导航。

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksan-dra Faust. 2023. A real-world webagent with planning, long context understanding, and program synthesis.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, 和 Aleksandra Faust. 2023. 具备规划、长上下文理解及程序合成能力的现实世界网页代理。

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. Chartllama: A multimodal llm for chart understanding and generation.

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, 和 Hanwang Zhang. 2023. Chartllama: 用于图表理解与生成的多模态大型语言模型 (LLM)。

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2023. Metagpt: Meta programming for a multi-agent collaborative framework.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, 和 Jürgen Schmidhuber. 2023. MetaGPT: 面向多智能体协作框架的元编程。

Zhiting Hu and Tianmin Shu. 2023. Language models, agent models, and world models: The law for machine reasoning and planning. arXiv preprint arXiv:2312.05230.

Zhiting Hu 和 Tianmin Shu. 2023. 语言模型、智能体模型与世界模型: 机器推理与规划的法则。arXiv 预印本 arXiv:2312.05230。

Yanda Li, Chi Zhang, Gang Yu, Zhibin Wang, Bin Fu, Guosheng Lin, Chunhua Shen, Ling Chen, and Yunchao Wei. 2023. Stablellava: Enhanced visual instruction tuning with synthesized image-dialogue data.

Yanda Li, Chi Zhang, Gang Yu, Zhibin Wang, Bin Fu, Guosheng Lin, Chunhua Shen, Ling Chen, 和 Yunchao Wei. 2023. StableLLaVA: 通过合成图像-对话数据增强视觉指令微调。

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Yuheng Li, 和 Yong Jae Lee. 2023a. 通过视觉指令微调改进基线模型。

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, 和 Yong Jae Lee. 2023b. 视觉指令微调。

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Ao-han Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023c. Agent-Bench: Evaluating LLMs as agents. arXiv preprint arXiv: 2308.03688.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Ao-han Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, 和 Jie Tang. 2023c. Agent-Bench: 评估大型语言模型 (LLMs) 作为代理的性能。arXiv 预印本 arXiv: 2308.03688。

OpenAI. 2021. Chatgpt. https://openai.com/ research/chatgpt.

OpenAI. 2021. ChatGPT. https://openai.com/research/chatgpt。

OpenAI. 2023. Gpt-4 technical report.

OpenAI. 2023. GPT-4 技术报告。

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, pages 1-22.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, 和 Michael S Bernstein. 2023. 生成代理: 人类行为的交互式模拟。在第 36 届年度 ACM 用户界面软件与技术研讨会论文集，页码 1-22。

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. arXiv preprint arXiv:2307.07924.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, 和 Maosong Sun. 2023. 用于软件开发的交互式代理。arXiv 预印本 arXiv:2307.07924。

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. 2022. A generalist agent. arXiv preprint arXiv:2205.06175.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg 等. 2022. 通用代理。arXiv 预印本 arXiv:2205.06175。

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. In Advances in Neural Information Processing Systems.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, 和 Yueting Zhuang. 2023. HuggingGPT: 利用 ChatGPT 及其伙伴在 HuggingFace 上解决 AI 任务。在神经信息处理系统进展中。

Chunyi Sun, Junlin Han, Weijian Deng, Xinlong Wang, Zishan Qin, and Stephen Gould. 2023. 3d-gpt: Procedural 3d modeling with large language models. arXiv preprint arXiv:2310.12945.

Chunyi Sun, Junlin Han, Weijian Deng, Xinlong Wang, Zishan Qin, 和 Stephen Gould. 2023. 3D-GPT: 基于大型语言模型的程序化三维建模。arXiv 预印本 arXiv:2310.12945。

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https:// github.com/tatsu-lab/stanford_alpaca.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, 和 Tatsunori B. Hashimoto. 2023. Stanford Alpaca: 一个遵循指令的 LLaMA 模型。https://github.com/tatsu-lab/stanford_alpaca。

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, 和 Guillaume Lample. 2023a. LLaMA: 开放且高效的基础语言模型。

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Moly-bog, Yixin Nie, Andrew Poulton, Jeremy Reizen-stein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subrama-nian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Moly-bog, Yixin Nie, Andrew Poulton, Jeremy Reizen-stein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subrama-nian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: 开放基础模型与微调聊天模型。

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jin-bing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, Xiaojian Ma, and Yitao Liang. 2023. Jarvis-1: Open-world multitask agents with memory-augmented multimodal language models.

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jin-bing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, Xiaojian Ma, and Yitao Liang. 2023. Jarvis-1: 具备记忆增强多模态语言模型的开放世界多任务代理。

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. arXiv preprint arXiv:2309.07864.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, 等. 2023. 基于大型语言模型代理的兴起与潜力: 一项综述。arXiv 预印本 arXiv:2309.07864。

Tianbao Xie, Fan Zhou, Zhoujun Cheng, Peng Shi, Lu-oxuan Weng, Yitao Liu, Toh Jing Hua, Junning Zhao, Qian Liu, Che Liu, Leo Z. Liu, Yiheng Xu, Hongjin Su, Dongchan Shin, Caiming Xiong, and Tao Yu. 2023. Openagents: An open platform for language agents in the wild.

Tianbao Xie, Fan Zhou, Zhoujun Cheng, Peng Shi, Lu-oxuan Weng, Yitao Liu, Toh Jing Hua, Junning Zhao, Qian Liu, Che Liu, Leo Z. Liu, Yiheng Xu, Hongjin Su, Dongchan Shin, Caiming Xiong, 和 Tao Yu. 2023. Openagents: 一个面向实际应用的语言代理开放平台。

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xi-aolong Wang, Weidong Liu, and Yang Liu. 2023. Exploring large language models for communication games: An empirical study on werewolf. arXiv preprint arXiv:2309.04658.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xi-aolong Wang, Weidong Liu, 和 Yang Liu. 2023. 探索大型语言模型在交流游戏中的应用: 狼人杀的实证研究。arXiv 预印本 arXiv:2309.04658。

An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, Zicheng Liu, and Lijuan Wang. 2023. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. arXiv preprint arXiv: 2311.07562.

An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, Zicheng Liu, 和 Lijuan Wang. 2023. GPT-4V 奇境: 用于零样本智能手机 GUI 导航的大型多模态模型。arXiv 预印本 arXiv:2311.07562。

Hui Yang, Sifu Yue, and Yunzhong He. 2023a. Auto-gpt for online decision making: Benchmarks and additional opinions.

Hui Yang, Sifu Yue, 和 Yunzhong He. 2023a. Auto-GPT 用于在线决策: 基准测试与额外观点。

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023b. The dawn of lmms: Preliminary explorations with gpt-4v (ision). arXiv preprint arXiv:2309.17421.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, 和 Lijuan Wang. 2023b. LMMs 的曙光: 基于 GPT-4V(视觉) 的初步探索。arXiv 预印本 arXiv:2309.17421。

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023c. The dawn of lmms: Preliminary explorations with gpt-4v(ision). arXiv preprint arXiv: 2309.17421.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, 和 Lijuan Wang. 2023c. lMM 的曙光: 基于 GPT-4V(视觉) 的初步探索。arXiv 预印本 arXiv:2309.17421。

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In ICLR.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, 和 Yuan Cao. 2023. ReAct: 在语言模型中协同推理与行动。发表于 ICLR 会议。

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. arXiv preprint arXiv:2210.02414.

曾奥涵，刘晓，杜正孝，王子涵，赖涵宇，丁明，杨卓毅，徐一帆，郑文迪，夏晓，等。2022 年。Glm-130b: 一个开放的双语预训练模型。arXiv 预印本 arXiv:2210.02414。

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

郑连民，蒋伟林，盛颖，庄思远，吴章浩，庄永浩，林子，李卓翰，李大成，Eric P. Xing，张浩，Joseph E. Gonzalez，Ion Stoica。2023 年。使用 mt-bench 和 chatbot arena 评估大型语言模型 (LLM) 作为评审的表现。