LLM-Powered GUI Agents in Phone Automation: Surveying Progress and Prospects

基于大型语言模型 (LLM) 的手机自动化图形用户界面 (GUI) 代理: 进展与展望综述

Guangyi Liu [1,†] , Pengxiang Zhao [1,†] , Liang Liu [2,†,‡] , Yaxuan Guo [2] , Han Xiao [3] , Weifeng Lin [3] Yuxiang Chai [3] , Yue Han [1] , Shuai Ren [2] , Hao Wang [1] , Xiaoyu Liang [1] , Wenhao Wang [1]

刘光义 [1,†]，赵鹏翔 [1,†]，刘亮 [2,†,‡]，郭雅轩 [2]，肖涵 [3]，林伟峰 [3]，柴宇翔 [3]，韩越 [1]，任帅 [2]，王昊 [1]，梁晓宇 [1]，王文浩 [1]

Tianze Wu[1] , Linghao Li[1] , Hao Wang[2] , Guanjing Xiong [2] , Yong Liu [1,[∞]] , Hongsheng Li[3,] [1] Zhejiang University [2] vivo Al Lab [3] CUHK MMLab

田泽 Wu[1]，凌浩 Li[1]，昊 Wang[2]，熊冠京 [2]，刘勇 [1,[∞]]，洪升 Li[3,] [1] 浙江大学 [2] vivo AI 实验室 [3] 香港中文大学多媒体实验室

[†] Equal Contribution, [‡] Project Lead, [⊠] Corresponding Authors

[†] 贡献相等，[‡] 项目负责人，[⊠] 通讯作者

yongliu@iipc.zju.edu.cn; hsli@ee.cuhk.edu.hk

yongliu@iipc.zju.edu.cn; hsli@ee.cuhk.edu.hk

Project Homepage: github.com/PhoneLLM/Awesome-LLM-Powered-Phone-GUI-Agents

项目主页:github.com/PhoneLLM/Awesome-LLM-Powered-Phone-GUI-Agents

Abstract-With the rapid rise of large language models (LLMs), phone automation has undergone transformative changes. This paper systematically reviews LLM-driven phone GUI agents, highlighting their evolution from script-based automation to intelligent, adaptive systems. We first contextualize key challenges, (i) limited generality, (ii) high maintenance overhead, and (iii) weak intent comprehension, and show how LLMs address these issues through advanced language understanding, multimodal perception, and robust decision-making. We then propose a taxonomy covering fundamental agent frameworks (single-agent, multi-agent, plan-then-act), modeling approaches (prompt engineering, training-based), and essential datasets and benchmarks. Furthermore, we detail task-specific architectures, supervised fine-tuning, and reinforcement learning strategies that bridge user intent and GUI operations. Finally, we discuss open challenges such as dataset diversity, on-device deployment efficiency, user-centric adaptation, and security concerns, offering forward-looking insights into this rapidly evolving field. By providing a structured overview and identifying pressing research gaps, this paper serves as a definitive reference for researchers and practitioners seeking to harness LLMs in designing scalable, user-friendly phone GUI agents.

摘要-随着大型语言模型 (LLMs) 的快速兴起，手机自动化经历了变革性的发展。本文系统回顾了基于 LLM 的手机 GUI 代理，重点介绍了其从基于脚本的自动化向智能、自适应系统的演进。我们首先阐述了关键挑战:(i) 泛化能力有限，(ii) 维护成本高，(iii) 意图理解薄弱，并展示了 LLM 如何通过先进的语言理解、多模态感知和稳健的决策能力解决这些问题。随后，我们提出了涵盖基础代理框架 (单代理、多代理、先规划后执行)、建模方法 (提示工程、基于训练) 以及核心数据集和基准的分类体系。此外，本文详细介绍了任务特定架构、监督微调和强化学习策略，这些策略桥接了用户意图与 GUI 操作。最后，我们讨论了数据集多样性、设备端部署效率、以用户为中心的适应性及安全性等开放挑战，并对该快速发展的领域提出了前瞻性见解。通过提供结构化的综述并识别紧迫的研究空白，本文为研究人员和实践者设计可扩展、用户友好的手机 GUI 代理提供了权威参考。

Index Terms-Large Language Models, GUI Agents, Phone Automation, Mobile Interfaces, Natural Language Processing

关键词-大型语言模型，GUI 代理，手机自动化，移动界面，自然语言处理

1 INTRODUCTION

1 引言

THE core of phone GUI automation involves programmatically simulating human interactions with mobile interfaces to accomplish complex tasks. This technology has wide applications in testing and shortcut creation, enhancing efficiency and reducing manual effort [1], [2], [3], [4], [5]. Traditional approaches rely on predefined scripts and templates which, while functional, lack flexibility when confronting variable interfaces and dynamic environments [6], [7], [8], [9], [10].

手机 GUI 自动化的核心在于通过程序模拟人类与移动界面的交互，以完成复杂任务。该技术广泛应用于测试和快捷方式创建，提升效率并减少人工操作 [1]，[2]，[3]，[4]，[5]。传统方法依赖预定义脚本和模板，虽能实现功能，但在面对多变的界面和动态环境时缺乏灵活性 [6]，[7]，[8]，[9]，[10]。

In computer science, an agent perceives its environment through sensors and acts via actuators to achieve goals [11], [12], [13], [14], [15]. These range from simple scripts to complex systems capable of learning and adaptation [13], [14], [16]. Traditional phone automation agents are constrained by static scripts and limited adaptability, making them ill-suited for modern mobile interfaces' dynamic nature.

在计算机科学中，代理通过传感器感知环境，通过执行器采取行动以实现目标 [11]，[12]，[13]，[14]，[15]。这些代理从简单脚本到具备学习和适应能力的复杂系统不等 [13]，[14]，[16]。传统手机自动化代理受限于静态脚本和有限的适应性，难以应对现代移动界面的动态特性。

Building intelligent autonomous agents with planning, decision-making, and execution capabilities remains a long-term AI goal [17]. As technologies advanced, agents evolved from traditional forms [18], [19], [20] to AI agents [21], [22], [23] incorporating machine learning and probabilistic decision-making. However, these still struggle with complex instructions [24], [25] and dynamic environments [26], [27].

构建具备规划、决策和执行能力的智能自主代理仍是人工智能的长期目标 [17]。随着技术进步，代理从传统形式 [18]，[19]，[20] 演变为融合机器学习和概率决策的 AI 代理 [21]，[22]，[23]。然而，这些代理在处理复杂指令 [24]，[25] 和动态环境 [26]，[27] 时仍面临挑战。

With the rapid development of Large Language Models (LLMs) like the GPT series [28], [29], [30], [31] and specialized models such as Fuyu-8B [32], LLM-based agents have demonstrated powerful capabilities across numerous domains [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43]. As Figure 1 illustrates, conversational LLMs primarily focus on language understanding and generation, while LLM-based agents extend these capabilities by integrating perception and action components. This integration enables interaction with external environments through multimodal inputs and operational outputs [33], [34], [41], bridging language understanding and real-world interactions [11], [12], [44], [45].

随着 GPT 系列 [28]、[29]、[30]、[31] 等大型语言模型 (LLMs) 及 Fuyu-8B[32] 等专用模型的快速发展，基于 LLM 的智能体在众多领域展现出强大能力 [33]、[34]、[35]、[36]、[37]、[38]、[39]、[40]、[41]、[42]、[43]。如图 1 所示，会话型 LLM 主要聚焦于语言理解与生成，而基于 LLM 的智能体通过整合感知与行动组件，扩展了这些能力。该整合使其能够通过多模态输入和操作输出与外部环境交互 [33]、[34]、[41]，实现语言理解与现实世界交互的桥接 [11]、[12]、[44]、[45]。

Applying LLM-based agents to phone automation has created a new paradigm, making mobile interface operations more intelligent [46], [47], [48], [49]. LLM-powered phone GUI agents are intelligent systems that leverage large language models to understand, plan, and execute tasks on mobile devices by integrating natural language processing, multimodal perception, and action execution capabilities. These agents can recognize interfaces, understand instructions, perceive changes in real time, and respond dynamically. Unlike script-based automation, they can autonomously plan complex sequences through multimodal processing of instructions and interface information. Their adaptability and flexibility improve user experience through intent understanding, planning, and automated task execution, enhancing efficiency across scenarios from app testing to complex operations like configuring settings [50], navigating maps [51], [52], and shopping [48].

将基于 LLM 的智能体应用于手机自动化，开创了新的范式，使移动界面操作更加智能 [46]、[47]、[48]、[49]。基于 LLM 的手机 GUI 智能体是一种智能系统，利用大型语言模型结合自然语言处理、多模态感知和行动执行能力，理解、规划并执行移动设备上的任务。这些智能体能够识别界面、理解指令、实时感知变化并动态响应。不同于基于脚本的自动化，它们能通过多模态处理指令和界面信息，自主规划复杂操作序列。其适应性和灵活性通过意图理解、规划和自动任务执行提升用户体验，增强了从应用测试到复杂操作 (如配置设置 [50]、导航地图 [51]、[52] 及购物 [48]) 等多场景的效率。

Clarifying the development trajectory of phone GUI agents is crucial. On one hand, with the support of large language models [28], [29], [30], [31], phone GUI agents can significantly enhance the efficiency of phone automation scenarios, making operations more intelligent and no longer limited to coding fixed operation paths. This enhancement not only optimizes phone automation processes but also expands the application scope of automation. On the other hand, phone GUI agents can understand and execute complex natural language instructions, transforming human intentions into specific operations such as automatically scheduling appointments, booking restaurants, summoning transportation, and even achieving functionalities similar to autonomous driving in advanced automation. These capabilities demonstrate the potential of phone GUI agents in executing complex tasks, providing convenience to users and laying practical foundations for AI development.

明确手机 GUI 智能体的发展轨迹至关重要。一方面，在大型语言模型 [28] 支持下，手机 GUI 智能体能显著提升手机自动化场景的效率，使操作更智能，不再局限于编码固定操作路径。这种提升不仅优化了手机自动化流程，也拓展了自动化的应用范围。另一方面，手机 GUI 智能体能够理解并执行复杂的自然语言指令，将人类意图转化为具体操作，如自动安排预约、预订餐厅、召唤交通工具，甚至在高级自动化中实现类似自动驾驶的功能。这些能力展示了手机 GUI 智能体执行复杂任务的潜力，为用户带来便利，并为人工智能发展奠定了实践基础。

Fig. 1: Comparison between conversational LLMs and phone GUI agents. While a conversational LLM can understand queries and provide informative responses (e.g., recommending coffee beans), a Phone GUI agent can go beyond text generation to perceive the device's interface, decide on an appropriate action (like tapping an app icon), and execute it in the real environment, thus enabling tasks like ordering a latte directly on the user's phone.

图 1: 会话型 LLM 与手机 GUI 智能体的对比。会话型 LLM 能够理解查询并提供信息性回答 (例如推荐咖啡豆)，而手机 GUI 智能体则超越文本生成，能够感知设备界面，决定合适的操作 (如点击应用图标)，并在真实环境中执行，从而实现如直接在用户手机上点单拿铁等任务。

With the increasing research on large language models in phone automation [50], [51], [52], [53], [54], [55], [56], the research community's attention to this field has grown rapidly. However, there is still a lack of dedicated systematic surveys in this area, especially comprehensive explorations of phone automation from the perspective of large language models. Given the importance of phone GUI agents, the purpose of this paper is to fill this gap by systematically summarizing current research achievements, reviewing relevant literature, analyzing the application status of large language models in phone automation, and pointing out directions for future research.

随着大型语言模型在手机自动化领域的研究日益增多 [50]、[51]、[52]、[53]、[54]、[55]、[56]，学术界对此领域的关注迅速提升。然而，当前仍缺乏专门的系统性综述，尤其是从大型语言模型视角对手机自动化的全面探讨。鉴于手机 GUI 智能体的重要性，本文旨在填补该空白，系统总结现有研究成果，回顾相关文献，分析大型语言模型在手机自动化中的应用现状，并指出未来研究方向。

To provide a comprehensive overview of the current state and future prospects of LLM-Powered GUI Agents in Phone Automation, we present a taxonomy that categorizes the field into three main areas: Frameworks of LLM-powered phone GUI agents, Large Language Models for Phone Automation, and Datasets and Evaluation Methods Figure 2. This taxonomy highlights the diversity and complexity of the field, as well as the interdisciplinary nature of the research involved.

为全面概述基于 LLM 的手机 GUI 智能体的现状与未来前景，本文提出了一个分类体系，将该领域划分为三个主要部分: 基于 LLM 的手机 GUI 智能体框架、手机自动化中的大型语言模型，以及数据集与评估方法 (见图 2)。该分类体系凸显了该领域的多样性与复杂性，以及研究的跨学科特性。

Unlike previous literature reviews, which primarily focus on traditional phone automated testing methods, most existing surveys emphasize manual scripting or rule-based automation approaches without leveraging LLMs [6], [7], [8], [9], [10]. These traditional methods face significant challenges in coping with dynamic changes, complex user interfaces, and the scalability required for modern applications. Although recent surveys have explored broader areas of multimodal agents and foundation models for GUI automation, such as Foundations and Recent Trends in Multimodal Mobile Agents: A Survey [157], GUI Agents with Foundation Models: A Comprehensive Survey [158], and Large Language Model-Brained GUI Agents: A Survey [159], these works primarily cover general GUI-based automation and multimodal applications.

不同于以往主要聚焦传统手机自动化测试方法的文献综述, 大多数现有调查强调手动脚本或基于规则的自动化方法，未充分利用 LLM[6]、[7]、[8]、[9]、[10]。这些传统方法在应对动态变化、复杂用户界面及现代应用所需的可扩展性方面面临重大挑战。尽管近期有综述探讨了多模态智能体和基础模型在 GUI 自动化中的更广泛应用，如《多模态移动智能体的基础与最新趋势综述》[157]、《基于基础模型的 GUI 智能体综合综述》[158] 及《大型语言模型驱动的 GUI 智能体综述》[159]，但这些工作主要涵盖一般 GUI 自动化和多模态应用。

However, a dedicated and focused survey on the role of large language models in phone GUI automation remains absent in the existing literature. This paper addresses the above-mentioned gap by systematically reviewing the latest developments, challenges, and opportunities in LLM-powered phone GUI agents, thereby offering a more targeted exploration of this emerging domain. Our main contributions can be summarized as follows:

然而，现有文献中尚无专门聚焦大型语言模型在手机 GUI 自动化中作用的系统综述。本文针对上述空白，系统回顾了基于 LLM 的手机 GUI 智能体的最新进展、挑战与机遇，提供了对该新兴领域更具针对性的探讨。我们的主要贡献总结如下:

- A Comprehensive and Systematic Survey of LLM-Powered Phone GUI Agents. We provide an in-depth and structured overview of recent literature on LLM-powered phone automation, examining its developmental trajectory, core technologies, and real-world application scenarios. By comparing LLM-driven methods to traditional phone automation approaches, this survey clarifies how large models transform GUI-based tasks and enable more intelligent, adaptive interaction paradigms.

  - 基于大型语言模型 (LLM) 的手机图形用户界面 (GUI) 代理的全面系统综述。我们深入且结构化地回顾了近期关于 LLM 驱动手机自动化的文献，考察其发展轨迹、核心技术及实际应用场景。通过将 LLM 驱动的方法与传统手机自动化方法进行比较，本综述阐明了大型模型如何变革基于 GUI 的任务，并实现更智能、适应性更强的交互范式。

- Methodological Framework from Multiple Perspectives. Leveraging insights from existing studies, we propose a unified methodology for designing LLM-driven phone GUI agents. This encompasses framework design (e.g., single-agent vs. multi-agent vs. plan-then-act frameworks), LLM model selection and training (prompt engineering vs. training-based methods), data collection and preparation strategies (GUI-specific datasets and annotations), and evaluation protocols (benchmarks and metrics). Our systematic taxonomy and method-oriented discussion serve as practical guidelines for both academic and industrial practitioners.

- 多视角的方法论框架。借鉴现有研究的见解，我们提出了一个统一的方法论，用于设计 LLM 驱动的手机 GUI 代理。该方法涵盖框架设计 (如单代理、多代理及先规划后执行框架)、LLM 模型选择与训练 (提示工程与基于训练的方法)、数据收集与准备策略 (GUI 专用数据集及标注) 以及评估协议 (基准与指标)。我们的系统分类法和方法导向讨论为学术界和工业界实践者提供了实用指南。

- In-Depth Analysis of Why LLMs Empower Phone Automation. We delve into the fundamental reasons behind LLMs' capacity to enhance phone automation. By detailing their advancements in natural language comprehension, multimodal grounding, reasoning, and decision-making, we illustrate how LLMs bridge the gap between user intent and GUI actions. This analysis elucidates the critical role of large models in tackling issues of scalability, adaptability, and human-like interaction in real-world mobile environment.

- 深入分析 LLM 为何赋能手机自动化。我们探讨了 LLM 提升手机自动化能力的根本原因。通过详述其在自然语言理解、多模态基础、推理与决策方面的进展，展示了 LLM 如何弥合用户意图与 GUI 操作之间的鸿沟。此分析阐明了大型模型在解决移动环境中可扩展性、适应性及类人交互问题上的关键作用。

- Insights into Latest Developments, Datasets, and Benchmarks. We introduce and evaluate the most recent progress in the field, highlighting innovative datasets that capture the complexity of modern GUIs and benchmarks that allow reliable performance assessment. These resources form the backbone of LLM-based phone automation, enabling systematic training, fair evaluation, and transparent comparisons across different agent designs.

- 最新进展、数据集与基准的洞见。我们介绍并评估了该领域的最新成果，重点突出捕捉现代 GUI 复杂性的创新数据集及支持可靠性能评估的基准。这些资源构成了基于 LLM 的手机自动化的基础，支持系统化训练、公平评估及不同代理设计间的透明比较。

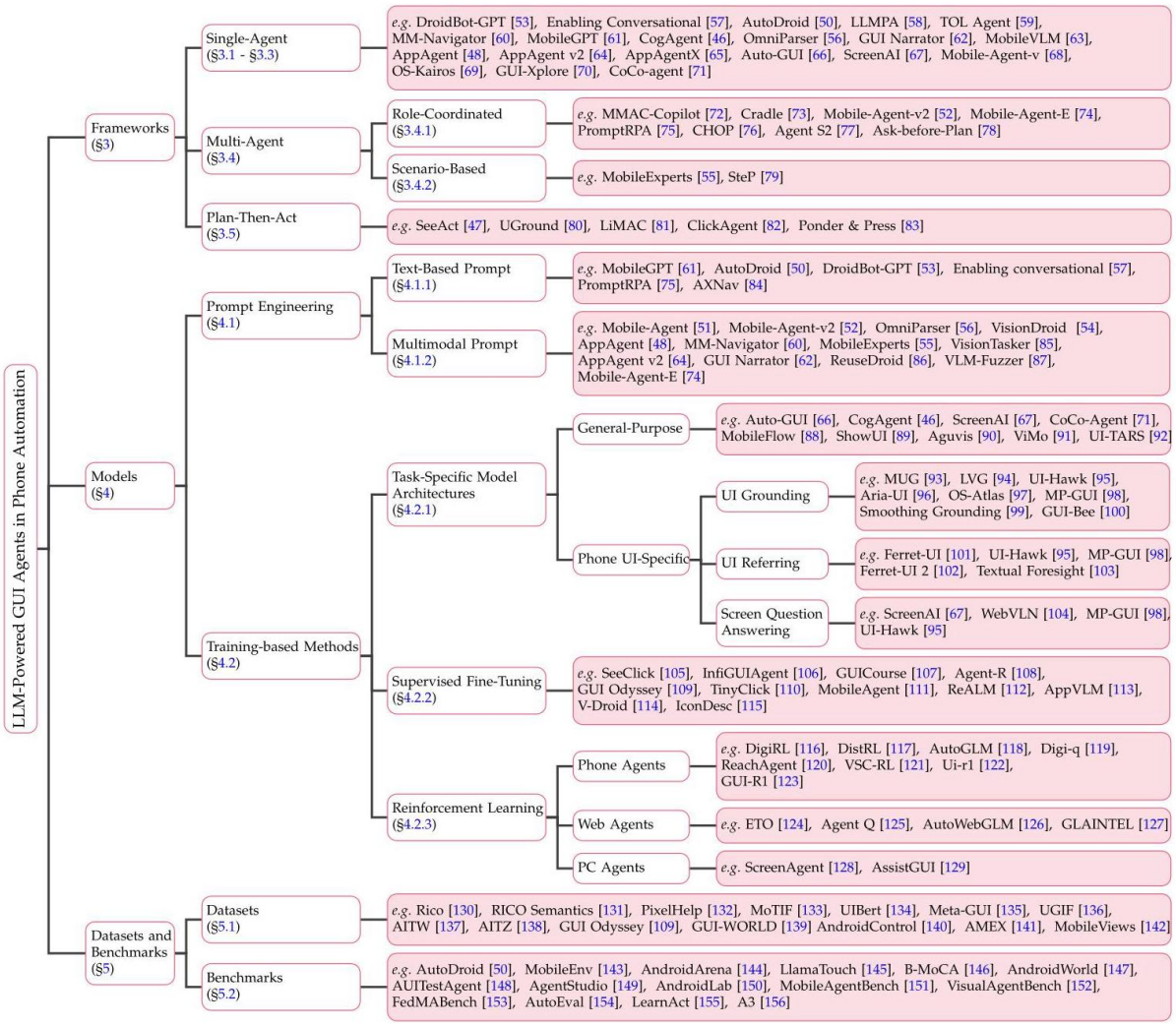

Fig. 2: A comprehensive taxonomy of LLM-powered phone GUI agents in phone automation. Note that only a selection of representative works is included in this categorization.

图 2: 基于 LLM 的手机 GUI 代理在手机自动化中的全面分类法。注意，此分类仅包含部分具有代表性的工作。

- Identification of Key Challenges and Novel Perspectives for Future Research. Beyond discussing mainstream hurdles (e.g., dataset coverage, on-device constraints, reliability), we propose forward-looking viewpoints on user-centric adaptations, security and privacy considerations, long-horizon planning, and multi-agent coordination. These novel perspectives shed light on how researchers and developers might advance the current state of the art toward more robust, secure, and personalized phone GUI agents.

- 关键挑战识别及未来研究的新视角。除讨论主流难题 (如数据集覆盖、设备端限制、可靠性) 外，我们提出了面向用户适应、安全隐私考量、长远规划及多代理协作的前瞻性观点。这些新视角为研究者和开发者如何推动现有技术向更稳健、安全及个性化的手机 GUI 代理发展提供了启示。

By addressing these aspects, our survey not only provides an up-to-date map of LLM-powered phone GUI automation but also offers a clear roadmap for future exploration. We hope this work will guide researchers in iden-

tifying pressing open problems and inform practitioners about promising directions to harness LLMs in designing efficient, adaptive, and user-friendly phone GUI agents.

通过涵盖这些方面，我们的综述不仅提供了基于 LLM 的手机 GUI 自动化的最新全景图，还为未来探索指明了清晰路线。我们希望此工作能引导研究者识别紧迫的开放问题，并为实践者提供利用 LLM 设计高效、适应性强且用户友好手机 GUI 代理的有益方向。

2 DEVELOPMENT OF PHONE AUTOMATION

2 手机自动化的发展

The evolution of phone automation has been marked by significant technological advancements [160], particularly with the emergence of LLMs [28], [29], [30], [31]. This section explores the historical development of phone automation, the challenges faced by traditional methods, and how LLMs have revolutionized the field.

手机自动化的发展历程伴随着显著的技术进步 [160]，尤其是大型语言模型 (LLM)[28]，[29]，[30]，[31] 的出现。本节探讨手机自动化的历史发展、传统方法面临的挑战及 LLM 如何革新该领域。

2.1 Phone Automation Before the LLM Era

2.1 LLM 时代之前的手机自动化

Before the advent of LLMs, phone automation was predominantly achieved through traditional technical methods [1], [160], [161], [162], [163], [164]. This subsection delves into the primary areas of research and application during that period, including automation testing, shortcuts, and Robotic Process Automation (RPA), highlighting their methodologies and limitations.

在 LLM 出现之前，手机自动化主要依赖传统技术方法 [1]，[160]，[161]，[162]，[163]，[164]。本小节深入探讨该时期的主要研究与应用领域，包括自动化测试、快捷方式及机器人流程自动化 (RPA)，重点介绍其方法论及局限性。

2.1.1 Automation Testing

2.1.1 自动化测试

Phone applications (apps) have become extremely popular, with approximately 1.68 million apps in the Google Play Store [1]. The increasing complexity of apps [165] has raised significant concerns about app quality. Moreover, due to rapid release cycles and limited human resources, developers find it challenging to manually construct test cases. Therefore, various automated phone app testing techniques have been developed and applied, making phone automation testing the main application of phone automation before the era of large models [160], [161], [163], [166]. Test cases for phone apps are typically represented by a sequence of GUI events [167] to simulate user interactions with the app. The goal of automated test generators is to produce such event sequences to achieve high code coverage or detect bugs [164].

手机应用程序 (app) 极为流行，Google Play 商店中约有 168 万个应用 [1] 。应用复杂性的增加 [165] 引发了对应用质量的高度关注。此外，由于快速的发布周期和有限的人力资源，开发者难以手动构建测试用例。因此，各种自动化手机应用测试技术被开发并应用，使手机自动化测试成为大型模型时代之前手机自动化的主要应用 [160]，[161]，[163]，[166]。手机应用的测试用例通常由一系列 GUI 事件 [167] 表示，以模拟用户与应用的交互。自动测试生成器的目标是产生此类事件序列，以实现高代码覆盖率或发现缺陷 [164]。

In the development history of phone automation testing, we have witnessed several key breakthroughs and advancements. Initially, random testing (e.g., Monkey Testing [168]) was used as a simple and fundamental testing method, detecting application stability and robustness by randomly generating user actions. Although this method could cover a wide range of operational scenarios, its testing process lacked focus and was difficult to reproduce and pinpoint specific issues [160].

在手机自动化测试的发展历程中，我们见证了若干关键突破和进展。最初，随机测试 (如 Monkey Testing[168]) 作为一种简单且基础的测试方法被采用，通过随机生成用户操作检测应用的稳定性和鲁棒性。尽管该方法能覆盖广泛的操作场景，但其测试过程缺乏针对性，难以复现和定位具体问题 [160]。

Subsequently, model-based testing [1], [162], [169] became a more systematic testing approach. It establishes a user interface model of the application, using predefined states and transition rules to generate test cases. This method improved testing coverage and efficiency, but the construction and maintenance of the model required substantial manual involvement, and updating the model became a challenge for highly dynamic applications.

随后，基于模型的测试 [1]，[162]，[169] 成为了一种更系统的测试方法。它通过建立应用程序的用户界面模型，利用预定义的状态和转换规则生成测试用例。该方法提高了测试覆盖率和效率，但模型的构建和维护需要大量人工参与，且对于高度动态的应用程序，更新模型成为一大挑战。

With the development of machine learning techniques, learning-based testing methods began to emerge [2], [3], [4], [5]. These methods generate test cases by analyzing historical data to learn user behavior patterns. For example, Humanoid [4] uses deep learning to mimic human tester interaction behavior and uses the learned model to guide test generation like a human tester. However, this method relies on human-generated datasets to train the model and needs to combine the model with a set of predefined rules to guide testing.

随着机器学习技术的发展，基于学习的测试方法开始出现 [2]，[3]，[4]，[5]。这些方法通过分析历史数据学习用户行为模式来生成测试用例。例如，Humanoid[4] 利用深度学习模拟人类测试者的交互行为，并使用学习到的模型指导测试生成，类似于人类测试者。然而，该方法依赖于人类生成的数据集来训练模型，并需要将模型与一套预定义规则结合以指导测试。

Recently, reinforcement learning [170] has shown great potential in the field of automated testing. DinoDroid [164] is an example that uses Deep Q-Network (DQN) [171] to automate testing of Android applications. By learning behavior models of existing applications, it automatically explores and generates test cases, not only improving code coverage but also enhancing bug detection capabilities. Deep reinforcement learning methods can handle more complex state spaces and make more intelligent decisions but also face challenges such as high training costs and poor model generalization capabilities [172].

近年来，强化学习 [170] 在自动化测试领域展现出巨大潜力。DinoDroid[164] 是一个使用深度 Q 网络 (Deep Q-Network，DQN)[171] 实现安卓应用自动化测试的例子。通过学习现有应用的行为模型，它能够自动探索并生成测试用例，不仅提升了代码覆盖率，还增强了缺陷检测能力。深度强化学习方法能够处理更复杂的状态空间并做出更智能的决策，但也面临训练成本高和模型泛化能力差等挑战 [172]。

2.1.2 Shortcuts

2.1.2 快捷方式

Shortcuts on mobile devices refer to predefined rules or trigger conditions that enable users to execute a series of actions automatically [173], [174], [175]. These shortcuts are designed to streamline interaction by reducing repetitive manual input. For instance, the Tasker app on the Android platform [2] and the Shortcuts feature on iOS [3] allow users to automate tasks like turning on Wi-Fi, sending text messages, or launching apps under specific conditions such as time, location, or events. These implementations leverage simple IF-THEN and manually-designed logic but are inherently limited in scope and flexibility.

移动设备上的快捷方式指预定义的规则或触发条件，使用户能够自动执行一系列操作 [173]，[174]，[175]。这些快捷方式旨在通过减少重复的手动输入来简化交互。例如，安卓平台上的 Tasker 应用 [2] 和 iOS 上的快捷指令功能 [3] 允许用户在特定条件 (如时间、位置或事件) 下自动执行打开 Wi-Fi、发送短信或启动应用等任务。这些实现利用简单的 IF-THEN 逻辑和手动设计的规则，但在范围和灵活性上存在固有限制。

2.1.3 Robotic Process Automation

2.1.3 机器人流程自动化

Robotic Process Automation(RPA) applications on phone devices aim to simulate human users performing repetitive tasks across applications [176]. Phone RPA tools generate repeatable automation processes by recording user action sequences. These tools are used in enterprise environment to automate tasks such as data entry and information gathering, reducing human errors and improving efficiency, but they struggle with dynamic interfaces and require frequent script updates [177], [178].

手机设备上的机器人流程自动化 (Robotic Process Automation，RPA) 应用旨在模拟人类用户跨应用执行重复任务 [176]。手机 RPA 工具通过记录用户操作序列生成可重复的自动化流程。这些工具在企业环境中用于自动化数据录入和信息收集等任务，减少人为错误并提高效率，但它们在应对动态界面时表现不佳，且需要频繁更新脚本 [177]，[178]。

2.2 Challenges of Traditional Methods

2.2 传统方法的挑战

Despite the advancements made, traditional phone automation methods faced significant challenges that hindered further development. This subsection analyzes these challenges, including lack of generality and flexibility, high maintenance costs, difficulty in understanding complex user intentions, and insufficient intelligent perception, highlighting the need for new approaches.

尽管取得了一定进展，传统手机自动化方法仍面临诸多阻碍进一步发展的重大挑战。本小节分析这些挑战，包括缺乏通用性和灵活性、高维护成本、难以理解复杂用户意图以及智能感知不足，强调了新方法的必要性。

2.2.1 Limited Generality

2.2.1 通用性有限

Traditional automation methods are often tailored to specific applications and interfaces, lacking adaptability to different apps and dynamic user environment [179], [180], [181], [182]. For example, automation scripts designed for a specific app may not function correctly if the app updates its interface or if the user switches to a different app with similar functionality. This inflexibility makes it difficult to extend automation across various usage scenarios without significant manual reconfiguration.

传统自动化方法通常针对特定应用和界面定制，缺乏对不同应用和动态用户环境的适应能力 [179], [180], [181], [182]。例如，为某一特定应用设计的自动化脚本在该应用更新界面或用户切换到功能相似的其他应用时可能无法正常工作。这种不灵活性使得在各种使用场景中推广自动化变得困难，且需大量人工重新配置。

These methods typically follow predefined sequences of actions and cannot adjust their operations based on changing contexts or user preferences. For instance, if a user wants an automation to send a customized message to contacts who have birthdays on a particular day, traditional methods struggle because they cannot dynamically access and interpret data from the contacts app, calendar, and messaging app simultaneously. Similarly, automating tasks that require conditional logic-such as playing different music genres based on the time of day or weather conditions-poses a challenge for traditional automation tools, as they lack the ability to integrate real-time data and make intelligent decisions accordingly [183], [184].

这些方法通常遵循预定义的操作序列，无法根据变化的上下文或用户偏好调整操作。例如，若用户希望自动化发送定制消息给当天生日的联系人，传统方法难以实现，因为它们无法动态访问并同时解析联系人应用、日历和消息应用的数据。同样，自动化执行基于时间或天气条件播放不同音乐类型等需要条件逻辑的任务，对传统自动化工具来说也是挑战，因为它们缺乏整合实时数据并据此做出智能决策的能力 [183]，[184]。

2.2.2 High Maintenance Costs

2.2.2 高维护成本

Writing and maintaining automation scripts require professional knowledge and are time-consuming and labor-intensive [185], [186], [187], [188], [189]. Taking RPA as an example, as applications continually update and

iterate, scripts need frequent modifications. When an application's interface layout changes or functions are updated, RPA scripts originally written for the old version may not work properly, requiring professionals to spend considerable time and effort readjusting and optimizing the scripts [190], [191], [192].

编写和维护自动化脚本需要专业知识，且耗时费力 [185]，[186]，[187]，[188]，[189]。以 RPA 为例，随着应用不断更新迭代，脚本需要频繁修改。当应用界面布局变化或功能更新时，原为旧版本编写的 RPA 脚本可能无法正常工作，需专业人员投入大量时间和精力进行调整和优化 [190]，[191]，[192]。

---

2. https://play.google.com.

2. https://play.google.com.

3. https://support.apple.com.

3. https://support.apple.com.

1. https://www.statista.com.

1. https://www.statista.com.

The high entry barrier also limits the popularity of some automation features [193], [194]. For example, Apple's Shortcuts [4] can combine complex operations, such as starting an Apple Watch fitness workout, recording training data, and sending statistical data to the user's email after the workout. However, setting up such a complex shortcut often requires the user to perform a series of complicated operations on the phone following fixed rules. This is challenging for ordinary users, leading many to abandon usage due to the complexity of manual script writing.

高门槛也限制了一些自动化功能的普及 [193]，[194]。例如，苹果的快捷指令 (Shortcuts) [4] 可以组合复杂操作，如启动 Apple Watch 健身锻炼、记录训练数据，并在锻炼后将统计数据发送到用户邮箱。然而，设置如此复杂的快捷指令通常需要用户按照固定规则在手机上执行一系列复杂操作。这对普通用户来说具有挑战性，导致许多人因手动编写脚本的复杂性而放弃使用。

2.2.3 Poor Intent Comprehension

2.2.3 意图理解能力不足

Rule-based and script-based systems can only execute predefined tasks or engage in simple natural language interactions [195], [196]. Simple instructions like "open the browser" can be handled using traditional natural language processing algorithms, but complex instructions like "open the browser, go to Amazon, and purchase a product" cannot be completed. These traditional systems are based on fixed rules and lack in-depth understanding and parsing capabilities for complex natural language [197], [198], [199].

基于规则和脚本的系统只能执行预定义任务或进行简单的自然语言交互 [195]，[196]。像“打开浏览器”这样的简单指令可以通过传统自然语言处理算法处理，但“打开浏览器，访问亚马逊并购买商品”这样的复杂指令则无法完成。这些传统系统基于固定规则，缺乏对复杂自然语言的深入理解和解析能力 [197]，[198]，[199]。

They require users to manually write scripts to interact with the phone, greatly limiting the application of intelligent assistants that can understand complex human instructions. For example, when a user wants to check flight information for a specific time and book a ticket, traditional systems cannot accurately understand the user's intent and automatically complete the series of related operations, necessitating manual script writing with multiple steps, which is cumbersome and requires high technical skills.

它们需要用户手动编写脚本与手机交互，极大限制了能够理解复杂人类指令的智能助手的应用。例如，当用户想查询特定时间的航班信息并预订机票时，传统系统无法准确理解用户意图并自动完成一系列相关操作，必须通过多步骤的手动脚本编写，既繁琐又要求较高的技术水平。

2.2.4 Weak Screen GUI Perception

2.2.4 屏幕 GUI 感知能力弱

Different applications present a wide variety of GUI elements, making it challenging for traditional methods like RPA to accurately recognize and interact with diverse controls [200], [201], [202], [203]. Traditional automation often relies on fixed sequences of actions targeting specific controls or input fields, exhibiting Weak Screen GUI Perception that limits their ability to adapt to variations in interface layouts and component types. For example, in an e-commerce app, the product details page may include dynamic content like carousels, embedded videos, or interactive size selection menus, which differ significantly from the simpler layout of a search results page. Traditional methods may fail to accurately identify and interact with the ”Add to Cart” button or select product options, leading to unsuccessful automation of purchasing tasks.

不同应用呈现多样化的 GUI 元素，使得传统方法如 RPA 难以准确识别和操作各种控件 [200]，[201]，[202]，[203]。传统自动化通常依赖针对特定控件或输入框的固定操作序列，表现出屏幕 GUI 感知能力弱，限制了其适应界面布局和组件类型变化的能力。例如，在电商应用中，商品详情页可能包含轮播图、嵌入视频或交互式尺码选择菜单等动态内容，这与搜索结果页的简单布局有显著差异。传统方法可能无法准确识别和操作“加入购物车”按钮或选择商品选项，导致购买任务自动化失败。

Moreover, traditional automation struggles with understanding complex screen information such as dynamic content updates, pop-up notifications, or context-sensitive menus that require adaptive interaction strategies. Without the ability to interpret visual cues like icons, images, or contextual hints, these methods cannot handle tasks that involve navigating through multi-layered interfaces or responding to real-time changes. For instance, automating the process of booking a flight may involve selecting dates from a calendar widget, choosing seats from an interactive seat map, or handling security prompts-all of which require sophisticated perception and interpretation of the interface [145].

此外，传统自动化难以理解动态内容更新、弹出通知或上下文相关菜单等复杂屏幕信息，这些都需要自适应的交互策略。缺乏对图标、图片或上下文提示等视觉线索的解读能力，使得这些方法无法处理多层界面导航或实时变化响应的任务。例如，自动化预订航班过程可能涉及从日历控件选择日期、从交互式座位图选择座位或处理安全提示——所有这些都需要对界面进行复杂的感知和解析[145]。

In phone automation, many apps do not provide open API interfaces, forcing solutions to rely directly on the GUI for triggering actions and retrieving information. Even when tools are used to parse the Android UI [204], non-standard controls often prevent accurate JSON parsing, further complicating automated testing and interaction. Additionally, because the GUI is a universal and consistent interface across apps regardless of their internal design, it naturally becomes the central focus of phone automation methods.

在手机自动化中，许多应用不提供开放 API 接口，迫使解决方案直接依赖 GUI 触发操作和获取信息。即使使用工具解析 Android UI[204]，非标准控件也常阻碍准确的 JSON 解析，进一步增加自动化测试和交互的难度。此外，由于 GUI 是跨应用通用且一致的接口，自然成为手机自动化方法的核心焦点。

These limitations significantly impede the widespread application and deep development of traditional phone automation technologies. Without intelligent perception capabilities, automation cannot adapt to the complexities of modern app interfaces, which are increasingly dynamic and rich in interactive elements. This underscores the urgent need for new methods and technologies that can overcome these bottlenecks and achieve more intelligent, flexible, and efficient phone automation.

这些限制严重阻碍了传统手机自动化技术的广泛应用和深入发展。缺乏智能感知能力，自动化无法适应现代应用界面日益动态且丰富的交互元素。这凸显了亟需新方法和技术以突破瓶颈，实现更智能、灵活和高效的手机自动化。

2.3 LLMs Boost Phone Automation

2.3 大型语言模型 (LLMs) 推动手机自动化

The advent of LLMs has marked a significant shift in the landscape of phone automation, enabling more dynamic, context-aware, and sophisticated interactions with mobile devices. As illustrated in Figure 3, the research on LLM-powered phone GUI agents has progressed through pivotal milestones, where models become increasingly adept at interpreting multimodal data, reasoning about user intents, and autonomously executing complex tasks. This section clarifies how LLMs address traditional limitations and examines why scaling laws can further propel large models in phone automation. As will be detailed in §4 and §5 , LLM-based solutions for phone automation generally follow two routes: (1) Prompt Engineering, where pre-trained models are guided by carefully devised prompts, and (2) Training-Based Methods, where LLMs undergo additional optimization on GUI-focused datasets. The following subsections illustrate how LLMs mitigate the core challenges of traditional phone automation-ranging from contextual semantic understanding and GUI perception to reasoning and decision making-and briefly highlight the role of scaling laws in enhancing these capabilities.

大型语言模型 (LLMs) 的出现标志着手机自动化领域的重大转变，使得与移动设备的交互更加动态、具备上下文感知能力且更为复杂。如图 3 所示，基于 LLM 的手机 GUI 代理研究经历了关键里程碑，模型在多模态数据理解、用户意图推理及自主执行复杂任务方面日益成熟。本节阐明了 LLM 如何解决传统限制，并探讨了规模定律如何进一步推动大型模型在手机自动化中的发展。如 §4 和 §5 所述，基于 LLM 的手机自动化解决方案通常有两条路径:(1) 提示工程 (Prompt Engineering)，通过精心设计的提示引导预训练模型；(2) 基于训练的方法 (Training-Based Methods)，对 LLM 进行针对 GUI 数据集的额外优化。以下小节将展示 LLM 如何缓解传统手机自动化的核心挑战——从上下文语义理解、GUI 感知到推理决策，并简要强调规模定律在提升这些能力中的作用。

Scaling Laws in LLM-Based Phone Automation. Scaling laws-originally observed in general-purpose LLMs, where increasing model capacity and training data yields emergent capabilities [205], [206], [207]-have similarly begun to manifest in phone GUI automation. As datasets enlarge and encompass more diverse apps, usage scenarios, and user behaviors, recent findings [105], [107], [109], [110] show consistent gains in step-by-step automation tasks such as clicking buttons or entering text. This data scaling not only captures broader interface layouts and device contexts but also reveals latent ”emergent” competencies, allowing LLMs to handle more abstract, multi-step instructions. Empirical evidence from in-domain scenarios [140] further underscores how expanded coverage of phone apps and user patterns systematically refines automation accuracy. In essence, as model sizes and data complexity grow, phone GUI agents exploit these scaling laws to bridge the gap between user intent and real-world GUI interactions with increasing efficiency and sophistication.

基于大语言模型 (LLM) 的手机自动化中的规模定律。规模定律最初在通用大语言模型中被观察到，即随着模型容量和训练数据的增加，出现了新兴能力 [205]，[206]，[207]，这种现象也开始在手机图形用户界面 (GUI) 自动化中显现。随着数据集的扩大，涵盖更多样化的应用程序、使用场景和用户行为，最新研究 [105]，[107]，[109]，[110] 显示在逐步自动化任务 (如点击按钮或输入文本) 中持续取得进展。这种数据规模的扩大不仅捕捉了更广泛的界面布局和设备环境，还揭示了潜在的“新兴”能力，使 LLM 能够处理更抽象的多步骤指令。来自领域内场景的实证证据 [140] 进一步强调，手机应用和用户模式的覆盖范围扩大系统性地提升了自动化的准确性。本质上，随着模型规模和数据复杂度的增长，手机 GUI 代理利用这些规模定律，以日益高效和复杂的方式弥合用户意图与现实 GUI 交互之间的差距。

---

4. https://support.apple.com.

4. https://support.apple.com.

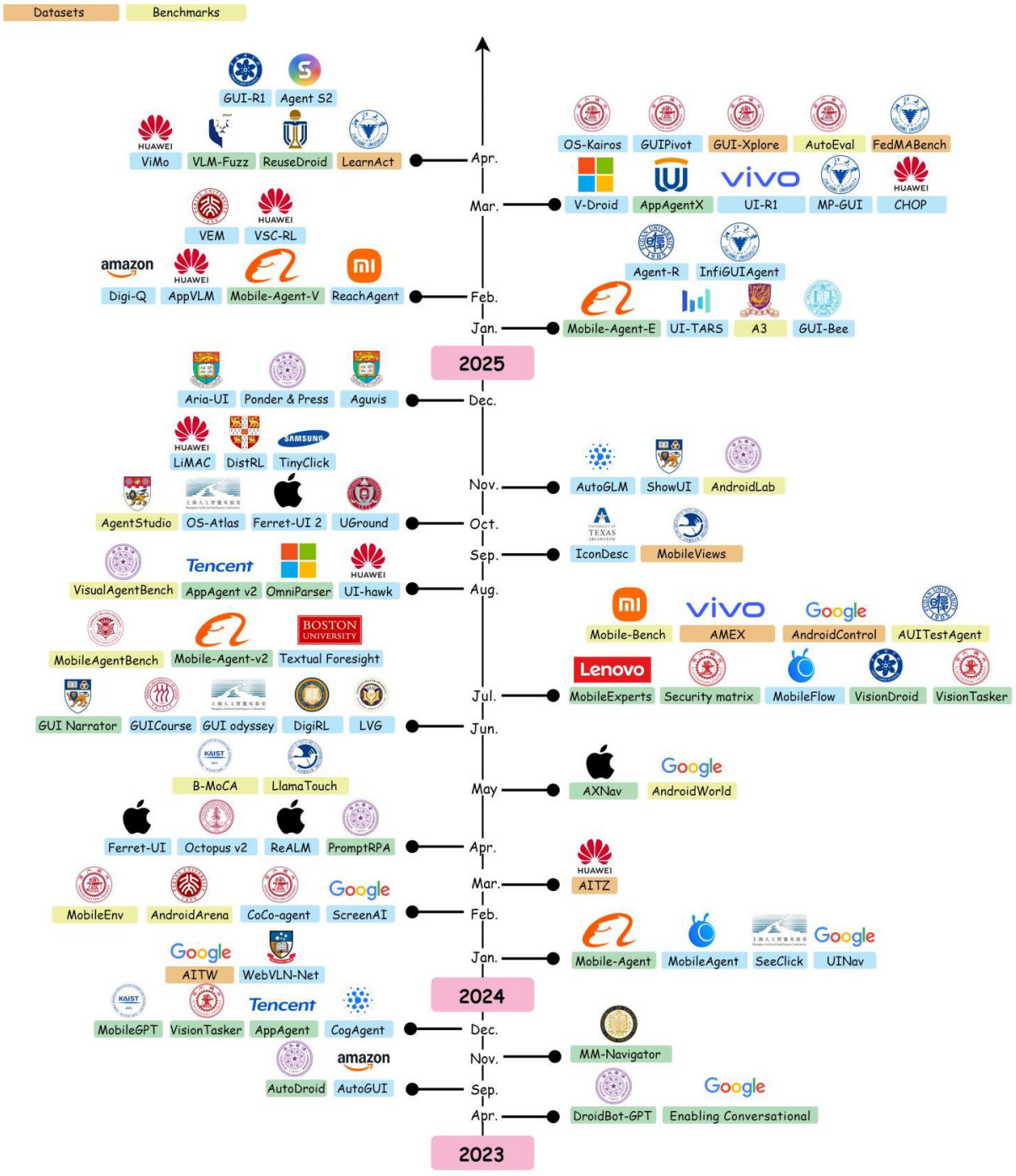

Fig. 3: Milestones in the development of LLM-powered phone GUI agents. This figure divides advancements into four primary parts: Prompt Engineering, Training-Based Methods, Datasets and Benchmarks. Prompt Engineering leverages pre-trained LLMs by strategically crafting input prompts, as detailed in §4.1, to perform specific tasks without modifying model parameters. In contrast, Training-Based Methods, discussed in §4.2, involve adapting LLMs via supervised fine-tuning or reinforcement learning on GUI-specific data, thereby enhancing their ability to understand and interact with mobile UIs.

图 3: 基于 LLM 的手机 GUI 代理发展的里程碑。该图将进展划分为四个主要部分: 提示工程、基于训练的方法、数据集和基准测试。提示工程通过策略性设计输入提示，利用预训练的 LLM，如 §4.1 所述，在不修改模型参数的情况下执行特定任务。相比之下，基于训练的方法，如 §4.2 讨论的，通过在 GUI 特定数据上进行监督微调或强化学习，调整 LLM，从而增强其理解和交互移动用户界面的能力。

Contextual Semantic Understanding. LLMs have transformed natural language processing for phone automation by learning from extensive textual corpora [48], [50], [205], [208], [209], [210]. This training captures intricate linguistic structures and domain knowledge [199], allowing agents to parse multi-step commands and generate context-informed responses. MobileAgent [51], for example, interprets user directives like scheduling appointments or performing transactions with high precision, harnessing the Transformer architecture [208] for efficient encoding of complex prompts. Consequently, phone GUI agents benefit from stronger natural language grounding, bridging user-intent gaps once prevalent in script-based systems.

上下文语义理解。LLM 通过从大量文本语料库中学习 [48]，[50]，[205]，[208]，[209]，[210]，彻底改变了手机自动化的自然语言处理。这种训练捕捉了复杂的语言结构和领域知识 [199]，使代理能够解析多步骤命令并生成基于上下文的响应。例如，MobileAgent[51] 能够高精度地解释用户指令，如安排预约或执行交易，利用 Transformer 架构 [208] 高效编码复杂提示。因此，手机 GUI 代理受益于更强的自然语言基础，弥合了脚本系统中常见的用户意图差距。

Screen GUI with Multi-Modal Perception. Screen GUI perception in earlier phone automation systems typically depended on static accessibility trees or rigid GUI element detection, which struggled to adapt to changing app interfaces. Advances in LLMs, supported by large-scale multimodal datasets [211], [212], [213], allow models to unify textual and visual signals in a single representation. Systems like UGround [80], Ferret-UI [101], and UI-Hawk [95] excel at grounding natural language descriptions to on-screen elements, dynamically adjusting as interfaces evolve. Moreover, SeeClick [105] and ScreenAI [67] demonstrate that learning directly from screenshots-rather than purely textual meta-data—can further enhance adaptability. By integrating visual cues with user language, LLM-based agents can respond more flexibly to a wide range of UI designs and interaction scenarios.

带多模态感知的屏幕 GUI。早期手机自动化系统中的屏幕 GUI 感知通常依赖静态的辅助功能树或刚性的 GUI 元素检测，难以适应不断变化的应用界面。得益于大规模多模态数据集 [211]，[212]，[213]，LLM 的进步使模型能够将文本和视觉信号统一表示。诸如 UGround[80]、Ferret-UI[101] 和 UI-Hawk[95] 等系统擅长将自然语言描述与屏幕元素对应，能够动态调整以适应界面演变。此外，SeeClick[105] 和 ScreenAI[67] 展示了直接从截图而非纯文本元数据学习，进一步提升了适应性。通过整合视觉线索与用户语言，基于 LLM 的代理能够更灵活地应对各种 UI 设计和交互场景。

Reasoning and Decision Making. LLMs also enable advanced reasoning and decision-making by combining language, visual context, and historical user interactions. Pretraining on broad corpora equips these models with the capacity for complex reasoning [214], [215], multi-step planning [216], [217], and context-aware adaptation [218], [219]. MobileAgent-V2 [52], for instance, introduces a specialized planning agent to track task progress while a decision agent optimizes actions. Auto-GUI [66] applies a multimodal chain-of-action approach that accounts for both previous and forthcoming steps, and SteP [79] uses stacked LLM modules to solve diverse web tasks. Similarly, MobileGPT [61] leverages an app memory system to minimize repeated mistakes and bolster adaptability. Such architectures demonstrate higher success rates in complex phone operations, reflecting a new level of autonomy in orchestrating tasks that previously demanded handcrafted scripts.

推理与决策。LLM 还通过结合语言、视觉上下文和历史用户交互，实现了高级推理和决策能力。广泛语料的预训练赋予这些模型复杂推理 [214]，[215]、多步骤规划 [216]，[217] 和上下文感知适应 [218]，[219] 的能力。例如，MobileAgent-V2[52] 引入了专门的规划代理以跟踪任务进度，同时决策代理优化操作。Auto-GUI[66] 采用多模态动作链方法，考虑前后步骤，SteP[79] 利用堆叠的 LLM 模块解决多样化的网页任务。类似地，MobileGPT[61] 利用应用记忆系统减少重复错误并增强适应性。这些架构在复杂手机操作中表现出更高的成功率，体现了在以往需手工编写脚本的任务中实现的新自主水平。

Overall, LLMs are transforming phone automation by reinforcing semantic understanding, expanding multi-modal perception, and enabling sophisticated decision-making strategies. The scaling laws observed in datasets like An-droidControl [140] reinforce the notion that a larger volume and diversity of demonstrations consistently elevate model accuracy. As these techniques mature, LLM-driven phone GUI agents continue to redefine how users interact with mobile devices, ultimately paving the way for a more seamless and user-centric automation experience.

总体而言，LLM 正在通过强化语义理解、扩展多模态感知和实现复杂决策策略，变革手机自动化。在如 AndroidControl[140] 等数据集上观察到的规模定律强化了这样一个观点: 更多样化和更大规模的示范数据持续提升模型准确性。随着这些技术的成熟，基于 LLM 的手机 GUI 代理不断重新定义用户与移动设备的交互方式，最终为更无缝、更以用户为中心的自动化体验铺平道路。

2.4 Emerging Commercial Applications

2.4 新兴商业应用

The integration of LLMs has enabled novel commercial applications that leverage phone automation, offering innovative solutions to real-world challenges. This subsection highlights several prominent cases, presented in chronological order based on their release dates, where LLM-based GUI agents are reshaping user experiences, improving efficiency, and providing personalized services.

LLM 的集成催生了利用手机自动化的新型商业应用，提供了应对现实挑战的创新解决方案。本小节重点介绍若干重要案例，按发布时间顺序排列，展示基于 LLM 的 GUI 代理如何重塑用户体验、提升效率并提供个性化服务。

Apple Intelligence. On June 11, 2024, Apple introduced its personal intelligent system, Apple Intelligence [5] , seamlessly integrating AI capabilities into iOS, iPadOS, and macOS. It enhances communication, productivity, and focus features through intelligent summarization, priority notifications, and context-aware replies. For instance, Apple Intelligence can summarize long emails, transcribe and interpret call recordings, and generate personalized images or ”Genmoji.” A key aspect is on-device processing, which ensures user privacy and security. By enabling the system to operate directly on the user’s device, Apple Intelligence safeguards personal information while providing an advanced, privacy-preserving phone automation experience.

苹果智能。2024 年 6 月 11 日，苹果推出了其个人智能系统 Apple Intelligence [5]，将人工智能功能无缝集成到 iOS、iPadOS 和 macOS 中。它通过智能摘要、优先通知和上下文感知回复，提升了通信、生产力和专注功能。例如，Apple Intelligence 可以总结长邮件，转录并解读通话录音，生成个性化图像或“Genmoji”。其关键特点是设备端处理，确保用户隐私和安全。通过使系统直接在用户设备上运行，Apple Intelligence 在提供先进且保护隐私的手机自动化体验的同时，保障了个人信息安全。

vivo PhoneGPT. On October 10, 2024, vivo unveiled Origi-nOS 5 [6], its newest mobile operating system, featuring an AI agent ability named PhoneGPT. By harnessing large language models, PhoneGPT can understand user instructions, preferences, and on-screen information, autonomously engaging in dialogues and detecting GUI states to operate the smart-phone. Notably, it allows users to order coffee or takeout with ease and can even carry out a full phone reservation process at a local restaurant through extended conversations. By integrating the capabilities of large language models with native system states and APIs, PhoneGPT illustrates the great potential of phone GUI agents.

vivo PhoneGPT。2024 年 10 月 10 日，vivo 发布了其最新移动操作系统 OriginOS 5 [6]，其中包含名为 PhoneGPT 的 AI 代理功能。PhoneGPT 利用大型语言模型，能够理解用户指令、偏好及屏幕信息，自主进行对话并检测图形用户界面状态以操作智能手机。值得注意的是，它允许用户轻松点咖啡或外卖，甚至能通过延续对话完成本地餐厅的完整电话预订流程。通过将大型语言模型的能力与本地系统状态和 API 集成，PhoneGPT 展示了手机图形界面代理的巨大潜力。

Honor YOYO Agent. Released on October 24, 2024, the Honor YOYO Agent [7] exemplifies an phone automation assistant that adapts to user habits and complex instructions. With just one voice or text command, YOYO can automate multi-step processes-such as comparing prices to secure discounts when shopping, automatically filling out forms, ordering beverages aligned with user preferences, or silencing notifications during online meetings. By learning from user behaviors, YOYO reduces the complexity of human-device interaction, offering a more effortless and intelligent phone experience.

荣耀 YOYO 代理。2024 年 10 月 24 日发布的荣耀 YOYO 代理 [7] 是一个适应用户习惯和复杂指令的手机自动化助手。只需一条语音或文本命令，YOYO 即可自动执行多步骤流程——如比价以获取购物折扣、自动填写表单、根据用户偏好订购饮品，或在在线会议期间静音通知。通过学习用户行为，YOYO 降低了人机交互的复杂性，提供了更轻松智能的手机使用体验。

Anthropic Claude Computer Use. On October 22, 2024, Anthropic unveiled the Computer Use feature for its Claude 3.5 Sonnet model8. This feature allows an AI agent to interact with a computer as if a human were operating it, observing screenshots, moving the virtual cursor, clicking buttons, and typing text. Instead of requiring specialized environment adaptations, the AI can ”see” the screen and perform actions that humans would, bridging the gap between language-based instructions and direct computer operations. Although initial performance is still far below human proficiency, this represents a paradigm shift in human-computer interaction. By teaching AI to mimic human tool usage, Anthropic reframes the challenge from ”tool adaptation for models” to ”model adaptation to existing tools.” Achieving balanced performance, security, and cost-effectiveness remains an ongoing endeavor.

Anthropic Claude 计算机使用。2024 年 10 月 22 日，Anthropic 发布了其 Claude 3.5 Sonnet 模型的 Computer Use 功能 8。该功能允许 AI 代理像人类操作一样与计算机交互，观察屏幕截图、移动虚拟光标、点击按钮和输入文本。AI 无需专门环境适配，即可“看见”屏幕并执行人类操作，弥合了基于语言指令与直接计算机操作之间的鸿沟。尽管初期性能远低于人类水平，但这代表了人机交互的范式转变。通过教 AI 模仿人类工具使用，Anthropic 将挑战从“为模型适配工具”转变为“让模型适应现有工具”。实现性能、安全与成本的平衡仍是持续努力的方向。

---

5. https://www.apple.com/apple-intelligence/.

5. https://www.apple.com/apple-intelligence/.

6. https://www.vivo.com.cn/originos

6. https://www.vivo.com.cn/originos

7. https://www.honor.com/cn/magic-os/.

7. https://www.honor.com/cn/magic-os/.

8. https://www.anthropic.com/news/3-5-models-and-computer-use

8. https://www.anthropic.com/news/3-5-models-and-computer-use

---

Zhipu.AI AutoGLM. On October 25, 2024, Zhipu.AI introduced AutoGLM [118], an intelligent agent that simulates human operations on smartphones. With simple text or voice commands, AutoGLM can like and comment on social media posts, purchase products, book train tickets, or order takeout. Its capabilities extend beyond mere API calls-AutoGLM can navigate interfaces, interpret visual cues, and execute tasks that mirror human interaction steps. This approach streamlines daily tasks and demonstrates the versatility and practicality of LLM-driven phone automation in commercial applications.

智谱 AI AutoGLM。2024 年 10 月 25 日，智谱 AI 推出了 AutoGLM[118]，一款模拟人类在智能手机上操作的智能代理。通过简单的文本或语音命令，AutoGLM 可以点赞和评论社交媒体帖子、购买商品、预订火车票或点外卖。其能力不仅限于 API 调用——AutoGLM 能够导航界面、解读视觉线索并执行模仿人类交互步骤的任务。这种方法简化了日常任务，展示了基于大型语言模型的手机自动化在商业应用中的多样性和实用性。

These emerging commercial applications-from Apple’s privacy-focused on-device intelligence to vivo’s PhoneGPT, Honor’s YOYO agent, Anthropic’s Computer Use, and Zhipu.AI’s AutoGLM-showcase how LLM-based agents are transcending traditional user interfaces. They enable more natural, efficient, and personalized human-device interactions. As models and methods continue to evolve, we can anticipate even more groundbreaking applications, further integrating AI into the fabric of daily life and professional workflows.

这些新兴的商业应用——从苹果注重隐私的设备端智能，到 vivo 的 PhoneGPT、荣耀的 YOYO 代理、Anthropic 的 Computer Use 以及智谱 AI 的 AutoGLM——展示了基于大型语言模型的代理如何超越传统用户界面，实现更自然、高效和个性化的人机交互。随着模型和方法的不断发展，我们可以期待更多突破性应用，进一步将人工智能融入日常生活和专业工作流程的方方面面。

3 FRAMEWORKS AND COMPONENTS OF PHONE GUI AGENTS

3 手机图形界面代理的框架与组件

MLLM-powered phone GUI agents can be designed using different architectural paradigms and components, ranging from straightforward, single-agent systems [48], [50], [51], [53], [57] to more elaborate multi-agent [52], [55], [220] or multi-stage [47], [80], [82] approaches. A fundamental scenario involves a single agent that operates incrementally, without precomputing an entire action sequence from the outset. Instead, the agent continuously observes the dynamically changing mobile environment-where available UI elements, device states, and relevant contextual factors may shift in unpredictable ways-and cannot be exhaustively enumerated in advance. As a result, the agent must adapt its strategy step-by-step, making decisions based on the current situation rather than following a fixed plan. This iterative decision-making process can be effectively modeled using a Partially Observable Markov Decision Process (POMDP), a well-established framework for handling sequential decision-making under uncertainty [221], [222]. By modeling the task as a POMDP, we capture its dynamic nature, the impossibility of pre-planning all actions, and the necessity of adjusting the agent's approach at each decision point.

基于多模态大语言模型 (MLLM) 的手机图形用户界面 (GUI) 代理可以采用不同的架构范式和组件设计，范围从简单的单代理系统 [48]，[50]，[51]，[53]，[57] 到更复杂的多代理 [52]，[55]，[220] 或多阶段 [47]，[80]，[82] 方法。一个基本场景涉及一个单一代理，该代理以增量方式操作，而非从一开始就预先计算完整的动作序列。相反，代理持续观察动态变化的移动环境——其中可用的 UI 元素、设备状态和相关上下文因素可能以不可预测的方式变化，且无法提前穷举。因此，代理必须逐步调整其策略，基于当前情境做出决策，而非遵循固定计划。这种迭代决策过程可以通过部分可观测马尔可夫决策过程 (POMDP，Partially Observable Markov Decision Process) 有效建模，POMDP 是处理不确定性下序列决策的成熟框架 [221]，[222]。通过将任务建模为 POMDP，我们捕捉了其动态特性、无法预先规划所有动作的事实，以及在每个决策点调整代理策略的必要性。

As illustrated in Figure 4, consider a simple example: the agent's goal is to order a latte through the Starbucks app. The app's interface may vary depending on network latency, promotions displayed, or the user's last visited screen. The agent cannot simply plan all steps in advance; it must observe the current screen, identify which UI elements are present, and then choose an action (like tapping the Starbucks icon, swiping to a menu, or selecting the latte). After each action, the state changes, and the agent re-evaluates its options. This dynamic, incremental decision-making is precisely why POMDPs are a suitable framework. In the POMDP formulation for phone automation:

如图 4 所示，考虑一个简单示例: 代理的目标是通过星巴克应用程序订购一杯拿铁。应用界面可能因网络延迟、促销活动展示或用户上次访问的屏幕而有所不同。代理不能简单地提前规划所有步骤；它必须观察当前屏幕，识别存在的 UI 元素，然后选择一个动作 (如点击星巴克图标、滑动到菜单或选择拿铁)。每执行一个动作，状态都会发生变化，代理重新评估其选项。这种动态、增量的决策正是 POMDP 框架适用的原因。在手机自动化的 POMDP 表述中:

States (S). At each decision point, the agent's perspective is described as a state, a comprehensive snapshot of all relevant information that could potentially influence the decision-making process. This state encompasses the current UI information (e.g., screenshots, UI trees, OCR-extracted text, icons), the phone's own status (network conditions, battery level, location), and the task context (the user's goal—"order a latte"—and the agent's progress toward it). The state S_t represents the complete, underlying situation of the environment at time t , which may not be directly observable in its entirety.

状态 (S)。在每个决策点，代理的视角被描述为一个状态，即所有可能影响决策过程的相关信息的全面快照。该状态包括当前的 UI 信息 (例如，屏幕截图、UI 树、OCR 提取的文本、图标)、手机自身状态 (网络状况、电池电量、位置) 以及任务上下文 (用户目标——“订购拿铁”——及代理的进展)。状态 S_t 表示时间点 t 环境的完整、潜在情况，该情况可能无法被完全直接观测到。

Actions (A) . Given the state S_t at time t , the agent selects from available actions (taps, swipes, typing text, launching apps) that influence the subsequent state. The details of how phone GUI agents make decisions are introduced in §3.2, and the design of the action space is discussed in §3.3.

动作 (A) 。在时间点 t 给定状态 S_t 时，代理从可用动作中选择 (点击、滑动、输入文本、启动应用)，这些动作会影响后续状态。手机 GUI 代理如何做出决策的细节在第 3.2 节介绍，动作空间的设计在第 3.3 节讨论。

Transition Dynamics $(P(s' \mid s, a))$. When the agent executes an action a_t at time t , it leads to a new state S_{t+1} . Some transitions may be deterministic (e.g., tapping a known button reliably opens a menu), while others are uncertain (e.g., network delays, unexpected pop-ups). Mathematically, we have the transition probability $P(s' \mid s, a)$ which describes the likelihood of transitioning from state S_t to state S_{t+1} given action a_t .

转移动态 $(P(s' \mid s, a))$ 。当代理在时间点 t 执行动作 a_t 时，会导致新状态 S_{t+1} 。某些转移可能是确定性的 (例如，点击已知按钮可靠地打开菜单)，而其他则存在不确定性 (例如，网络延迟、意外弹窗)。数学上，我们有转移概率 $P(s' \mid s, a)$ ，描述在动作 a_t 下从状态 S_t 转移到状态 S_{t+1} 的可能性。

Observations (O) . The agent receives observations O_t at time t which are partial and imperfect reflections of the true state S_t . In the phone automation context, these observations could be, for example, a glimpse of the visible UI elements (not the entire UI tree), a brief indication of the network status (such as a signal icon without detailed connection parameters), or a partial view of the battery level indicator. These observations O_t provide the agent with some, but not all, of the information relevant to the state S_t . The agent must infer and make decisions based on these limited observations, attempting to reach the desired goal state despite the partial observability. The details of phone GUI agent perception are discussed in §3.1.

观测 (O)。代理在时间点 t 接收观测 O_t，这些观测是对真实状态 S_t 的部分且不完美的反映。在手机自动化场景中，这些观测可能是可见 UI 元素的一瞥 (而非完整 UI 树)、网络状态的简要指示 (如信号图标而非详细连接参数) 或电池电量指示的部分视图。这些观测 O_t 为代理提供了部分但非全部与状态 S_t 相关的信息。代理必须基于这些有限观测进行推断和决策，尽管存在部分可观测性，仍努力达到期望的目标状态。手机 GUI 代理感知的细节在第 3.1 节讨论。

Under this POMDP-based paradigm, the agent aims to make decisions that lead to the goal state by observing the current state and choosing appropriate actions. It continuously re-evaluates its strategy as conditions evolve, promoting real-time responsiveness and dynamic adaptation. The agent observes the state S_t at time t , chooses an action a_t , and then based on the resulting observation O_{t+1} and new state S_{t+1} , refines its strategy.

在基于部分可观测马尔可夫决策过程 (POMDP) 的范式下，智能体旨在通过观察当前状态并选择适当的动作来实现目标状态。随着条件的变化，它不断重新评估策略，促进实时响应和动态适应。智能体在时间 t 观察状态 S_t，选择动作 a_t，然后根据得到的观察 O_{t+1} 和新状态 S_{t+1}，优化其策略。

As illustrated in Figure 5, frameworks of phone GUI agents aim to integrate perception, reasoning, and action capabilities into cohesive agents that can interpret user intentions, understand complex UI states, and execute appropriate operations within mobile environment. By examining these frameworks, we can identify best practices, guide future advancements, and choose the right approach for various applications and contexts.

如图 5 所示，手机 GUI 智能体框架旨在将感知、推理和行动能力整合为统一的智能体，能够解读用户意图，理解复杂的界面状态，并在移动环境中执行适当操作。通过研究这些框架，我们可以识别最佳实践，指导未来发展，并为不同应用和场景选择合适的方法。

To address limitations in adaptability and scalability, §3.4 introduces multi-agent frameworks, where specialized agents collaborate, enhance efficiency, and handle more diverse tasks in parallel. Finally, §3.5 presents the Plan-Then-Act Framework, which explicitly separates the planning phase from the execution phase. This approach allows agents to refine their conceptual plans before acting, potentially improving both accuracy and robustness.

为解决适应性和可扩展性的局限，§3.4 介绍了多智能体框架，其中专门化智能体协作，提高效率，并并行处理更多样化的任务。最后，§3.5 提出了“先规划后执行”框架，该方法明确区分规划阶段与执行阶段，使智能体在行动前优化其概念性计划，可能提升准确性和鲁棒性。

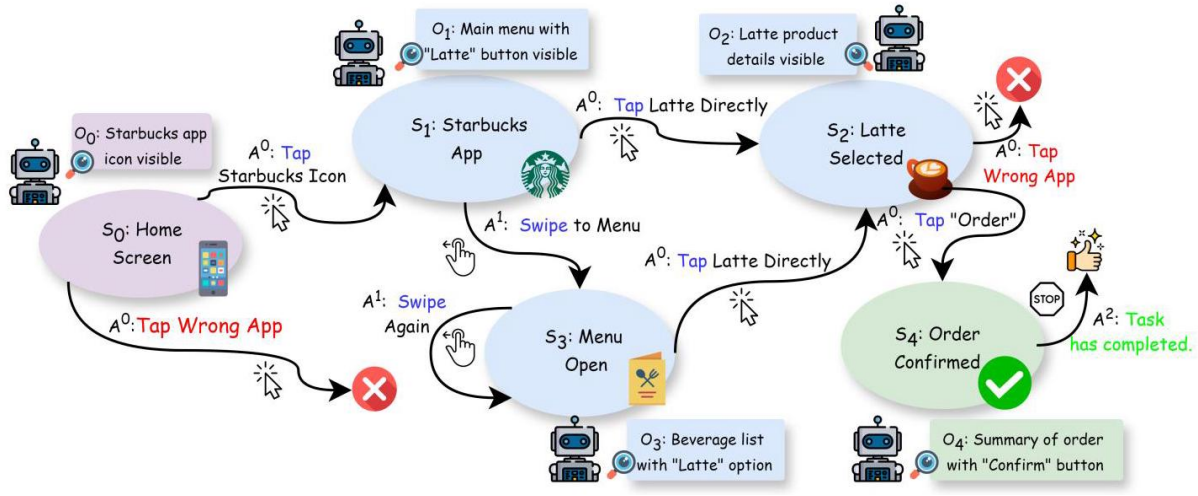

Fig. 4: POMDP model for ordering a latte. Each circle represents a state (e.g., Home Screen, App Homepage, Latte Details Page, Customize Order, Order Confirmation, Order Complete). The agent starts at the initial state S_0 (Home Screen) and makes decisions at each step (e.g., tapping the Starbucks app icon, selecting the "Latte" button, viewing latte details). Due to partial observability, the agent receives limited information at each decision point (e.g., O_0 : Starbucks app icon visible, O_1 : "Latte" button visible, O_2 : Latte product details visible). Some actions correctly advance towards the goal, while others may cause errors requiring corrections. The final goal is to confirm the order.

图 4: 点单拿铁的 POMDP 模型。每个圆圈代表一个状态 (例如，主屏幕、应用首页、拿铁详情页、定制订单、订单确认、订单完成)。智能体从初始状态 S_0 (主屏幕) 开始，在每一步做出决策 (例如，点击星巴克应用图标，选择“拿铁”按钮，查看拿铁详情)。由于部分可观测性，智能体在每个决策点获得有限信息 (例如，O_0 : 星巴克应用图标可见，O_1 :“拿铁”按钮可见，O_2 : 拿铁产品详情可见)。部分动作正确推进目标，而其他动作可能导致错误需纠正。最终目标是确认订单。

3.1 Perception in Phone GUI Agents

3.1 手机 GUI 智能体中的感知

Perception is a fundamental component of the basic framework for MLLM-powered phone GUI agents. It is responsible for capturing and interpreting the state of the mobile environment, enabling the agent to understand the current context and make informed decisions. In the overall pipeline, perception serves as the initial step in the POMDP, providing the necessary input for the reasoning and action modules to operate effectively.

感知是基于多模态大语言模型 (MLLM) 驱动的手机 GUI 智能体基本框架的核心组成部分。它负责捕捉和解读移动环境状态，使智能体理解当前上下文并做出明智决策。在整体流程中，感知作为 POMDP 的初始步骤，为推理和行动模块提供必要输入，确保其有效运行。

3.1.1 UI Information Perception

3.1.1 界面信息感知

UI information is crucial for agents to interact seamlessly with the mobile interface. It can be further categorized into UI tree-based and screenshot-based approaches, supplemented by techniques like Set-of-Marks (SoM) and Icon & OCR enhancements.

界面信息对于智能体与移动界面无缝交互至关重要。它可进一步分为基于界面树和基于截图的方法，并辅以标记集 (Set-of-Marks，SoM) 及图标与光学字符识别 (OCR) 增强技术。

UI tree is a structured, hierarchical representation of the UI elements on a mobile screen [223], [224]. Each node in the UI tree corresponds to a UI component, containing attributes such as class type, visibility, and resource identifiers. [9] Early datasets like PixelHelp [132], MoTIF [133], and UIBert [134] utilized UI tree data to enable tasks such as mapping natural language instructions to UI actions and performing interactive visual environment

interactions. DroidBot-GPT [53] was the first work to investigate how pre-trained language models can be applied to app automation without modifying the app or the model. DroidBot-GPT uses the UI tree as its primary perception information. The challenge lies in converting the structured UI tree into a format that LLMs can process effectively. DroidBot-GPT addresses this by transforming the UI tree into natural language sentences. Specifically, it extracts all user-visible elements, generates prompts like "A view <name>that can..." for each element, and combines them into a cohesive description of the current UI state. This approach mitigates the issue of excessively long and complex UI trees by presenting the information in a more natural and concise format suitable for LLMs. Subsequent developments, such as Enabling Conversational Interaction [57] and AutoDroid [50], further refined this approach by representing the view hierarchy as HTML. Enabling Conversational Interaction introduces a method to convert the view hierarchy into HTML syntax, mapping Android UI classes to corresponding HTML tags and preserving essential attributes such as class type, text, and resource identifiers. This representation aligns closely with the training data distribution of LLMs, enhancing their ability to perform few-shot learning and improving overall UI understanding. AutoDroid extends this work by developing a GUI parsing module that converts the GUI into a simplified HTML representation using specific HTML tags like <button>, <checkbox>, <scroller>, <input>, and <p>. Additionally, AutoDroid implements automatic scrolling of scrollable components to ensure that comprehensive UI information is available to the LLM, thereby enhancing decision-making accuracy and reducing computational overhead. Furthermore, LLMPA [58] employs object detection models to comprehend page layouts and optimizes the grouping of UI elements for potential actions. This approach reduces redundant information in the UI tree, thereby enhancing the accuracy and speed of decision making. Similar to this approach, the TOL Agent [59] introduces a variant of the UI tree, known as the Hierarchical Layout Tree, to represent the hierarchical layout of screen captures. In this tree, nodes represent different levels of regions. This structure, combined with a trained DINO model, aids in generating more accurate screen descriptions for MLLM.

UI 树是移动屏幕上 UI 元素的结构化层级表示 [223]，[224]。UI 树中的每个节点对应一个 UI 组件，包含类类型、可见性和资源标识符等属性。[9] 早期的数据集如 PixelHelp[132]、MoTIF[133] 和 UIBert[134] 利用 UI 树数据实现了将自然语言指令映射到 UI 操作及执行交互式视觉环境交互等任务。DroidBot-GPT[53] 是首个研究如何在不修改应用或模型的情况下，将预训练语言模型应用于应用自动化的工作。DroidBot-GPT 以 UI 树作为其主要感知信息。挑战在于如何将结构化的 UI 树转换为大语言模型 (LLMs) 能够有效处理的格式。DroidBot-GPT 通过将 UI 树转化为自然语言句子来解决这一问题。具体而言，它提取所有用户可见元素，为每个元素生成类似“A view <name>that can...”的提示，并将它们组合成当前 UI 状态的连贯描述。这种方法通过以更自然简洁的格式呈现信息，缓解了 UI 树过长过复杂的问题，更适合 LLMs 处理。后续发展如 Enabling Conversational Interaction[57] 和 AutoDroid[50] 进一步完善了该方法，将视图层级表示为 HTML。Enabling Conversational Interaction 引入了一种将视图层级转换为 HTML 语法的方法，将 Android UI 类映射到对应的 HTML 标签，并保留类类型、文本和资源标识符等关键属性。这种表示方式与 LLMs 的训练数据分布高度契合，增强了其少样本学习能力并提升整体 UI 理解。AutoDroid 在此基础上开发了 GUI 解析模块，使用 <button>、<checkbox>、<scroller>、<input> 和 <p> 等特定 HTML 标签将 GUI 转换为简化的 HTML 表示。此外，AutoDroid 实现了对可滚动组件的自动滚动，确保 LLM 获取全面的 UI 信息，从而提升决策准确性并降低计算开销。此外，LLMPA[58] 采用目标检测模型理解页面布局，优化 UI 元素的分组以支持潜在操作。这种方法减少了 UI 树中的冗余信息，提升了决策的准确性和速度。与此类似，TOL Agent[59] 引入了一种 UI 树变体，称为层级布局树，用于表示屏幕截图的层级布局。该树中节点代表不同层级的区域。该结构结合训练好的 DINO 模型，有助于为多模态大语言模型 (MLLM) 生成更准确的屏幕描述。

9. https://developer.android.com/reference/android/view/View.

9. https://developer.android.com/reference/android/view/View.

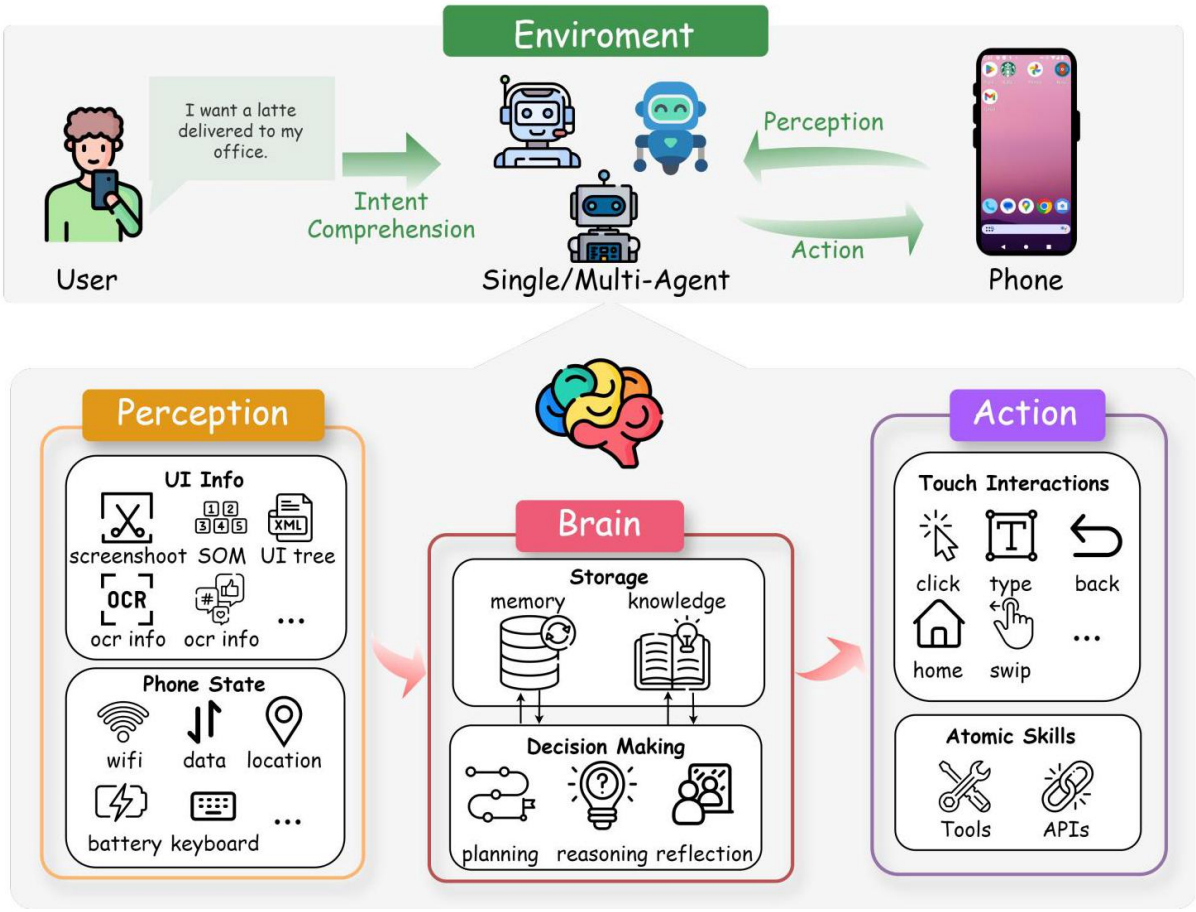

Fig. 5: Overview of MLLM-powered phone GUI agent framework. The user's intent, expressed through natural language, is mapped to UI operations. By perceiving UI information and phone state(§3.1), the agent leverages stored knowledge and memory to plan, reason, and reflect (§3.2). Finally, it executes actions to fulfill the user's goals(§3.3).

图 5: 基于 MLLM 的手机 GUI 代理框架概览。用户通过自然语言表达意图，该意图被映射为 UI 操作。代理通过感知 UI 信息和手机状态 (§3.1)，利用存储的知识和记忆进行规划、推理和反思 (§3.2)。最终执行操作以实现用户目标 (§3.3)。

Screenshots provide a visual snapshot of the current UI, capturing the appearance and layout of UI elements. Unlike UI trees, which require API access and can become unwieldy with complex hierarchies, screenshots offer a more flexible and often more comprehensive representation of the UI. Additionally, UI trees present challenges such as missing or overlapping controls and the inability to directly reference UI elements programmatically, making screenshots a more practical and user-friendly alternative for quickly assessing and sharing the state of a user interface. Auto-GUI [66] introduced a multimodal agent that relies on screenshots for GUI control, eliminating the dependency on UI trees. This approach allows the agent to interact with the UI directly through visual perception, enabling more natural and human-like interactions. Auto-GUI employs a chain-of-action technique that uses

both previously executed actions and planned future actions to guide decision-making, achieving high action type prediction accuracy and efficient task execution. Following Auto-GUI, a series of multimodal solutions emerged, including MM-Navigator [60], CogA-gent [46], AppAgent [48], VisionTasker [85], MobileGPT [61], GUI Narrator [62], MobileVLM [63], AdaptAgent [225], WebVoyager [226] and Steward [227]. These frameworks leverage screenshots in combination with supplementary information to enhance UI understanding and interaction capabilities.

截图提供了当前 UI 的视觉快照，捕捉了 UI 元素的外观和布局。与需要 API 访问且在复杂层级下可能变得难以管理的 UI 树不同，截图提供了更灵活且通常更全面的 UI 表示。此外，UI 树存在控件缺失或重叠、无法程序化直接引用 UI 元素等问题，使得截图成为快速评估和共享用户界面状态的更实用且用户友好的替代方案。Auto-GUI[66] 引入了一种多模态代理，依赖截图进行 GUI 控制，消除了对 UI 树的依赖。该方法使代理能够通过视觉感知直接与 UI 交互，实现更自然、更类人的交互。Auto-GUI 采用链式动作技术，结合已执行动作和计划中的未来动作指导决策，实现了高动作类型预测准确率和高效的任务执行。继 Auto-GUI 之后，出现了一系列多模态解决方案，包括 MM-Navigator[60]、CogAgent[46]、AppAgent[48]、VisionTasker[85]、MobileGPT[61]、GUI Narrator[62]、MobileVLM[63]、AdaptAgent[225]、WebVoyager[226] 和 Steward[227]。这些框架结合截图与辅助信息，提升了 UI 理解和交互能力。

Set-of-Mark (SoM) is a prompting technique used to annotate screenshots with OCR, icon, and UI tree information, thereby enriching the visual data with textual descriptors [228]. For example, MM-Navigator [60] uses SoM to label UI elements with unique identifiers, allowing the LLM to reference and interact with specific components more effectively. This method has been widely adopted in subsequent works such as AppAgent [48], VisionDroid [54], OmniParser [56] and VisualWebArena [229], which utilize SoM to enhance the agent's ability to interpret and act upon UI elements based on visual, textual, and structural cues.

Set-of-Mark (SoM) 是一种用于通过 OCR、图标和 UI 树信息标注截图的提示技术，从而用文本描述丰富视觉数据 [228]。例如，MM-Navigator [60] 使用 SoM 为 UI 元素标注唯一标识符，使大型语言模型 (LLM) 能够更有效地引用和交互特定组件。该方法已被后续工作广泛采用，如 AppAgent [48]、VisionDroid [54]、OmniParser [56] 和 VisualWebArena [229]，它们利用 SoM 增强代理基于视觉、文本和结构线索解释和操作 UI 元素的能力。

Icon & OCR enhancements provide additional layers of information that complement the visual data, enabling more precise action decisions. For instance, Mobile-Agent-v2 [52] integrates OCR and icon data with screenshots to provide a richer context for the LLM, allowing it to interpret and execute more complex instructions that require understanding both text and visual icons. Icon & OCR enhancements are employed in various works, including VisionTasker [85], MobileGPT [61], OmniParser [56], and WindowsAgentArena [230], to improve the accuracy and reliability of phone GUI agents.

图标和 OCR 增强提供了补充视觉数据的额外信息层，使动作决策更为精准。例如，Mobile-Agent-v2 [52] 将 OCR 和图标数据与截图结合，为 LLM 提供更丰富的上下文，使其能够理解并执行需要同时理解文本和视觉图标的复杂指令。图标和 OCR 增强被应用于多个工作中，包括 VisionTasker [85]、MobileGPT [61]、OmniParser [56] 和 WindowsAgentArena [230]，以提升手机 GUI 代理的准确性和可靠性。

3.1.2 Phone State Perception

3.1.2 手机状态感知

Phone state information, such as keyboard status and location data, further contextualizes the agent's interactions. For example, Mobile-Agent-v2 [52] uses keyboard status to determine when text input is required. Location data, while not currently utilized, represents a potential form of phone state information that could be used to recommend nearby services or navigate to specific addresses. This additional state information enhances the agent's ability to perform context-aware actions, making interactions more intuitive and efficient.

手机状态信息，如键盘状态和位置信息，进一步为代理的交互提供上下文。例如，Mobile-Agent-v2 [52] 利用键盘状态判断何时需要文本输入。位置信息虽尚未被利用，但作为一种潜在的手机状态信息，可用于推荐附近服务或导航至特定地址。这些额外的状态信息增强了代理执行上下文感知操作的能力，使交互更直观高效。

The perception information gathered through UI trees, screenshots, SoM, OCR, and phone state is converted into prompt tokens that the LLM can process. This conversion is crucial for enabling seamless interaction between the perception module and the reasoning and action modules. Detailed methodologies for transforming perception data into prompt formats are discussed in §4.1.

通过 UI 树、截图、SoM、OCR 和手机状态收集的感知信息被转换为 LLM 可处理的提示令牌。这一转换对于实现感知模块与推理及动作模块之间的无缝交互至关重要。将感知数据转化为提示格式的详细方法见 §4.1。

3.2 Brain in Phone GUI Agents

3.2 手机 GUI 代理中的“大脑”

The brain of an LLM-based phone automation agent is its cognitive core, primarily constituted by a LLM. The LLM serves as the agent's reasoning and decision-making center, enabling it to interpret inputs, generate appropriate responses, and execute actions within the mobile environment [231], [232]. Leveraging the extensive knowledge embedded within LLMs, agents benefit from advanced language understanding, contextual awareness, and the ability to generalize across diverse tasks and scenarios.

基于大型语言模型 (LLM) 的手机自动化代理的“大脑”是其认知核心，主要由 LLM 构成。LLM 作为代理的推理和决策中心，使其能够解释输入、生成适当响应并在移动环境中执行操作 [231], [232]。借助 LLM 中蕴含的丰富知识，代理具备高级语言理解、上下文感知及跨多样任务和场景泛化的能力。

3.2.1 Storage

3.2.1 存储

Storage encompasses both memory and knowledge, which are critical for maintaining context and informing the agent's decision-making processes.

存储包括记忆和知识，这对维持上下文和支持代理决策过程至关重要。

Memory refers to the agent's ability to retain information from past interactions with users and the environment [44]. This is particularly useful for cross-application operations, where continuity and coherence are essential for completing multi-step tasks. For example, Mobile-Agent-v2 [52] integrates a memory unit that records task-related focus content from historical screens. This memory is accessed by the decision-making module when generating operations, ensuring that the agent can reference and update relevant information dynamically. The Self-MAP framework [233] establishes a memory repository based on the history of conversational interactions. It utilizes a multifaceted matching approach to retrieve the top-K memory snippets that are semantically relevant to the current dialogue state and have similar trajectories. This assists the agent in effectively utilizing limited context space during multi-turn interactions, thereby enhancing its ability to comprehend and execute user instructions.

记忆指代理保留与用户及环境过去交互信息的能力 [44]。这对于跨应用操作尤为重要，因其需要连续性和连贯性以完成多步骤任务。例如，Mobile-Agent-v2 [52] 集成了一个记忆单元，记录历史屏幕中的任务相关焦点内容。决策模块在生成操作时访问该记忆，确保代理能够动态引用和更新相关信息。Self-MAP 框架 [233] 基于对话历史建立记忆库，采用多维匹配方法检索与当前对话状态语义相关且轨迹相似的前 K 条记忆片段，帮助代理在多轮交互中有效利用有限上下文空间，提升理解和执行用户指令的能力。

Knowledge pertains to the agent's understanding of phone automation tasks and the functionalities of various apps. This knowledge can originate from multiple sources:

知识指代理对手机自动化任务及各类应用功能的理解。该知识可来源于多种途径:

- Pre-trained Knowledge. LLMs are inherently equipped with a vast amount of general knowledge, including common-sense reasoning and familiarity with programming and markup languages such as HTML. This preexisting knowledge allows the agent to interpret and generate meaningful actions based on the UI representations.

  - 预训练知识。LLM 天生具备大量通用知识，包括常识推理及对编程和标记语言 (如 HTML) 的熟悉。这些先验知识使代理能够基于 UI 表示解释并生成有意义的操作。

- Domain-Specific Training. Some agents enhance their knowledge by training on phone automation-specific datasets. Works such as Auto-GUI [66], CogAgent [46], ScreenAI [67], CoCo-agent [71], and Ferret-UI [101] have trained LLMs on datasets tailored for mobile UI interactions, thereby improving their capability to understand and manipulate mobile interfaces effectively. For a more detailed discussion of knowledge acquisition through model training, see §4.2.

  - 领域特定训练。一些代理通过在手机自动化专用数据集上训练来增强知识。诸如 Auto-GUI [66]、CogAgent [46]、ScreenAI [67]、CoCo-agent [71] 和 Ferret-UI [101] 等工作，针对移动 UI 交互定制数据集训练 LLM，从而提升其理解和操作移动界面的能力。关于通过模型训练获取知识的详细讨论见 §4.2。

- Knowledge Injection. Agents can enhance their decision-making by incorporating knowledge derived from exploratory interactions and stored contextual information. This involves utilizing data collected during offline exploration phases or from observed human demonstrations to inform the LLM's reasoning process. For instance, AutoDroid [50] explores app functionalities and records UI transitions in a UI Transition Graph (UTG) memory, which are then used to generate task-specific prompts for the LLM. Similarly, AppAgent [48] compiles knowledge from autonomous interactions and human demonstrations into structured documents, enabling the LLM to make informed decisions based on comprehensive UI state information and task requirements. AppAgent v2 [64] introduces a more efficient mechanism for knowledge base construction and updating. It leverages Retrieval-Augmented Generation (RAG) technology to achieve real-time dynamic updates of knowledge base information. This significantly enhances the agent's adaptability in new environments.AppAgentX [65] introduces an evolutionary mechanism that enables dynamic learning from past interactions and replaces inefficient low-level operations with high-level actions. Other similar works include AdaptAgent [225], Mobile-Agent-V [68], LearnAct [155] and others.

- 知识注入。智能体可以通过整合来自探索交互和存储的上下文信息的知识来增强决策能力。这包括利用离线探索阶段收集的数据或观察到的人类示范，以辅助大语言模型 (LLM) 的推理过程。例如，AutoDroid [50] 探索应用功能并将界面转换记录在界面转换图 (UI Transition Graph, UTG) 记忆中，随后用于生成针对特定任务的提示给 LLM。类似地，AppAgent [48] 将自主交互和人类示范中汇编的知识整理成结构化文档，使 LLM 能够基于全面的界面状态信息和任务需求做出明智决策。AppAgent v2 [64] 引入了更高效的知识库构建与更新机制，利用检索增强生成 (Retrieval-Augmented Generation, RAG) 技术实现知识库信息的实时动态更新，显著提升了智能体在新环境中的适应能力。AppAgentX [65] 引入了进化机制，支持从过去交互中动态学习，并用高级动作替代低效的底层操作。其他类似工作包括 AdaptAgent [225]、Mobile-Agent-V [68]、LearnAct [155] 等。

3.2.2 Decision Making

3.2.2 决策制定

Decision Making is the process by which the agent determines the appropriate actions to perform based on the current perception and stored information [44]. The LLM processes the input prompts, which include the current UI state, historical interactions from memory, and relevant knowledge, to generate action sequences that accomplish the assigned tasks.

决策制定是智能体基于当前感知和存储信息确定适当行动的过程 [44]。LLM 处理输入提示，包括当前界面状态、记忆中的历史交互及相关知识，以生成完成指定任务的动作序列。

Planning involves devising a sequence of actions to achieve a specific task goal [44], [216]. Effective planning is essential for decomposing complex tasks into manageable steps and adapting to changes in the environment. For instance, Mobile-Agent-v2 [52] incorporates a planning agent that generates task progress based on historical operations, ensuring effective operation generation by the decision agent. Additionally, approaches like Dynamic Planning of Thoughts (D-PoT) have been proposed to dynamically adjust plans based on environmental feedback and action history, significantly improving accuracy and adaptability in task execution [220]. Simultaneously, by reducing the number of calls to LLMs and employing a phased planning strategy, the agent can plan all actions in

a given state at once, thereby enhancing planning efficiency [234].

规划涉及设计一系列动作以实现特定任务目标 [44]，[216]。有效的规划对于将复杂任务分解为可管理步骤及适应环境变化至关重要。例如，Mobile-Agent-v2 [52] 集成了规划智能体，根据历史操作生成任务进展，确保决策智能体有效生成操作。此外，动态思维规划 (Dynamic Planning of Thoughts, D-PoT) 等方法被提出以基于环境反馈和动作历史动态调整计划，显著提升任务执行的准确性和适应性 [220]。同时，通过减少对 LLM 的调用次数并采用分阶段规划策略，智能体能够一次性规划给定状态下的所有动作，从而提高规划效率 [234]。

Reasoning enables the agent to interpret and analyze information to make informed decisions [235], [236], [237]. It involves understanding the context, evaluating possible actions, and selecting the most appropriate ones to achieve the desired outcome. By leveraging chain-of-thought(COT) [238], LLMs enhance their reasoning capabilities, allowing them to think step-by-step and handle intricate decision-making processes. This structured approach facilitates the generation of coherent and logical action sequences, ensuring that the agent can navigate complex UI interactions effectively. The best-first tree search algorithm is utilized in real-world environments to iteratively construct, explore, and prune trajectory graphs, thereby enhancing the reasoning and decision-making capabilities of agents. A value function serves as a reward signal to guide agents in conducting efficient searches [239]. Additionally, research indicates that LLMs to estimate the latent states of agents, in combination with reasoning methods, can further improve the agents' reasoning performance [240].

推理使智能体能够解释和分析信息以做出明智决策 [235]，[236]，[237]。这包括理解上下文、评估可能的动作并选择最合适的以达成预期结果。通过利用链式思维 (chain-of-thought, COT)[238]，LLM 增强了推理能力，能够逐步思考并处理复杂的决策过程。这种结构化方法促进生成连贯且逻辑合理的动作序列，确保智能体能有效应对复杂的界面交互。最佳优先树搜索算法被应用于实际环境中，迭代构建、探索和剪枝轨迹图，从而提升智能体的推理和决策能力。价值函数作为奖励信号，引导智能体进行高效搜索 [239]。此外，研究表明结合推理方法，LLM 估计智能体的潜在状态可进一步提升推理性能 [240]。

Reflection allows the agent to assess the outcomes of its actions and make necessary adjustments to improve performance [241]. It involves evaluating whether the executed actions meet the expected results and identifying any discrepancies or errors. For example, Mobile-Agent-v2 [52] includes a reflection agent that evaluates whether the decision agent's operations align with the task goals. If discrepancies are detected, the reflection agent generates appropriate remedial measures to correct the course of action. This continuous feedback loop enhances the agent's reliability and ensures that it can recover from unexpected states or errors during task execution. Furthermore, structured self-reflection identifies initial erroneous actions, which prevents agents from repeating the same mistakes. It also draws on reflective memory to avoid known unsuccessful actions [234]. Additionally, regular reflection through automated evaluation methods significantly enhances the performance of agents [242], [243].

反思使智能体能够评估其行动结果并进行必要调整以提升性能 [241]。这包括评估执行的动作是否达到预期效果，识别任何偏差或错误。例如，Mobile-Agent-v2 [52] 包含反思智能体，评估决策智能体的操作是否符合任务目标。如发现偏差，反思智能体会生成适当的补救措施以纠正行动方向。此持续反馈循环增强了智能体的可靠性，确保其能在任务执行中从意外状态或错误中恢复。此外，结构化自我反思识别初始错误动作，防止智能体重复同样错误，并利用反思记忆避免已知的失败动作 [234]。另外，通过自动化评估方法的定期反思显著提升了智能体的性能 [242]，[243]。

By integrating robust planning, advanced reasoning, and reflective capabilities, the Decision Making compo-

nent of the Brain ensures that MLLM-powered phone GUI agents can perform tasks intelligently and adaptively. These mechanisms enable the agents to handle a wide range of scenarios, maintain task continuity, and improve their performance over time through iterative learning and adjustment.

通过整合稳健的规划、先进的推理和反思能力，Brain 的决策制定组件确保基于多模态大语言模型 (MLLM) 的手机界面智能体能够智能且自适应地执行任务。这些机制使智能体能够应对多种场景，保持任务连续性，并通过迭代学习与调整不断提升性能。

3.3 Action in Phone GUI Agents

3.3 手机界面智能体中的动作

The Action component is a critical part of MLLM-powered phone GUI agents, responsible for executing decisions made by the Brain within the mobile environment. By bridging high-level commands generated by the LLM with low-level device operations, the agent can effectively interact with the phone's UI and system functionalities. Actions encompass a wide variety of operations, ranging from simple interactions like tapping a button to complex tasks such as launching applications or modifying device settings. Execution mechanisms leverage tools like Android's UI Automator [244], iOS's XCTest [245], or popular automation frameworks such as Appium [246] and Selenium [247], [248] to send precise commands to the phone. Through these mechanisms, the agent ensures that decisions are translated into tangible, reliable operations on the device.

动作组件是基于多模态大模型 (MLLM) 的手机图形用户界面代理中的关键部分，负责在移动环境中执行大脑 (Brain) 所做的决策。通过将大语言模型 (LLM) 生成的高级命令与底层设备操作连接起来，代理能够有效地与手机的用户界面和系统功能交互。动作涵盖了各种操作，从简单的点击按钮到启动应用程序或修改设备设置等复杂任务。执行机制利用了如 Android 的 UI Automator [244]、iOS 的 XCTest [245]，以及流行的自动化框架如 Appium [246] 和 Selenium [247]，[248] 等工具，向手机发送精确指令。通过这些机制，代理确保决策被转化为设备上的具体且可靠的操作。

The types of actions in phone GUI agents are diverse and can be broadly categorized based on their functionalities. Table 1 summarizes these actions, providing a clear overview of the operations agents can perform.

手机图形用户界面代理中的动作类型多样，可根据其功能大致分类。表 1 总结了这些动作，清晰展示了代理可以执行的操作。

The above categories reflect the key interactions required for phone automation. Touch interactions form the foundation of UI navigation, while gesture-based actions add flexibility for dynamic control. Typing and input enable text-based operations, whereas system operations and media controls extend the agent's capabilities to broader device functionalities. By combining these actions, phone GUI agents can achieve high accuracy and adaptability in executing user tasks, ensuring a seamless experience even in complex and dynamic environment.

上述类别反映了手机自动化所需的关键交互。触控交互构成了用户界面导航的基础，而基于手势的动作为动态控制增添了灵活性。输入和打字支持基于文本的操作，系统操作和媒体控制则扩展了代理对设备更广泛功能的掌控。通过结合这些动作，手机图形用户界面代理能够在执行用户任务时实现高精度和适应性，确保即使在复杂且动态的环境中也能提供流畅的体验。

3.4 Multi-Agent Framework

3.4 多代理框架

While single-agent frameworks based on LLMs have achieved significant progress in screen understanding and reasoning, they operate as isolated entities [249], [250], [251]. This isolation limits their flexibility and scalability in complex tasks that may require diverse, coordinated skills and adaptive capabilities. Single-agent systems may struggle with tasks that demand continuous adjustments based on real-time feedback, multi-stage decision-making, or specialized knowledge in different domains. Furthermore, they lack the ability to leverage shared knowledge or collaborate with other agents, reducing their effectiveness in dynamic environment [44], [52], [72], [73].

尽管基于大语言模型 (LLM) 的单代理框架在屏幕理解和推理方面取得了显著进展，但它们作为孤立实体运行 [249]，[250]，[251]。这种孤立性限制了它们在需要多样化、协调技能和适应能力的复杂任务中的灵活性和可扩展性。单代理系统在需要基于实时反馈进行持续调整、多阶段决策或不同领域专业知识的任务中可能表现不佳。此外，它们缺乏利用共享知识或与其他代理协作的能力，降低了在动态环境中的有效性 [44]，[52]，[72]，[73]。

Multi-agent frameworks address these limitations by facilitating collaboration among multiple agents, each with specialized functions or expertise [252], [253], [254], [255], [256], [257], [258], [259]. This collaborative approach enhances task efficiency, adaptability, and scalability, as agents can perform tasks in parallel or coordinate their actions based on their specific capabilities. As illustrated in Figure 6, multi-agent frameworks in phone automation can be categorized into two primary types: the Role-Coordinated Multi-Agent Framework and the Scenario-Based Task Execution Framework. These frameworks enable more flexible, efficient, and robust solutions in phone automation by either organizing agents based on general functional roles or dynamically assembling specialized agents according to specific task scenarios.

多代理框架通过促进多个具有专门功能或专业知识的代理之间的协作来解决这些限制 [252]，[253]，[254]，[255]，[256]，[257]，[258]，[259]。这种协作方式提升了任务效率、适应性和可扩展性，因为代理可以并行执行任务或根据各自的能力协调行动。如图 6 所示，手机自动化中的多代理框架可分为两大类: 基于角色协调的多代理框架和基于场景的任务执行框架。这些框架通过根据通用功能角色组织代理或根据特定任务场景动态组建专门代理，实现了手机自动化中更灵活、高效和稳健的解决方案。

TABLE 1: Types of actions in phone GUI agents

表 1: 手机图形用户界面代理中的动作类型

Action Type	Description
Touch Interactions Double Tap: Quickly tap twice to trigger an action. Long Press: Hold a touch for extended interaction, triggering contextual options or menus.	Tap: Select a specific UI element.
Gesture-Based Actions Pinch: Zoom in/out by bringing fingers together/apart. Drag: Move UI elements to a new location.	Swipe: Move a finger in a direction (left, right, up, down).
Typing and Input Select Text: Highlight text for editing or copying.	Type Text: Enter text into input fields.
System Operations Change Settings: Modify system settings (e.g., Wi-Fi, brightness). Navigate Menus: Access app sections or system menus.	Launch Application: Open a specific app.
Media Control Adjust Volume: Increase or decrease device volume.	Play/Pause: Control media playback.

操作类型	描述
触控交互 双击: 快速连续点击两次以触发操作。 长按: 持续按压以进行扩展交互，触发上下文选项或菜单。	轻触: 选择特定的界面元素。
基于手势的操作 捏合: 通过手指合拢或分开实现缩放。 拖动: 将界面元素移动到新位置。	滑动: 手指向某一方向移动 (左、右、上、下)。
输入与编辑 选择文本: 高亮文本以便编辑或复制。	输入文本: 在输入框中输入文字。
系统操作 更改设置: 修改系统设置 (如 Wi-Fi、亮度)。 导航菜单: 访问应用部分或系统菜单。	启动应用: 打开特定应用。
媒体控制 调节音量: 增加或减少设备音量。	播放/暂停: 控制媒体播放。

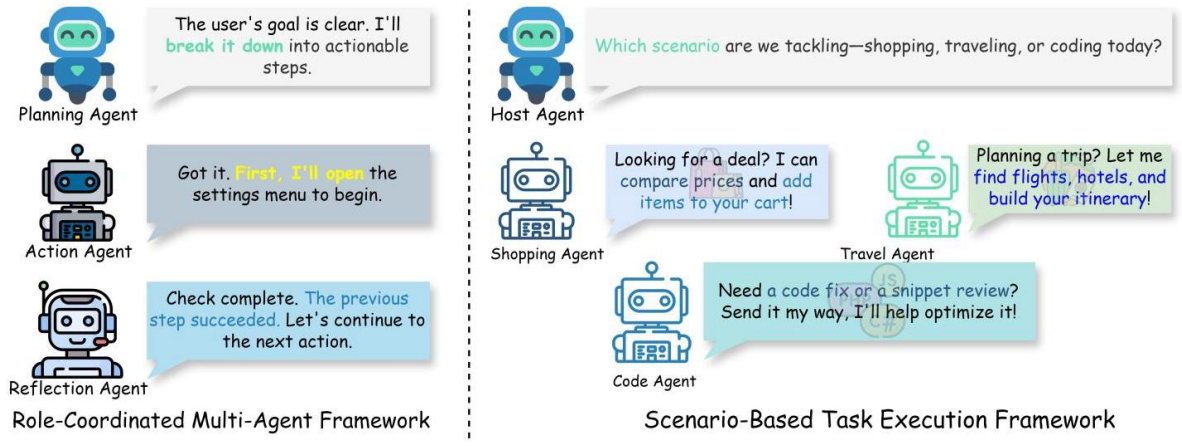

Fig. 6: Comparison of the role-coordinated and scenario-based multi-agent frameworks. The Role-Coordinated framework organizes agents based on general functional roles with a fixed workflow, while the Scenario-Based framework dynamically assigns tasks to specialized agents tailored for specific scenarios, allowing for increased flexibility and adaptability in handling diverse tasks.

图 6: 角色协调与基于场景的多智能体框架比较。角色协调框架基于通用功能角色组织智能体，采用固定工作流程；而基于场景的框架则动态分配任务给针对特定场景定制的专业智能体，从而在处理多样化任务时实现更高的灵活性和适应性。

3.4.1 Role-Coordinated Multi-Agent

3.4.1 角色协调多智能体

In the Role-Coordinated Multi-Agent Framework, agents are assigned general functional roles such as planning, decision-making, memory management, reflection, or tool invocation. These agents collaborate through a predefined workflow, with each agent focusing on its specific function to collectively achieve the overall task. This approach is particularly beneficial for tasks that require a combination of these general capabilities, allowing each agent to specialize and optimize its role within the workflow.

在角色协调多智能体框架中，智能体被分配通用功能角色，如规划、决策、记忆管理、反思或工具调用。这些智能体通过预定义的工作流程协作，每个智能体专注于其特定功能，共同完成整体任务。该方法特别适用于需要结合这些通用能力的任务，使每个智能体能够专精并优化其在工作流程中的角色。

For example, in MMAC-Copilot [72], multiple agents with distinct general functions collaborate as an OS copilot. The Planner strategically manages and allocates tasks to other agents, optimizing workflow efficiency. Meanwhile, the Librarian handles information retrieval and provides foundational knowledge, and the Programmer is responsible for coding and executing scripts, directly interacting with the software environment. The Viewer interprets complex visual information and translates it into actionable commands, while the Video Analyst processes and analyzes video content. Additionally, the Mentor offers strategic oversight and troubleshooting support. Each agent contributes its specialized function to the collaborative workflow, thereby enhancing the system's overall capability to handle complex interactions with the operating system.

例如，在 MMAC-Copilot [72] 中，多个具有不同通用功能的智能体协作为操作系统副驾驶。规划者 (Planner) 战略性地管理并分配任务以优化工作流程效率；图书管理员 (Librarian) 负责信息检索并提供基础知识；程序员 (Programmer) 负责编码和执行脚本，直接与软件环境交互；观察者 (Viewer) 解读复杂视觉信息并转化为可执行命令；视频分析师 (Video Analyst) 处理和分析视频内容；导师 (Mentor) 提供战略监督和故障排除支持。每个智能体贡献其专业功能，增强系统处理复杂操作系统交互的整体能力。

Similarly, in Mobile-Agent-v2 [52], three agents with general roles are utilized: a planning agent, a decision agent, and a reflection agent. The planning agent compresses historical actions and state information to provide a concise representation of task progress. The decision agent uses this information to navigate the task effectively, while the reflection agent monitors the outcomes of actions and corrects any errors, ensuring accurate task completion. This role-based collaboration reduces context length, improves task progression, and enhances focus content retention through a memory unit managed by the decision agent.

类似地，在 Mobile-Agent-v2 [52] 中，使用了三个具有通用角色的智能体: 规划智能体、决策智能体和反思智能体。规划智能体压缩历史动作和状态信息，提供任务进展的简明表示；决策智能体利用该信息有效导航任务；反思智能体监控动作结果并纠正错误，确保任务准确完成。基于角色的协作减少了上下文长度，提升了任务推进，并通过决策智能体管理的记忆单元增强了关注内容的保留。

In contrast, Mobile-Agent-E [74] decomposes tasks into high-level planning and low-level action execution, creating a system with a Manager Agent responsible for high-level planning and four subordinate agents: the Perceptor Agent, Operator Agent, Action Reflector Agent, and Notetaker Agent. The Perceptor Agent is responsible

for fine-grained visual perception. The Operator Agent determines the next specific actions based on task and perception information. The Action Reflector Agent checks the screenshots before and after operations to verify if the expected outcomes are achieved and provides feedback to the Manager and Operator Agents. The Notetaker Agent extracts task-related information for use in subsequent steps. Additionally, Mobile-Agent-E incorporates a Self-Evolution Module, using two specialized agents, AES and AET, to update long-term memory after each task completion. AES summarizes lessons learned, while AET records reusable operational sequences, helping the agent in efficiently completing common subtasks and making better decisions in similar future tasks.

相比之下，Mobile-Agent-E [74] 将任务分解为高层规划和低层动作执行，构建了一个由管理智能体负责高层规划，及四个下属智能体组成的系统: 感知智能体 (Perceptor Agent)、操作智能体 (Operator Agent)、动作反思智能体 (Action Reflector Agent) 和记录智能体 (Notetaker Agent)。感知智能体负责细粒度视觉感知；操作智能体基于任务和感知信息确定下一步具体动作；动作反思智能体通过检查操作前后的截图验证预期结果是否达成，并向管理和操作智能体反馈；记录智能体提取任务相关信息供后续步骤使用。此外，Mobile-Agent-E 引入了自我进化模块，利用两个专门智能体 AES 和 AET 在每次任务完成后更新长期记忆。AES 总结经验教训，AET 记录可复用的操作序列，帮助智能体高效完成常见子任务并在类似未来任务中做出更优决策。

CHOP [76] introduces a mobile operating assistant with Constrained High-frequency Optimized subtask Planning. This approach addresses challenges in the subtask level, which links high-level goals with low-level executable actions. CHOP overcomes VLM's deficiency in GUI scenario planning by using human-planned subtasks as basis vectors, significantly improving both effectiveness and efficiency across multiple applications in both English and Chinese contexts. The framework specifically targets two common issues: ineffective subtasks that lower-level agents cannot execute and inefficient subtasks that fail to contribute to higher-level task completion.

CHOP [76] 引入了一种具有限制性高频优化子任务规划的移动操作助手。该方法解决了连接高层目标与低层可执行动作的子任务层面挑战。CHOP 通过使用人工规划的子任务作为基向量，显著提升了视觉语言模型 (VLM) 在图形用户界面 (GUI) 场景规划中的效果和效率，适用于中英文多种应用。该框架针对两个常见问题: 低层智能体无法执行的无效子任务和未能促进高层任务完成的低效子任务。

In general computer automation, Cradle [73] leverages foundational agents with general roles to achieve versatile computer control. Agents specialize in functions like command generation or state monitoring, enabling Cradle to tackle general-purpose tasks across multiple software environment. Additionally, studies such as Ask-before-Plan [78], PromptRPA [75], LUMOS [260], and WebPilot [261] also utilize general-purpose role agents to execute tasks and excel in complex tasks like planning. Among these, LUMOS provides high-quality training data and methods for future intelligent agent research. Agent S2 [77] presents a compositional generalist-specialist framework for computer use agents that delegates cognitive responsibilities across various models. It introduces a Mixture-of-Grounding technique for precise GUI localization and Proactive Hierarchical Planning that dynamically refines action plans at multiple temporal scales based on evolving observations.

在通用计算机自动化领域，Cradle [73] 利用具有通用角色的基础智能体实现多功能计算机控制。智能体专注于命令生成或状态监控等功能，使 Cradle 能够处理多软件环境中的通用任务。此外，诸如 Ask-before-Plan [78]、PromptRPA [75]、LUMOS [260] 和 WebPilot [261] 等研究也采用通用角色智能体执行任务，并在规划等复杂任务中表现出色。其中，LUMOS 为未来智能体研究提供了高质量训练数据和方法。Agent S2 [77] 提出了一个组合型通用-专家框架，将认知职责分配给不同模型，采用基于定位的混合技术 (Mixture-of-Grounding) 实现精确 GUI 定位，并通过主动分层规划 (Proactive Hierarchical Planning) 根据动态观察在多个时间尺度上动态优化行动计划。

3.4.2 Scenario-Based Task Execution

3.4.2 基于场景的任务执行

In the Scenario-Based Task Execution Framework, tasks are dynamically assigned to specialized agents based on specific task scenarios or application domains. Each agent is endowed with capabilities tailored to a particular scenario, such as shopping, code editing, or navigation. By assigning tasks to agents specialized in the relevant domain, the system improves task success rates and efficiency.

在基于场景的任务执行框架中，任务根据具体任务场景或应用领域动态分配给专业智能体。每个智能体具备针对特定场景 (如购物、代码编辑或导航) 的定制能力。通过将任务分配给相关领域的专业智能体，系统提升了任务成功率和执行效率。

For instance, MobileExperts [55] forms different expert agents through an Expert Exploration phase. In the exploration phase, each agent receives tailored tasks broken down into sub-tasks to streamline the exploration process. Upon completion of a sub-task, the agent extracts three types of memories from its trajectory: interface memories, procedural memories (tools), and insight memories for use in subsequent execution phases. When a new task arrives, the system dynamically forms an expert team by selecting agents whose expertise matches the task requirements, enabling them to collaboratively execute the task more effectively. Similarly, in the SteP [79] framework, agents are specialized based on specific web scenarios such as shopping, GitLab, maps, Reddit, or CMS platforms. Each scenario agent possesses specific capabilities and knowledge relevant to its domain. When a task is received, it is dynamically assigned to the appropriate scenario agent, which executes the task leveraging its specialized expertise. This approach enhances flexibility and adaptability, allowing the system to handle a wide range of tasks across different domains more efficiently.

例如，MobileExperts [55] 通过专家探索阶段形成不同的专家代理。在探索阶段，每个代理接收分解为子任务的定制任务，以简化探索过程。完成子任务后，代理从其轨迹中提取三种类型的记忆: 界面记忆、程序记忆 (工具) 和洞察记忆，用于后续的执行阶段。当新任务到来时，系统通过选择与任务需求匹配的专家代理动态组建专家团队，使其能够更有效地协同执行任务。类似地，在 SteP [79] 框架中，代理根据特定的网络场景 (如购物、GitLab、地图、Reddit 或内容管理系统平台) 进行专业化。每个场景代理拥有与其领域相关的特定能力和知识。任务接收后，动态分配给相应的场景代理，利用其专业知识执行任务。这种方法增强了灵活性和适应性，使系统能够更高效地处理不同领域的多样化任务。

Through dynamic task assignment and specialization, the Scenario-Based Task Execution Framework optimizes multi-agent systems to adapt to diverse and evolving contexts, significantly enhancing both the efficiency

and effectiveness of task execution. As illustrated in Figure 6, the Role-Coordinated Framework relies on agents with general functional roles collaborating through a fixed workflow, suitable for tasks requiring a combination of general capabilities. In contrast, the Scenario-Based Framework dynamically assigns tasks to specialized agents tailored to specific scenarios, providing a flexible structure that adapts to the varying complexity and requirements of real-world tasks.

通过动态任务分配和专业化，基于场景的任务执行框架优化了多代理系统以适应多样且不断变化的环境，显著提升了任务执行的效率和效果。如图 6 所示，角色协调框架依赖具有通用功能角色的代理通过固定工作流程协作，适用于需要通用能力组合的任务。相比之下，基于场景的框架动态将任务分配给针对特定场景定制的专业代理，提供灵活的结构以适应现实任务的复杂性和多样需求。

Despite the potential of multi-agent frameworks in phone automation, several challenges remain. In the Role-Coordinated Framework, coordinating agents with general functions requires efficient workflow design and may introduce overhead in communication and synchronization. In the Scenario-Based Framework, maintaining and updating a diverse set of specialized agents can be resource-intensive, and dynamically assigning tasks requires effective task recognition and agent selection mechanisms. Future research could explore hybrid frameworks that combine the strengths of both approaches, leveraging general functional agents while also incorporating specialized scenario agents as needed. Additionally, developing advanced algorithms for agent collaboration, learning, and adaptation can further enhance the intelligence and robustness of multi-agent systems. Integrating external knowledge bases, real-time data sources, and user feedback can also improve agents' decision-making capabilities and adaptability in dynamic environment.

尽管多代理框架在手机自动化中具有潜力，但仍存在若干挑战。在角色协调框架中，协调具有通用功能的代理需要高效的工作流程设计，可能带来通信和同步的开销。在基于场景的框架中，维护和更新多样化的专业代理资源消耗较大，动态任务分配需要有效的任务识别和代理选择机制。未来研究可探索结合两者优势的混合框架，既利用通用功能代理，也根据需要引入专业场景代理。此外，开发先进的代理协作、学习和适应算法，可进一步提升多代理系统的智能性和鲁棒性。整合外部知识库、实时数据源和用户反馈，也能增强代理在动态环境中的决策能力和适应性。

3.5 Plan-Then-Act Framework

3.5 先规划后执行框架

While single-agent and multi-agent frameworks enhance adaptability and scalability, some tasks benefit from explicitly separating high-level planning from low-level execution. This leads to what we term the Plan-Then-Act Framework. In this paradigm, the agent first formulates a conceptual plan—often expressed as human-readable instructions—before grounding and executing these instructions on the device's UI.

虽然单代理和多代理框架提升了适应性和可扩展性，但某些任务受益于明确区分高层规划与低层执行。这催生了我们所称的先规划后执行框架。在该范式中，代理首先制定概念性计划——通常以人类可读的指令形式表达——然后在设备的用户界面上落实并执行这些指令。

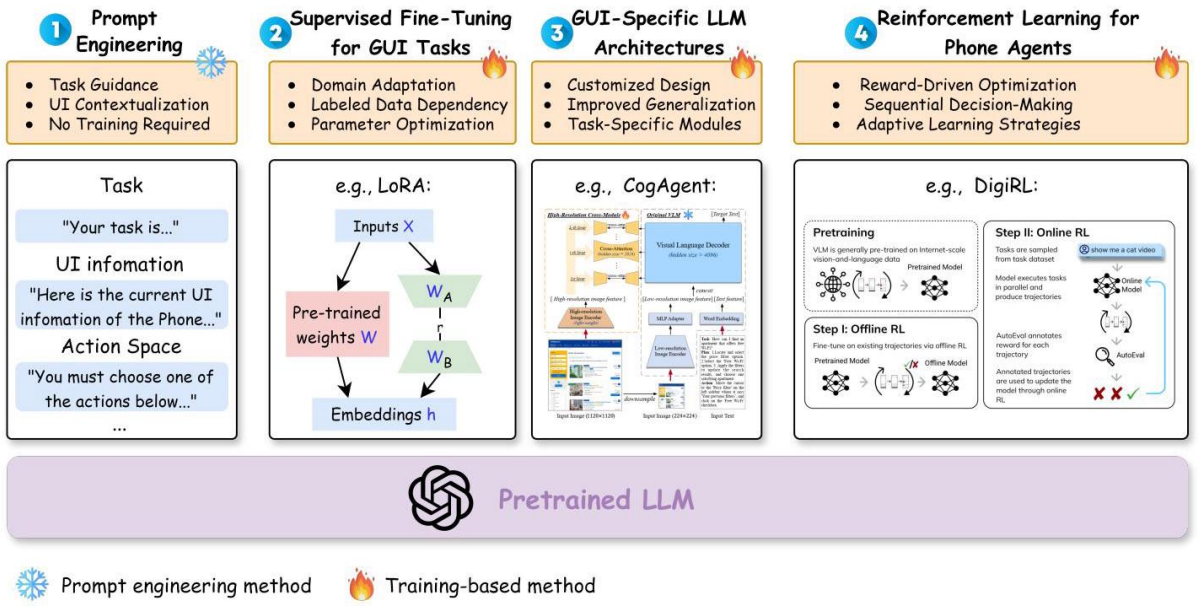

Fig. 7: Differences between training-based methods and prompt engineering in phone automation. Training-based methods adapt the model's parameters through additional training, enhancing its ability to perform specific tasks, whereas prompt engineering leverages the existing capabilities of pre-trained models by guiding them with well-designed prompts.

图 7: 基于训练的方法与提示工程在手机自动化中的区别。基于训练的方法通过额外训练调整模型参数，增强其执行特定任务的能力；而提示工程则利用预训练模型的现有能力，通过设计良好的提示引导模型。

The Plan-Then-Act approach addresses a fundamental challenge: although LLMs and multimodal LLMs (MLLMs) excel at interpreting instructions and reasoning about complex tasks, they frequently struggle to precisely map their textual plans to concrete UI actions. By decoupling these stages, the agent can focus on what should be done (planning) and then handle how to do it on the UI (acting). Recent works highlight the effectiveness of this approach:

先规划后执行方法解决了一个根本性挑战: 尽管大型语言模型 (LLMs) 和多模态大型语言模型 (MLLMs) 擅长理解指令和推理复杂任务，但它们常常难以将文本计划精确映射到具体的用户界面操作。通过将这两个阶段分离，代理可以专注于“做什么”(规划)，然后处理“如何在界面上做”(执行)。近期研究凸显了该方法的有效性:

- SeeAct [47] demonstrates that GPT-4V(ision) [31] can generate coherent plans for navigating websites. However, bridging the gap between textual plans and underlying UI elements remains challenging. By clearly delineating planning from execution, the system can better refine its plan before finalizing actions.

- SeeAct [47] 展示了 GPT-4V(ision) [31] 能生成连贯的网站导航计划。然而，文本计划与底层界面元素之间的桥接仍具挑战。通过明确区分规划与执行，系统能更好地完善计划后再执行操作。

- UGround [80] and related efforts [95], [101] emphasize advanced visual grounding. Under a Plan-Then-Act framework, the agent first crafts a task solution plan, then relies on robust visual grounding models to

locate and manipulate UI components. This modular design enhances performance across diverse GUIs and platforms, as the grounding model can evolve independently of the planning mechanism.

- UGround [80] 及相关工作 [95], [101] 强调了先进的视觉定位。在先规划后执行框架下，代理先制定任务解决方案计划，然后依赖强大的视觉定位模型定位和操作界面组件。这种模块化设计提升了在多样化图形用户界面和平台上的表现，因为定位模型可以独立于规划机制演进。

- LiMAC (Lightweight Multi-modal App Control) [81] also embodies a Plan-Then-Act spirit. LiMAC's Action Transformer (AcT) determines the required action type (the plan), and a specialized VLM is invoked only for natural language needs. By structuring decision-making and text generation into distinct stages, LiMAC improves responsiveness and reduces compute overhead, ensuring that reasoning and UI interaction are cleanly separated.

- LiMAC(轻量级多模态应用控制)[81] 也体现了先规划后执行的理念。LiMAC 的动作转换器 (AcT) 确定所需的动作类型 (即计划)，仅在自然语言需求时调用专门的视觉语言模型 (VLM)。通过将决策和文本生成分为不同阶段，LiMAC 提升了响应速度并降低计算开销，确保推理与界面交互清晰分离。

- ClickAgent [82] similarly employs a two-phase approach. The MLLM handles reasoning and action planning, while a separate UI location model pinpoints the relevant coordinates on the screen. Here, the MLLM's plan of which element to interact with is formed first, and only afterward is the element's exact location identified and the action executed.

- ClickAgent [82] 同样采用两阶段方法。多模态大型语言模型负责推理和动作规划，独立的界面定位模型确定屏幕上的相关坐标。这里，MLLM 先形成与哪个元素交互的计划，随后才确定元素的精确位置并执行操作。

- Ponder & Press [83] employs a general MLLM to decompose user instructions into executable actions. It then uses a GUI-specific MLLM to map the target elements in the action descriptions to pixel coordinates, thereby constructing a Plan-Then-Act Framework based solely on visual input. This framework is adaptable across various software environments without relying on supplementary information such as HTML or UI Trees.

- Ponder & Press [83] 使用通用多模态大语言模型 (MLLM) 将用户指令分解为可执行操作。随后，它利用针对图形用户界面 (GUI) 专门设计的 MLLM，将操作描述中的目标元素映射到像素坐标，从而构建了一个仅基于视觉输入的“先规划后执行”框架。该框架能够适应各种软件环境，无需依赖 HTML 或 UI 树等辅助信息。

The Plan-Then-Act Framework offers several advantages. Modularity allows improvements in planning without requiring changes to the UI grounding and execution modules, and vice versa. Error Mitigation enables the agent to revise its plan before committing to actions; if textual instructions are ambiguous or infeasible, they can be corrected, reducing wasted actions and improving reliability. Additionally, improved visual grounding models, OCR enhancements, and scenario-specific knowledge can further refine the Plan-Then-Act approach, making agents more adept at handling intricate, real-world tasks. In summary, the Plan-Then-Act Framework represents a natural evolution in designing MLLM-powered phone GUI agents. By separating planning from execution, agents can achieve clearer reasoning, improved grounding, and ultimately more effective and reliable task completion.

“先规划后执行”框架具有多项优势。模块化设计允许在不改变 UI 定位和执行模块的情况下改进规划，反之亦然。错误缓解机制使代理能够在执行操作前修正其计划；如果文本指令含糊或不可行，可以进行纠正，从而减少无效操作并提升可靠性。此外，改进的视觉定位模型、光学字符识别 (OCR) 增强以及特定场景知识的引入，均可进一步优化“先规划后执行”方法，使代理更擅长处理复杂的现实任务。总之，该框架代表了基于多模态大语言模型 (MLLM) 设计手机 GUI 代理的自然演进。通过将规划与执行分离，代理能够实现更清晰的推理、更精准的定位，最终完成更高效且可靠的任务。

4 LLMs for Phone Automation

4 手机自动化中的大语言模型 (LLMs)

LLMs [28], [29], [30], [31] have emerged as a transformative technology in phone automation, bridging natural language inputs with executable actions. By leveraging their advanced language understanding, reasoning, and generalization capabilities, LLMs enable agents to interpret complex user intents, dynamically interact with diverse mobile applications, and effectively manipulate GUIs.

大语言模型 (LLMs)[28], [29], [30], [31] 已成为手机自动化领域的变革性技术，连接自然语言输入与可执行操作。凭借其先进的语言理解、推理和泛化能力，LLMs 使代理能够解读复杂的用户意图，动态交互多样的移动应用，并有效操控图形用户界面 (GUI)。

In this section, we explore two primary approaches to leveraging LLMs for phone automation: Training-Based Methods and Prompt Engineering. Figure 7 illustrates the differences between these two approaches in the context of phone automation. Training-Based Methods involve adapting LLMs specifically for phone automation tasks through techniques like supervised fine-tuning [105], [107], [109], [110] and reinforcement learning [116], [117], [124]. These methods aim to enhance the models' capabilities by training them on GUI-specific data, enabling them to understand and interact with GUIs more effectively. Prompt Engineering, on the other hand, focuses on designing input prompts to guide pre-trained LLMs to perform desired tasks without additional training [238], [262], [263]. By carefully crafting prompts that include relevant information such as task descriptions, interface states, and action histories, users can influence the model's behavior to achieve specific automation goals [48], [49], [53].

本节探讨利用 LLMs 实现手机自动化的两种主要方法: 基于训练的方法和提示工程。图 7 展示了这两种方法在手机自动化中的差异。基于训练的方法通过监督微调 [105], [107], [109], [110] 和强化学习 [116], [117], [124] 等技术，专门调整 LLMs 以适应手机自动化任务。这些方法旨在通过训练 GUI 特定数据，提升模型理解和交互 GUI 的能力。提示工程则侧重于设计输入提示，引导预训练 LLMs 执行期望任务，无需额外训练 [238], [262], [263]。通过精心设计包含任务描述、界面状态和操作历史等相关信息的提示，用户可以影响模型行为，实现特定自动化目标 [48], [49], [53]。

4.1 Prompt Engineering

4.1 提示工程

LLMs like the GPT series [28], [29], [30] have demonstrated remarkable capabilities in understanding and generating human-like text. These models have revolutionized natural language processing by leveraging massive amounts of data to learn complex language patterns and representations.

如 GPT 系列 [28], [29], [30] 等 LLMs 在理解和生成类人文本方面展现出卓越能力。这些模型通过利用海量数据学习复杂语言模式和表示，革新了自然语言处理领域。

Prompt engineering is the practice of designing input prompts to effectively guide LLMs to produce desired outputs for specific tasks [238], [262], [263]. By carefully crafting the prompts, users can influence the model's behavior without the need for additional training or fine-tuning. This approach allows for leveraging the general capabilities of pre-trained models to perform a wide range of tasks by simply providing appropriate instructions or examples in the prompt.

提示工程是设计输入提示以有效引导 LLMs 为特定任务生成期望输出的实践 [238], [262], [263]。通过精心构造提示，用户无需额外训练或微调即可影响模型行为。此方法利用预训练模型的通用能力，仅通过提供适当的指令或示例，即可执行广泛任务。

In the context of phone automation, prompt engineering enables the utilization of general-purpose LLMs to perform automation tasks on mobile devices. Recently, a plethora of works have emerged that apply prompt engineering to achieve phone automation [48], [49], [51], [52], [53], [54], [55], [56], [60], [75], [84], [264]. These works leverage the strengths of LLMs in natural language understanding and reasoning to interpret user instructions and generate corresponding actions on mobile devices.

在手机自动化背景下，提示工程使得通用 LLMs 能够执行移动设备上的自动化任务。近年来，众多研究应用提示工程实现手机自动化 [48], [49], [51], [52], [53], [54], [55], [56], [60], [75], [84], [264]。这些工作利用 LLMs 在自然语言理解和推理方面的优势，解读用户指令并生成相应的移动设备操作。

The fundamental approach to achieving phone automation through prompt engineering entails the creation of prompts that encapsulate a comprehensive set of information. These prompts should include a detailed task description, such as searching for the best Korean restaurant on Yelp. They also integrate the current UI information of the phone, which may encompass screenshots, SoM, UI tree structures, icon details, and OCR data. Additionally, the prompts should account for the phone's real-time state, including its location, battery level, and keyboard status, as well as any pertinent action history and the range of possible actions (action space). The COT prompt [238], [265] is also a crucial component, guiding the thought process for the next operation. The LLM then analyzes this rich prompt and determines the subsequent action to execute. This methodical process is vividly depicted in Figure 8.

通过提示工程实现手机自动化的基本方法是创建包含全面信息的提示。这些提示应包括详细的任务描述，例如在 Yelp 上搜索最佳韩国餐厅；还应整合手机当前的 UI 信息，如截图、屏幕内容摘要 (SoM)、UI 树结构、图标详情及 OCR 数据。此外，提示需考虑手机的实时状态，包括位置、电量和键盘状态，以及相关操作历史和可能的操作空间。链式思维提示 (COT prompt)[238], [265] 也是关键组成部分，引导下一步操作的思考过程。LLM 据此丰富提示进行分析，确定后续执行的操作。该系统流程在图 8 中有生动展示。

This section explores the application of prompt engineering in phone automation, categorizing related works based on the type of prompts used: Text-Based Prompt and Multimodal Prompt. As illustrated in Figure 9, the

approach to automation significantly diverges between these two prompt types. Table 2 summarizes notable methods, highlighting their main UI information, the type of model used, and other relevant details such as task types and grounding strategies.

本节探讨提示工程在手机自动化中的应用，依据提示类型将相关工作分类为基于文本的提示和多模态提示。如图 9 所示，这两种提示类型在自动化方法上存在显著差异。表 2 总结了若干重要方法，突出其主要 UI 信息、所用模型类型及任务类型和定位策略等相关细节。

4.1.1 Text-Based Prompting

4.1.1 基于文本的提示

In the domain of text-based prompt automation, the primary architecture involves a single text-modal LLM serving as the agent for mobile device automation. This agent operates by interpreting UI information presented in the form of a UI tree. It is important to note that, to date, the approaches discussed have primarily utilized UI tree data and have not extensively incorporated OCR text and icon information. We believe that solely relying on OCR and icon information is insufficient for fully representing screen UI information; instead, as demonstrated in Mobile-agent-v2 [52], they are best used as auxiliary information alongside screenshots. These text-based prompt agents make decisions by selecting elements from a list of candidates based on the textual description of the UI elements. For instance, to initiate a search, the LLM would identify and select the search button by its index within the UI tree rather than its screen coordinates, as depicted in Figure 9.

在基于文本的提示自动化领域，主要架构涉及一个单一的文本模态大型语言模型 (LLM) 作为移动设备自动化的代理。该代理通过解释以 UI 树形式呈现的界面信息来操作。需要注意的是，迄今为止，所讨论的方法主要利用了 UI 树数据，尚未广泛整合 OCR 文本和图标信息。我们认为，仅依赖 OCR 和图标信息不足以全面表示屏幕 UI 信息；正如 Mobile-agent-v2 [52] 所示，它们更适合作为截图的辅助信息。这些基于文本的提示代理通过根据 UI 元素的文本描述，从候选列表中选择元素来做出决策。例如，为了启动搜索，LLM 会通过 UI 树中的索引而非屏幕坐标来识别并选择搜索按钮，如图 9 所示。

The study by Enabling Conversational [57] marked a significant step in this field. It explored the use of task descriptions, action spaces, and UI trees to map instructions to UI actions. However, it focused solely on the execution of individual instructions without delving into sequential decision-making processes. DroidBot-GPT [53] is a landmark in applying pre-trained language models to app automation. It is the first to explore the use of LLMs for app automation without requiring modifications to the app or the model. DroidBot-GPT perceives UI trees, which are structural representations of the app's UI, and integrates user-provided tasks along with action spaces and output requirements. This allows the model to engage in sequential decision-making and automate tasks effectively. AutoDroid [50] takes this concept further. It employs a UI Transition Graph (UTG) generated through random exploration to create an App Memory. This memory, combined with the commonsense knowledge of LLMs, enhances decision-making and significantly advances the capabilities of phone GUI agents. MobileGPT [61] introduces a hierarchical decision-making process. It simulates human cognitive processes-exploration, selection, derivation, and recall-to augment the efficiency and reliability of LLMs in mobile task automation. Lastly, AXNav [84] showcases an innovative application of Prompt Engineering in accessibility testing. AXNav interprets natural language instructions and executes them through an LLM, streamlining the testing process and improving

the detection of accessibility issues, thus aiding the manual testing workflows of QA professionals.

Enabling Conversational [57] 的研究在该领域迈出了重要一步。它探讨了使用任务描述、动作空间和 UI 树将指令映射到 UI 操作的方法，但仅关注单条指令的执行，未深入研究序列决策过程。DroidBot-GPT [53] 是将预训练语言模型应用于应用自动化的里程碑。它首次探索了无需修改应用或模型即可使用 LLM 进行应用自动化的方法。DroidBot-GPT 感知 UI 树，即应用 UI 的结构化表示，并整合用户提供的任务、动作空间及输出需求，使模型能够进行序列决策并有效自动化任务。AutoDroid [50] 在此基础上更进一步，利用通过随机探索生成的 UI 转换图 (UTG) 创建应用记忆。该记忆结合 LLM 的常识知识，提升了决策能力，显著增强了手机 GUI 代理的功能。MobileGPT [61] 引入了分层决策过程，模拟人类认知过程——探索、选择、推导和回忆，提升了 LLM 在移动任务自动化中的效率和可靠性。最后，AXNav [84] 展示了提示工程在无障碍测试中的创新应用。AXNav 通过 LLM 解释自然语言指令并执行，简化测试流程，提高无障碍问题的检测能力，助力 QA 专业人员的手动测试工作。

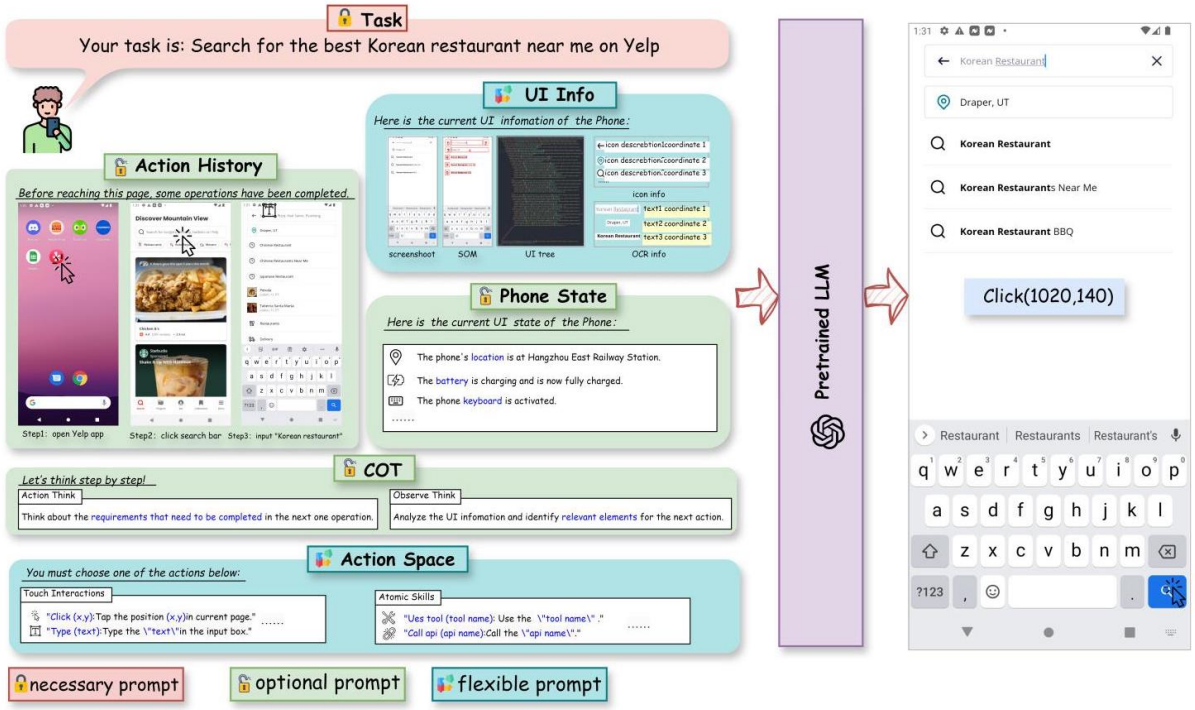

Fig. 8: Schematic of prompt engineering for phone automation. The necessary prompt is mandatory, initiating the task, e.g., searching for a Korean restaurant. The optional prompt are supplementary, enhancing tasks without being mandatory. The flexible prompt must include one or more elements from the UI Info, like a screenshot or OCR info, to adapt to task needs.

图 8: 手机自动化提示工程示意图。必要提示为强制性，启动任务，例如搜索韩国餐厅。可选提示为辅助性，增强任务但非必需。灵活提示必须包含来自 UI 信息的一个或多个元素，如截图或 OCR 信息，以适应任务需求。

Each of these contributions, while unique in their approach, is united by the common thread of Prompt Engineering. They demonstrate the versatility and potential of text-based prompt automation in enhancing the interaction between LLMs and mobile applications.

这些贡献各具特色，但均以提示工程为共同纽带，展示了基于文本的提示自动化在增强 LLM 与移动应用交互中的多样性和潜力。

4.1.2 Multimodal Prompting

4.1.2 多模态提示

With the advancement of large pre-trained models, Multimodal Large Language Models (MLLMs) have demonstrated exceptional performance across various domains [31], [229], [266], [267], [268], [269], [270], [271], [272], [273], [274], significantly contributing to the evolution of phone automation. Unlike text-only models, multimodal models integrate visual and textual information, addressing limitations such as the inability to access UI trees, missing control information, and inadequate global screen representation. By leveraging screenshots for decision-making, multimodal models facilitate a more natural simulation of human interactions with mobile devices, enhancing both accuracy and robustness in automated operations.

随着大型预训练模型的发展, 多模态大型语言模型 (MLLMs) 在多个领域展现出卓越性能 [31], [229], [266]，[267]，[268]，[269]，[270]，[271]，[272]，[273]，[274]，显著推动了手机自动化的演进。与纯文本模型不同，多模态模型融合视觉与文本信息，解决了无法访问 UI 树、缺失控件信息及屏幕全局表示不足等问题。通过利用截图进行决策，多模态模型更自然地模拟人类与移动设备的交互，提升了自动化操作的准确性和鲁棒性。

The fundamental framework for multimodal phone automation is illustrated in Figure 9. Multimodal prompts integrate visual perception (e.g., screenshots) and textual information (e.g., UI tree, OCR, and icon data) to guide MLLMs in generating actions. The action outputs can be categorized into two methods: SoM-Based Indexing Methods and Direct Coordinate Output Methods. These methods define how the agent identifies and interacts with UI elements, either by referencing annotated indices or by pinpointing precise coordinates. SoM-Based Indexing Methods. SoM-based methods involve annotating UI elements with unique identifiers within the screenshot, allowing the MLLM to reference these elements by their indices when generating actions. This approach mitigates the challenges associated with direct coordinate outputs, such as precision and adaptability to dynamic interfaces. MM-Navigator [60] represents a breakthrough in zero-shot GUI navigation using GPT-4V [31]. By employing SoM prompting [228], MM-Navigator annotates screenshots through OCR and icon recognition, assigning unique numeric IDs to actionable widgets. This enables GPT-4V to generate indexed action descriptions rather than precise coordinates, enhancing action execution accuracy. Building upon the SoM-based approach, AppAgent [48] integrates autonomous exploration and human demonstration observation to construct a comprehensive knowledge base. This framework allows the agent to navigate and operate smartphone applications through simplified action spaces, such as tapping and swiping, without requiring backend system access. Tested across 10 different applications and 50 tasks, AppAgent showcases superior adaptability and efficiency in handling diverse high-level tasks, further advancing multimodal phone automation. OmniParser [56] enhances the SoM-based method by introducing a robust screen parsing technique. It combines fine-tuned interactive icon detection models and functional captioning models to convert UI screenshots into structured elements with bounding boxes and labels. This comprehensive parsing significantly improves GPT-4V's ability to generate accurately grounded actions, ensuring reliable operation across multiple platforms and applications. GUI Narrator [62] utilizes video captioning to guide the VLM, aiding in the deeper understanding of GUI operations. The framework uses the mouse cursor as a visual prompt, highlighting it with a green bounding box to enhance the VLM's interpretative abilities with high-resolution screen-

shots. By extracting screenshots from before and after GUI actions occur in the video as keyframes, it provides temporal and spatial logic to the action screenshots. These are combined into prompts to further guide the VLM in producing accurate action descriptions, thereby improving its performance.

多模态手机自动化的基本框架如图 9 所示。多模态提示融合了视觉感知 (如截图) 和文本信息 (如 UI 树、OCR 及图标数据)，以引导多模态大语言模型 (MLLMs) 生成操作。操作输出可分为两类方法: 基于 SoM 的索引方法和直接坐标输出方法。这些方法定义了代理如何识别和交互 UI 元素，或通过引用带注释的索引，或通过精确定位坐标。基于 SoM 的索引方法涉及在截图中为 UI 元素标注唯一标识符，使 MLLM 在生成操作时可通过索引引用这些元素。此方法缓解了直接坐标输出所面临的精度和动态界面适应性挑战。MM-Navigator [60] 代表了使用 GPT-4V [31] 实现零样本 GUI 导航的突破。通过采用 SoM 提示技术 [228]，MM-Navigator 通过 OCR 和图标识别对截图进行注释，为可操作控件分配唯一数字 ID，使 GPT-4V 生成索引化的操作描述而非精确坐标，从而提升操作执行的准确性。在基于 SoM 方法的基础上，AppAgent [48] 整合了自主探索和人类示范观察，构建了全面的知识库。该框架允许代理通过简化的操作空间 (如点击和滑动) 导航和操作智能手机应用，无需后台系统访问。AppAgent 在 10 个不同应用和 50 个任务中测试，展示了其在处理多样化高级任务时的卓越适应性和效率，进一步推动了多模态手机自动化的发展。OmniParser [56] 通过引入强大的屏幕解析技术增强了基于 SoM 的方法。它结合了微调的交互式图标检测模型和功能性描述模型，将 UI 截图转换为带有边界框和标签的结构化元素。这种全面的解析显著提升了 GPT-4V 生成准确定位操作的能力，确保了跨多平台和应用的可靠运行。GUI Narrator [62] 利用视频字幕引导视觉语言模型 (VLM)，帮助其更深入理解 GUI 操作。该框架使用鼠标光标作为视觉提示，以绿色边框突出显示，增强 VLM 对高分辨率截图的解读能力。通过提取视频中 GUI 操作前后的截图作为关键帧，提供了操作截图的时空逻辑。这些内容被整合为提示，进一步引导 VLM 生成准确的操作描述，从而提升其性能。

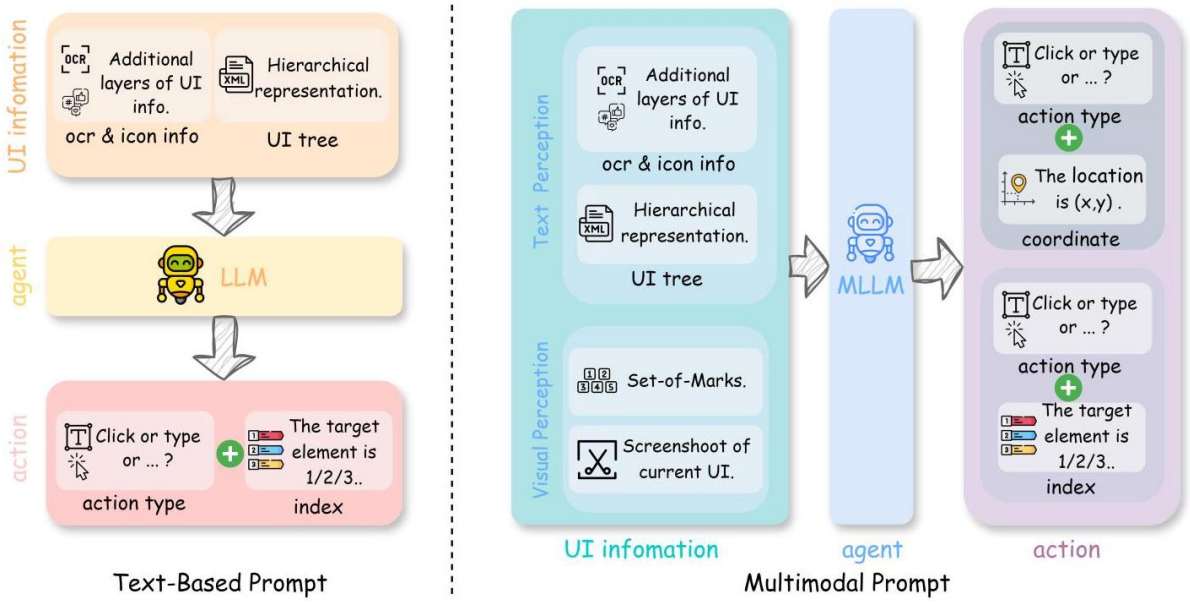

Fig. 9: Comparison between text-based prompt and multimodal prompt. In Text-Based Prompt, the LLM processes textual UI information, such as UI tree structures and OCR data, to determine the action type (index). In contrast, Multimodal Prompt integrates screenshot data with supplementary UI information to facilitate decision-making by the agent. The MLLM can then pinpoint the action location using either coordinates or indices.

图 9: 基于文本提示与多模态提示的比较。在基于文本的提示中，大语言模型处理文本 UI 信息，如 UI 树结构和 OCR 数据，以确定操作类型 (索引)。相比之下，多模态提示整合了截图数据及补充的 UI 信息，辅助代理决策。多模态大语言模型随后可通过坐标或索引定位操作位置。

Direct Coordinate Output Methods. Direct coordinate output methods enable MLLMs to determine the exact (x , y) positions of UI elements from screenshots, facilitating precise interactions without relying on indexed references. This approach leverages the advanced visual grounding capabilities of MLLMs to interpret and interact with the UI elements directly. VisionTasker [85] introduces a two-stage framework that combines vision-based UI understanding with LLM task planning. Utilizing models like YOLOv8 [275] and PaddleOCR [276], VisionTasker parses screenshots to identify widgets and textual information, transforming them into natural language descriptions. This structured semantic representation allows the LLM to perform step-by-step task planning, enhancing the accuracy and practicality of automated mobile task execution. The Mobile-Agent series [51], [52] leverages visual perception tools to accurately identify and locate both visual and textual UI elements within app screenshots. Mobile-Agent-v1 utilizes coordinate-based actions, enabling precise interaction with UI elements. Mobile-Agent-v2 extends this by introducing a multi-agent architecture comprising planning, decision, and reflection agents. Mobile-Agent-E [74] optimizes the multi-agent architecture by detailing the responsibilities of each agent. It also introduces a long-term memory mechanism through the design of a Self-Evolution Module, which accumulates experience and enables agents to evolve, thereby enhancing adaptability to new tasks. MobileExperts [55] advances the direct coordinate output method by incorporating tool formulation and multi-agent collaboration. This dynamic, tool-enabled agent team employs a dual-layer planning mechanism to efficiently execute multi-step operations while reducing reasoning costs by approximately 22%. By dynamically assembling specialized agents and utilizing reusable code block tools, MobileExperts demonstrates enhanced intelligence and operational efficiency in complex phone automation tasks. Unlike AppAgent, AppAgent v2 [64] integrates parsers with visual features and employs UI element coordinates along with Index information, creating a more flexible action space. This allows the agent to manage dynamic interfaces and non-standard UI elements more adeptly, thereby enhancing its adaptability to various complex tasks. VisionDroid [54] applies MLLMs to automated GUI testing, focusing on detecting non-crash functional bugs through vision-based UI understanding. By aligning textual and visual information, VisionDroid enables the MLLM to comprehend GUI semantics and operational logic, employing step-by-step task planning to enhance bug detection accuracy. Evaluations across multiple datasets and real-world applications highlight VisionDroid's superior performance in identifying and addressing functional bugs.

直接坐标输出方法。直接坐标输出方法使多模态大语言模型 (MLLMs) 能够从截图中确定 UI 元素的精确 (x，y) 位置，促进精确交互而无需依赖索引引用。该方法利用 MLLMs 先进的视觉定位能力，直接解释和操作 UI 元素。VisionTasker [85] 提出了一个结合基于视觉的 UI 理解与大语言模型任务规划的两阶段框架。通过使用 YOLOv8 [275] 和 PaddleOCR [276] 等模型，VisionTasker 解析截图以识别控件和文本信息，并将其转化为自然语言描述。这种结构化的语义表示使大语言模型能够进行逐步任务规划，提升自动化移动任务执行的准确性和实用性。Mobile-Agent 系列 [51]，[52] 利用视觉感知工具准确识别和定位应用截图中的视觉及文本 UI 元素。Mobile-Agent-v1 采用基于坐标的操作，实现对 UI 元素的精确交互。Mobile-Agent-v2 通过引入包含规划、决策和反思代理的多代理架构扩展了该方法。Mobile-Agent-E [74] 通过细化各代理职责优化了多代理架构，并通过设计自我进化模块引入长期记忆机制，积累经验并使代理进化，从而增强对新任务的适应能力。MobileExperts [55] 通过整合工具制定和多代理协作推进了直接坐标输出方法。该动态工具驱动的代理团队采用双层规划机制，高效执行多步骤操作，同时将推理成本降低约 22%。通过动态组建专业代理和利用可复用代码块工具，MobileExperts 在复杂手机自动化任务中展现出更高的智能和操作效率。与 AppAgent 不同，AppAgent v2 [64] 结合了解析器与视觉特征，利用 UI 元素坐标及索引信息，构建了更灵活的动作空间，使代理更善于处理动态界面和非标准 UI 元素，从而提升其对各种复杂任务的适应性。VisionDroid [54] 将 MLLM 应用于自动化 GUI 测试，重点通过基于视觉的 UI 理解检测非崩溃功能性缺陷。通过对齐文本与视觉信息，VisionDroid 使 MLLM 理解 GUI 语义和操作逻辑，采用逐步任务规划提升缺陷检测准确率。多数据集和实际应用评估显示 VisionDroid 在识别和解决功能性缺陷方面表现优异。

While multimodal prompt strategies have significantly advanced phone automation by integrating visual and textual data, they still face notable challenges. Approaches that do not utilize SoM maps and instead directly output coordinates rely heavily on the MLLM's ability to accurately ground UI elements for precise manipulation. Although recent innovations [52], [54], [55] have made progress in addressing the limitations of MLLMs' grounding capabilities, there remains considerable room for improvement. Enhancing the robustness and accuracy of UI grounding is essential to achieve more reliable and scalable phone automation.

尽管多模态提示策略通过整合视觉和文本数据显著推动了手机自动化的发展，但仍面临显著挑战。不利用 SoM 地图而直接输出坐标的方法高度依赖 MLLM 准确定位 UI 元素以实现精确操作。尽管近期创新 [52]，[54]，[55] 在解决 MLLM 定位能力局限方面取得进展，但仍有较大提升空间。提升 UI 定位的鲁棒性和准确性对于实现更可靠、更具扩展性的手机自动化至关重要。

TABLE 2: Summary of prompt engineering methods for phone GUI agents

表 2: 手机 GUI 代理的提示工程方法总结

Method	Date	Task Type	Model	Screenshot	**SoM**	UI tree	Icon & OCR	Grounding
DroidBot-GPT [53] :	2023.04	General	ChatGPT	✘	✘	✓	✘	Index
Enabling conversational [57]	2023.04	Screen Understanding, QA	PaLM	✘	✘	✓	✘	Index
AutoDroid [50] !	2023.09	General	GPT-4, GPT-3.5	✘	✘	✓	✘	Index
MM-Navigator [60] !	2023.11	General	GPT-4V	✓	✘	✘	✓	Index
VisionTasker [85] !	2023.12	Manual Teaching	GPT-4	✘	✓	✓	✓	Index
AppAgent [48] !	2023.12	General	GPT-4	✓	✓	✓	✓	Index
MobileGPT [61] ⎚	2023.12	General	GPT-3.5, GPT-4	✘	✘	✓	✘	Index
Mobile-Agent [51] ?	2024.01	General	GPT-4V	✓	✘	✘	✓	Coordinate
AXNav [84]	2024.05	Bug Testing	GPT-4	✘	✘	✓	✓	Index
Mobile-Agent-v2 [52] ?	2024.06	General	GPT-4V	✓	✘	✘	✓	Coordinate
GUI Narrator [62] ?	2024.06	GUI Video Captioning	GPT-40, QwenVL-7B	✓	✓	✘	✘	Index
MobileExpert [55]	2024.07	General	GPT-4V	✓	✘	✘	✘	Coordinate
VisionDroid [54] ?	2024.07	Non-Crash Functional Bug Detection	GPT-4	✓	✓	✓	✘	Index
AppAgent v2 [64]	2024.08	General	GPT-4	✓	✓	✓	✓	Coordinate, Index
OmniParser [56]	2024.08	General	GPT-4V	✓	✓	✘	✓	Index
Mobile-Agent-E [74] □	2025.01	General	GPT-40, Claude -3.5, Gemini-1.5	✓	✘	✘	✓	Coordinate
Mobile-Agent-V [68] ?	2025.02	General	GPT-4o	✓	✘	✘	✓	Coordinate
LearnAct [155] ?	2025.02	General	Gimini-1.5	✓	✘	✘	✘	Coordinate

方法	日期	任务类型	模型	截图	**SoM**	界面树	图标与 OCR	定位
DroidBot-GPT [53] :	2023.04	通用	ChatGPT	✘	✘	✓	✘	索引
启用对话式 [57]	2023.04	屏幕理解，问答	PaLM	✘	✘	✓	✘	索引
AutoDroid [50] !	2023.09	通用	GPT-4，GPT-3.5	✘	✘	✓	✘	索引
MM-Navigator [60] !	2023.11	通用	GPT-4V	✓	✘	✘	✓	索引
VisionTasker [85] !	2023.12	手动教学	GPT-4	✘	✓	✓	✓	索引
AppAgent [48] !	2023.12	通用	GPT-4	✓	✓	✓	✓	索引
MobileGPT [61] ⎚	2023.12	通用	GPT-3.5，GPT-4	✘	✘	✓	✘	索引
Mobile-Agent [51] ?	2024.01	通用	GPT-4V	✓	✘	✘	✓	坐标
AXNav [84]	2024.05	缺陷测试	GPT-4	✘	✘	✓	✓	索引
Mobile-Agent-v2 [52] ?	2024.06	通用	GPT-4V	✓	✘	✘	✓	坐标
GUI 讲述者 [62] ?	2024.06	GUI 视频字幕	GPT-40，QwenVL-7B	✓	✓	✘	✘	索引
MobileExpert [55]	2024.07	通用	GPT-4V	✓	✘	✘	✘	坐标
VisionDroid [54] ?	2024.07	非崩溃功能性缺陷检测	GPT-4	✓	✓	✓	✘	索引
AppAgent v2 [64]	2024.08	通用	GPT-4	✓	✓	✓	✓	坐标，索引
OmniParser [56]	2024.08	通用	GPT-4V	✓	✓	✘	✓	索引
Mobile-Agent-E [74] □	2025.01	通用	GPT-40，Claude -3.5，Gemini-1.5	✓	✘	✘	✓	坐标
Mobile-Agent-V [68] ?	2025.02	通用	GPT-4o	✓	✘	✘	✓	坐标
LearnAct [155] ?	2025.02	通用	Gimini-1.5	✓	✘	✘	✘	坐标

4.2 Training-Based Models

4.2 基于训练的模型

The subsequent sections delve into these approaches, discussing the development of task-specific model architectures, supervised fine-tuning strategies and reinforcement learning techniques in both general-purpose and Phone UI-specific scenarios.

以下章节深入探讨这些方法，讨论任务特定模型架构的开发、监督微调策略以及在通用和手机界面特定场景中的强化学习技术。

4.2.1 Task-Specific LLM-based Agents

4.2.1 基于任务的 LLM 代理

To advance AI agents for phone automation, significant efforts have been made to develop Task Specific Model Architectures that are tailored to understand and interact with GUIs by integrating visual perception with language

understanding. These models address unique challenges posed by GUI environment, such as varying screen sizes, complex layouts, and diverse interaction patterns. A summary of notable Task Specific Model Architectures is presented in Figure 3, highlighting their main contributions, domains, and other relevant details. General-Purpose Models. The general-purpose GUI-specific LLMs are designed to handle a wide range of tasks across different applications and interfaces. They focus on enhancing direct GUI interaction, high-resolution visual recognition, and comprehensive perception to improve the capabilities of AI agents in understanding and navigating complex mobile GUIs. One significant challenge in this domain is enabling agents to interact directly with GUIs without relying on environment parsing or application-specific APIs, which can introduce inefficiencies and error propagation. To tackle this, Auto-GUI [66] presents a multimodal agent that directly engages with the interface. It introduces a chain-of-action technique that leverages previous action histories and future action plans, enhancing the agent's decision-making process and leading to improved performance in GUI control tasks. High-resolution input is essential for recognizing tiny UI elements and text prevalent in GUIs. CogAgent [46] addresses this by employing both low-resolution and high-resolution image encoders within its architecture. Supporting input resolutions up to 1120×1120 , CogAgent effectively recognizes small page elements and text. Understanding UIs and infographics requires models to interpret complex visual languages and design principles. ScreenAI [67] improves upon existing architectures by introducing a flexible patching strategy and a novel textual representation for UIs. During pre-training, this representation teaches the model to interpret UI elements effectively. Leveraging large language models, ScreenAI automatically generates training data at scale, covering a wide spectrum of tasks in UI and infographic understanding. Enhancing both perception and action response is crucial for comprehensive GUI automation. CoCo-Agent [71] proposes two novel approaches: comprehensive environment perception (CEP) and conditional action prediction (CAP). CEP enhances GUI perception through multiple aspects, including visual channels (screenshots and detailed layouts) and textual channels (historical actions). CAP decomposes action prediction into determining the action type first, then identifying the action target conditioned on the action type. Addressing the need for effective GUI agents in applications featuring extensive Mandarin content, MobileFlow [88] introduces a multimodal LLM specifically designed for mobile GUI agents. MobileFlow employs a hybrid visual encoder trained on a vast array of GUI pages, enabling it to extract and comprehend information across diverse interfaces. The model incorporates a Mixture of Experts (MoE) and specialized modality alignment training tailored for GUIs. ShowUI [89] employs the UI-Guided visual tokens selection method, which randomly selects a subset of tokens from each component during training. This approach retains the original positional information while reducing redundant tokens by 33%, thereby accelerating training speed by 1.4 times. Furthermore, by using interleaved vision-language-action streaming combined with high-quality training data, it significantly improves the training speed and performance of GUI visual agents. Aguvis [90] employs a two-stage training method to enhance the generalization and efficiency of GUI agents. It uses single-step task data to train the model's grounding abilities and multi-step task data to develop the model's planning and reasoning capabilities. This approach significantly improves the overall performance of the agents. UI-TARS [92] employs a more in-depth and structurally robust System-2 reasoning method, combined with online bootstrapping and reflection tuning strategies. This combination effectively assists the model in handling complex tasks in dynamic environments and continuously optimizes overall performance. V-Droid [114] introduces a novel verifier-driven architecture where the LLM does not generate actions directly but instead scores and selects from a finite set of extracted actions, improving task success rates and significantly reducing latency. Collectively, these general-purpose Task Specific Model Architectures address key challenges in phone automation by enhancing direct GUI interaction, high-resolution visual recognition, comprehensive environment perception, and conditional action prediction. By leveraging multimodal inputs and innovative architectural designs, these models significantly advance the capabilities of AI agents in understanding and navigating complex mobile GUIs, paving the way for more intelligent and autonomous phone automation solutions. Phone UI-Specific Models. Phone UI-Specific Model Architectures have primarily focused

on screen understanding

为了推动手机自动化的 AI 代理发展，已投入大量精力开发针对图形用户界面 (GUI) 理解与交互的任务特定模型架构，这些架构通过整合视觉感知与语言理解来实现。此类模型解决了 GUI 环境中诸如屏幕尺寸多样、布局复杂及交互模式多样等独特挑战。图 3 总结了若干重要的任务特定模型架构，突出其主要贡献、应用领域及其他相关细节。通用模型。通用的 GUI 特定大型语言模型 (LLM) 旨在处理不同应用和界面中的广泛任务，重点提升直接 GUI 交互、高分辨率视觉识别及全面感知能力，以增强 AI 代理对复杂移动 GUI 的理解和导航能力。该领域的一大挑战是使代理能够直接与 GUI 交互，而不依赖环境解析或应用特定 API，这些方法可能导致效率低下和错误传播。为此，Auto-GUI [66] 提出了一种多模态代理，能够直接与界面交互。它引入了动作链 (chain-of-action) 技术，利用历史动作和未来动作计划，增强代理的决策过程，从而提升 GUI 控制任务的表现。高分辨率输入对于识别 GUI 中常见的微小 UI 元素和文本至关重要。CogAgent [46] 通过在架构中同时采用低分辨率和高分辨率图像编码器来解决这一问题。支持最高 1120×1120 分辨率输入，CogAgent 有效识别小型页面元素和文本。理解 UI 和信息图需要模型解读复杂的视觉语言和设计原则。ScreenAI [67] 通过引入灵活的分块策略和新颖的 UI 文本表示，改进了现有架构。在预训练阶段，该表示教会模型有效解读 UI 元素。借助大型语言模型，ScreenAI 自动生成大规模训练数据，涵盖 UI 和信息图理解的广泛任务。提升感知与动作响应对于全面的 GUI 自动化至关重要。CoCo-Agent [71] 提出了两种新方法: 全面环境感知 (CEP) 和条件动作预测 (CAP)。CEP 通过多方面增强 GUI 感知，包括视觉通道 (截图和详细布局) 及文本通道 (历史动作)。CAP 将动作预测分解为先确定动作类型，再基于动作类型识别动作目标。针对包含大量中文内容的应用中对高效 GUI 代理的需求，MobileFlow [88] 引入了一种专为移动 GUI 代理设计的多模态 LLM。MobileFlow 采用在大量 GUI 页面上训练的混合视觉编码器，能够提取并理解多样界面信息。该模型结合了专家混合 (MoE) 和专门针对 GUI 的模态对齐训练。ShowUI [89] 采用 UI 引导的视觉标记选择方法，在训练时随机选择每个组件的一部分标记。此方法保留了原始位置信息，同时减少了 33% 的冗余标记，从而加快了 1.4 倍的训练速度。此外，通过交错的视觉-语言-动作流和高质量训练数据，显著提升了 GUI 视觉代理的训练速度和性能。Aguvis [90] 采用两阶段训练方法提升 GUI 代理的泛化能力和效率。其利用单步任务数据训练模型的定位能力，利用多步任务数据培养模型的规划与推理能力，显著提升代理整体表现。UI-TARS [92] 采用更深入且结构稳健的系统 2 推理方法，结合在线自举和反思调优策略，有效辅助模型处理动态环境中的复杂任务，并持续优化整体性能。V-Droid [114] 引入了一种新颖的验证者驱动架构，LLM 不直接生成动作，而是对有限的提取动作集合进行评分和选择，提升任务成功率并显著降低延迟。总体来看，这些通用任务特定模型架构通过增强直接 GUI 交互、高分辨率视觉识别、全面环境感知和条件动作预测，解决了手机自动化中的关键挑战。借助多模态输入和创新架构设计，这些模型显著提升了 AI 代理理解和导航复杂移动 GUI 的能力，为更智能自主的手机自动化解决方案奠定基础。手机界面特定模型。手机界面特定模型架构主要聚焦于屏幕理解

TABLE 3: Summary of task-specific model architectures

表 3: 任务特定模型架构汇总

Method	Date	Task Type	Backbone	Size	Contributions
Auto-GUI [66] ?	2023.09	General	N/A	60M / 200M / 700M	Direct screen interaction; Chain-of-action; Action histories and future plans
CogAgent [46] ?	2023.12	General	CogVLM	18B	High-res input (1120 × 1120); Specialized in GUI understanding
WebVLN-Net [104] ?	2023.12	Screen Understanding, QA	N/A	N/A	Web navigation with visual and HTML content
ScreenAI [67] ?	2024.02	Screen Understanding, QA	N/A	4.6B	UI and infographic understanding; Flexible patching
CoCo-Agent [71] ?	2024.02	General	LLaVA (LLaMA-2- chat-7B, CLIP)	N/A	Comprehensive perception; Conditional action prediction; Enhanced automation
Ferret-UI [101] ?	2024.04	Screen Understanding, Referring	Ferret	N/A	”Any resolution” tech-niques; Precise referring and grounding
LVG [94]	2024.06	Screen Understanding, Grounding	SWIN Transformer, BERT	N/A	Visual UI grounding; Layout-guided contrastive learning
Textual Foresight [103]	2024.06	Screen Understanding, Referring	BLIP-2	N/A	Predict UI state; UI representation learning
MobileFlow [88]	2024.07	General	Qwen-VL-Chat	21B	Hybrid visual encoders; Variable resolutions; Multilingual support
UI-Hawk [95]	2024.08	Screen Understanding, Grounding	N/A	N/A	History-aware encoder; Screen stream processing; FunUI benchmark
Ferret-UI 2 [102]	2024.10	Screen Understanding, Referring	Ferret	N/A	Multi-platform; High-resolution encoding
OS-Atlas [97] :	2024.10	Screen Understanding, Grounding	Owen2-VL, InternVL-2	4B / 7B	Grounding data synthesis; Largest GUI grounding corpus
ShowUI [89] :	2024.11	General	Qwen2-VL	2B	Visual tokens selection; Cross-modal understanding
Aguvis [90] :	2024.12	General	Qwen2-VL	7B / 72B	Comprehensive data pipeline; Two-stage training;Cross-platform
Aria-UI [96] :	2024.12	Screen Understanding, Grounding	Aria	3.9B	Diversified dataset pipeline; Multimodal dynamic action history
UI-TARS [92] ⊠	2025.01	General	Qwen2-VL	2B / 7B / 72B	System-2 Reasoning; Online bootstrapping; Reflection tuning
GUI-Bee [100] □	2025.01	Screen Understanding, Grounding	SeeClick, UIX-7B, Qwen-GUI	N/A	Model-Environment alignment; Self-exploratory Data
V-Droid [114]	2025.03	General	Llama-3.1-8B	8b	Verifier-driven framework
MP-GUI [98] □	2025.03	General	InternVL2-8B	8B	Screen Understanding, Referring

方法	日期	任务类型	骨干网络	规模	贡献
Auto-GUI [66] ?	2023.09	通用	不适用	6000 万 / 2 亿 / 7 亿	直接屏幕交互；动作链；动作历史与未来计划
CogAgent [46] ?	2023.12	通用	CogVLM	18B	高分辨率输入 (1120 × 1120)；专注于 GUI 理解
WebVLN-Net [104] ?	2023.12	屏幕理解，问答	不适用	不适用	结合视觉和 HTML 内容的网页导航
ScreenAI [67] ?	2024.02	屏幕理解，问答	不适用	4.6B	用户界面和信息图理解；灵活的补丁处理
CoCo-Agent [71] ?	2024.02	通用	LLaVA(LLaMA-2-chat-7B，CLIP)	不适用	全面感知；条件动作预测；增强自动化
Ferret-UI [101] ?	2024.04	屏幕理解，指代	Ferret	不适用	“任意分辨率”技术；精准指代与定位
LVG [94]	2024.06	屏幕理解，定位	SWIN Transformer，BERT	不适用	视觉用户界面定位；布局引导的对比学习
Textual Foresight [103]	2024.06	屏幕理解，指代	BLIP-2	不适用	预测用户界面状态；用户界面表示学习
MobileFlow [88]	2024.07	通用	Qwen-VL-Chat	21B	混合视觉编码器；可变分辨率；多语言支持
UI-Hawk [95]	2024.08	屏幕理解，定位	不适用	不适用	历史感知编码器；屏幕流处理；FunUI 基准
Ferret-UI 2 [102]	2024.10	屏幕理解，指代	Ferret	不适用	多平台；高分辨率编码
OS-Atlas [97] :	2024.10	屏幕理解，定位	Owen2-VL，InternVL-2	40 亿 / 70 亿	定位数据合成；最大规模 GUI 定位语料库
ShowUI [89] :	2024.11	通用	Qwen2-VL	2B	视觉标记选择；跨模态理解
Aguvis [90] :	2024.12	通用	Qwen2-VL	70 亿 / 720 亿	全面数据管线；两阶段训练；跨平台
Aria-UI [96] :	2024.12	屏幕理解，定位	Aria	3.9B	多样化数据集管线；多模态动态动作历史
UI-TARS [92] ⊠	2025.01	通用	Qwen2-VL	20 亿 / 70 亿 / 720 亿	系统 2 推理；在线自举；反思调优
GUI-Bee [100] □	2025.01	屏幕理解，定位	SeeClick，UIX-7B，Qwen-GUI	不适用	模型-环境对齐；自我探索数据
V-Droid [114]	2025.03	通用	Llama-3.1-8B	8b	验证者驱动框架
MP-GUI [98] □	2025.03	通用	InternVL2-8B	8B	屏幕理解，指代

tasks, which are essential for enabling AI agents to interact effectively with graphical user interfaces. These tasks can be categorized into three main types: UI grounding, UI referring, and screen question answering (QA). Figure 10 illustrates the differences between these categories.

这些任务对于使人工智能代理能够有效地与图形用户界面交互至关重要。这些任务可分为三大类: 界面定位 (UI grounding)、界面指代 (UI referring) 和屏幕问答 (QA)。图 10 展示了这些类别之间的区别。

- UI Grounding involves identifying and localizing UI elements on a screen that correspond to a given natural language description. This task is critical for agents to perform precise interactions with GUIs based on user instructions. MUG [93] proposes guiding agent actions through multi-round interactions with users, improving the execution accuracy of UI grounding in complex or ambiguous instruction scenarios. It also leverages user instructions and previous interaction history to predict the next agent action. LVG (Layout-guided Visual Grounding) [94] addresses UI grounding by unifying detection and grounding of UI elements within application interfaces. LVG tackles challenges such as application sensitivity, where UI elements with similar appearances have different functions across applications, and context sensitivity, where the functionality of UI elements depends on their context within the interface. By introducing layout-guided contrastive learning, LVG learns the semantics of UI objects from their visual organization and spatial relationships, improving grounding accuracy. UI-Hawk [95] enhances UI grounding by incorporating a history-aware visual encoder and an efficient resampler to process screen sequences during GUI navigation. By understanding historical screens, UI-Hawk improves the agent’s ability to ground UI elements accurately over time. An

automated data curation method generates training data for UI grounding, contributing to the creation of the FunUI benchmark for evaluating screen understanding capabilities. Aria-UI [96] leverages strong MLLMs such as GPT-40 to generate diverse and high-quality element instructions for grounding training. It employs a two-stage training method that incorporates action history in textual or interleaved text-image formats, enabling the model to develop both single-step localization capabilities and multi-step context awareness. This approach demonstrates robust performance and generalization ability across various tasks. Similar research includes GUI-Bee [100], which autonomously explores environments to collect high-quality data, thereby aligning GUI action grounding models with new environments and significantly enhancing model performance. OS-Atlas [97] unifies the action space, enabling models to adapt to UI grounding tasks across multiple platforms. Additionally, TAG (Tuning-free Attention-driven Grounding) [277] introduces a method that leverages the inherent attention mechanisms of pre-trained MLLMs to accurately identify and locate elements within a GUI without the need for tuning. Validation shows that this method performs comparably to or even surpasses tuned approaches across multiple benchmark datasets, demonstrating exceptional generalization capabilities. This offers a new perspective for the application of MLLMs in UI grounding.

- 界面定位 (UI Grounding) 涉及识别和定位屏幕上与给定自然语言描述相对应的界面元素。该任务对于代理根据用户指令执行精确的图形用户界面操作至关重要。MUG [93] 提出通过与用户的多轮交互引导代理动作，提高在复杂或模糊指令场景下界面定位的执行准确性，同时利用用户指令和先前交互历史预测下一步代理动作。LVG(Layout-guided Visual Grounding)[94] 通过统一应用界面中界面元素的检测与定位来解决界面定位问题。LVG 应对了应用敏感性问题，即外观相似但功能不同的界面元素在不同应用中的差异，以及上下文敏感性问题，即界面元素的功能依赖于其在界面中的上下文。通过引入布局引导的对比学习，LVG 从视觉组织和空间关系中学习界面对象的语义，提升定位准确率。UI-Hawk [95] 通过结合历史感知视觉编码器和高效重采样器处理图形界面导航中的屏幕序列，增强了界面定位能力。通过理解历史屏幕，UI-Hawk 提升了代理随时间准确定位界面元素的能力。一种自动化数据整理方法生成界面定位的训练数据，助力 FunUI 基准的构建以评估屏幕理解能力。Aria-UI [96] 利用强大的多模态大语言模型 (MLLMs)，如 GPT-40，生成多样且高质量的元素指令用于定位训练。其采用两阶段训练方法，结合文本或交错文本-图像格式的动作历史，使模型既具备单步定位能力，又具备多步上下文感知能力。该方法在多种任务中表现出强大的性能和泛化能力。类似研究包括 GUI-Bee [100]，通过自主探索环境收集高质量数据，使界面动作定位模型适应新环境，显著提升模型性能。OS-Atlas [97] 统一动作空间，使模型能够适应多平台的界面定位任务。此外，TAG(Tuning-free Attention-driven Grounding)[277] 提出利用预训练多模态大语言模型固有的注意力机制，无需调优即可准确识别和定位图形界面中的元素。验证表明，该方法在多个基准数据集上的表现可与调优方法媲美甚至超越，展现出卓越的泛化能力，为多模态大语言模型在界面定位中的应用提供了新视角。

- UI Referring focuses on generating natural language descriptions for specified UI elements on a screen. This task enables agents to explain UI components to users or other agents, facilitating better communication and interaction. Ferret-UI [101] is a multimodal LLM designed for enhanced understanding of mobile UI screens, emphasizing precise referring and grounding tasks. By incorporating any resolution techniques to handle various screen aspect ratios and dividing screens into sub-images for detailed analysis, Ferret-UI generates accurate descriptions of UI elements. Training on a curated dataset of elementary UI tasks, Ferret-UI demonstrates strong performance in UI referring tasks. Leveraging the Ferret-UI framework, Ferret-UI 2 [102] integrates an adaptive N-grid partitioning mechanism. This system enhances image feature extraction by dynamically resizing grids, thereby improving the model's efficiency and accuracy without sacrificing

resolution. Additionally, Ferret-UI 2 demonstrates remarkable cross-platform portability. Textual Foresight [103] uses user actions as a bridge, requiring the model to predict the global textual description of the next UI state based on the current UI screen and a local action. With limited training data, the Textual Foresight method achieves superior performance compared to similar models, demonstrating exceptional data efficiency. UI-Hawk [95] also contributes to UI referring by defining tasks that require the agent to generate descriptions for UI elements based on their role and context within the interface. By processing screen sequences and understanding the temporal relationships between screens, UI-Hawk improves the agent's ability to refer to UI elements accurately.

- 界面指代 (UI Referring) 侧重于为屏幕上指定的界面元素生成自然语言描述。该任务使代理能够向用户或其他代理解释界面组件，促进更好的沟通与交互。Ferret-UI [101] 是一种多模态大语言模型，专为增强移动界面屏幕的理解而设计，强调精确的指代和定位任务。通过采用任意分辨率技术处理各种屏幕长宽比，并将屏幕划分为子图像进行细致分析，Ferret-UI 生成准确的界面元素描述。基于精心整理的基础界面任务数据集训练，Ferret-UI 在界面指代任务中表现出色。基于 Ferret-UI 框架，Ferret-UI 2 [102] 引入自适应 N 网格划分机制，通过动态调整网格大小提升图像特征提取能力，从而在不牺牲分辨率的前提下提高模型效率和准确性。此外，Ferret-UI 2 展现出卓越的跨平台移植性。Textual Foresight [103] 利用用户动作作为桥梁，要求模型基于当前界面屏幕和局部动作预测下一界面状态的全局文本描述。该方法在有限训练数据下表现优异，显示出卓越的数据效率。UI-Hawk [95] 也对界面指代有所贡献，定义了基于界面元素在界面中的角色和上下文生成描述的任务。通过处理屏幕序列并理解屏幕间的时间关系，UI-Hawk 提升了代理准确指代界面元素的能力。

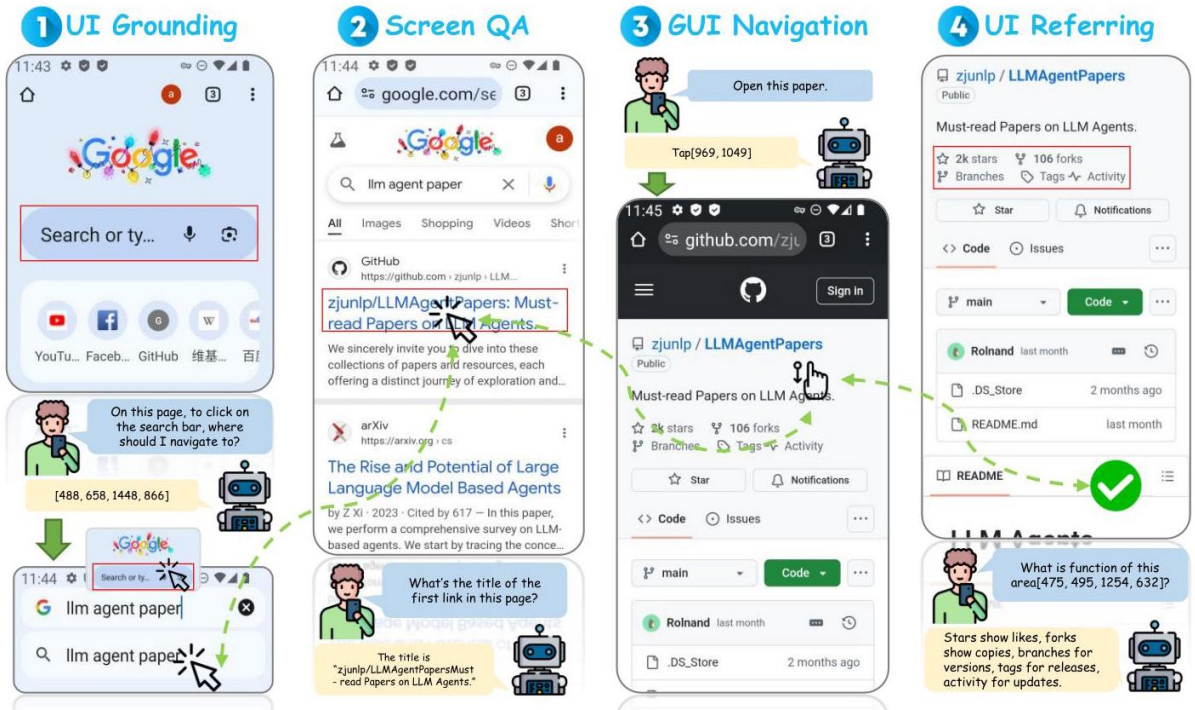

Fig. 10: Illustration of screen understanding tasks. (a) UI Grounding involves identifying UI elements corresponding to a given description; (b) UI Referring focuses on generating descriptions for specified UI elements; (c) Screen Question Answering requires answering questions based on the content of the screen.

图 10: 屏幕理解任务示意图。(a) 界面定位涉及识别与给定描述相对应的界面元素；(b) 界面指代侧重于为指定界面元素生成描述；(c) 屏幕问答要求基于屏幕内容回答问题。

- Screen Question Answering involves answering questions about the content and functionality of a screen based on visual and textual information. This task requires agents to comprehend complex screen layouts and extract relevant information to provide accurate answers. ScreenAI [67] specializes in understanding screen UIs and infographics, leveraging the common visual language and design principles shared between them. By introducing a flexible patching strategy and a novel textual representation for UIs, ScreenAI pre-trains models to interpret UI elements effectively. Using large language models to automatically generate training data, ScreenAI covers tasks such as screen annotation and screen QA. WebVLN [104] extends vision-and-language navigation to websites, where agents navigate based on question-based instructions and answer questions using information extracted from target web pages. By integrating visual inputs, linguistic instructions, and web-specific content like HTML, WebVLN enables agents to understand both the visual layout and underlying structure of web pages, enhancing screen QA capabilities. UI-Hawk [95] further enhances screen QA by enabling agents to process screen sequences and answer questions based on historical interactions. By incorporating screen question answering as one of its fundamental tasks, UI-Hawk improves the agent's ability to comprehend and reason about screen content over time. MP-GUI [98] introduces a specialized MLLM for GUI understanding with three dedicated perceivers for graphical, textual, and spatial modalities. Using a fusion gate to adaptively combine these modalities and an automated data collection pipeline to address training data scarcity, MP-GUI achieves strong performance on GUI understanding tasks including screen QA despite limited training data.

  - 屏幕问答 (Screen Question Answering) 涉及基于视觉和文本信息回答关于屏幕内容和功能的问题。该任务要求智能体理解复杂的屏幕布局并提取相关信息以提供准确答案。ScreenAI [67] 专注于理解屏幕用户界面 (UI) 和信息图表，利用它们之间共享的通用视觉语言和设计原则。通过引入灵活的补丁策略和一种新颖的 UI 文本表示，ScreenAI 对模型进行预训练以有效解读 UI 元素。利用大型语言模型自动生成训练数据，ScreenAI 涵盖了屏幕注释和屏幕问答等任务。WebVLN [104] 将视觉与语言导航扩展到网站，智能体根据基于问题的指令导航，并利用从目标网页提取的信息回答问题。通过整合视觉输入、语言指令和网页特有内容如 HTML，WebVLN 使智能体能够理解网页的视觉布局和底层结构，增强了屏幕问答能力。UI-Hawk [95] 进一步提升屏幕问答，支持智能体处理屏幕序列并基于历史交互回答问题。通过将屏幕问答作为其基本任务之一，UI-Hawk 增强了智能体随时间理解和推理屏幕内容的能力。MP-GUI [98] 引入了一种专门针对图形用户界面 (GUI) 理解的多模态大型语言模型 (MLLM)，配备了针对图形、文本和空间模态的三个专用感知器。通过融合门自适应地结合这些模态，并利用自动化数据收集流程解决训练数据匮乏问题，MP-GUI 在 GUI 理解任务 (包括屏幕问答) 中表现出色，尽管训练数据有限。

These Phone UI-Specific Model Model Architectures demonstrate the importance of focusing on screen understanding tasks to enhance AI agents' interaction with complex user interfaces. By categorizing these tasks into UI grounding, UI referring, and screen question answering, researchers have developed specialized models that address the unique challenges within each category. Integrating innovative techniques such as layout-guided contrastive learning, history-aware visual encoding, and flexible patching strategies has led to significant advancements in agents' abilities to understand, navigate, and interact with GUIs effectively.

这些针对手机用户界面 (UI) 的特定模型架构展示了聚焦屏幕理解任务以提升 AI 智能体与复杂用户界面交互能力的重要性。通过将这些任务分类为 UI 定位 (UI grounding)、UI 指代 (UI referring) 和屏幕问答，研究人员开发了针对各类别独特挑战的专门模型。集成布局引导的对比学习、历史感知视觉编码和灵活补丁策略等创新技术，显著提升了智能体理解、导航和有效交互图形用户界面 (GUI) 的能力。

4.2.2 Supervised Fine-Tuning

4.2.2 监督微调

Supervised fine-tuning has emerged as a crucial technique for enhancing the capabilities of LLMs in GUI tasks within phone automation. By tailoring models to specific tasks through fine-tuning on curated datasets, researchers have significantly improved models' abilities in GUI grounding, optical character recognition (OCR), cross-application navigation, and efficiency. A summary of notable works in this area is presented in Table 4, highlighting their main contributions, domains, and other relevant details.

监督微调已成为提升大型语言模型 (LLMs) 在手机自动化图形用户界面 (GUI) 任务中能力的关键技术。通过在精心策划的数据集上针对特定任务进行微调，研究人员显著增强了模型在 GUI 定位、光学字符识别 (OCR)、跨应用导航及效率方面的表现。表 4 总结了该领域的代表性工作，突出其主要贡献、应用领域及其他相关细节。

Supervised fine-tuning has been effectively applied to develop more versatile and efficient GUI agents by enhancing their fundamental abilities and GUI knowledge. One of the fundamental challenges in developing visual GUI agents is enabling accurate interaction with screen elements based solely on visual inputs, known as GUI grounding. SeeClick [105] addresses this challenge by introducing a visual GUI agent that relies exclusively on screenshots for task automation, circumventing the need for extracted structured data like HTML, which can be lengthy and sometimes inaccessible. Recognizing that GUI grounding is a key hurdle, SeeClick enhances the agent's capability by incorporating GUI grounding pre-training. The authors also introduce ScreenSpot, the first realistic GUI grounding benchmark encompassing mobile, desktop, and web environment. Experimental results demonstrate that improving GUI grounding through supervised fine-tuning directly correlates with enhanced performance in downstream GUI tasks. InfiGUIAgent [106] is trained using a supervised fine-tuning method and employs the Reference-Augmented Annotation approach to fully leverage spatial information, establishing bidirectional connections between GUI elements and text descriptions, thereby enhancing the model's understanding of GUI visual language. Additionally, the model incorporates Hierarchical Reasoning and Expectation-Reflection Reasoning capabilities, enabling the agent to perform complex reasoning natively, which improves its grounding ability. Beyond grounding, agents require robust OCR capabilities and comprehensive knowledge of GUI components and interactions to function effectively across diverse applications. GUICourse [107] tackles these challenges by presenting a suite of datasets designed to train visual-based GUI agents from general VLMs. The GUIEnv dataset strengthens OCR and grounding abilities by providing 10 million website page-annotation pairs for pre-training and 0.7 million region-text QA pairs for supervised fine-tuning. To enrich the agent's understanding of GUI components and interactions, the GUIAct and GUIChat datasets offer extensive single-step and multi-step action instructions and conversational data with text-rich images and bounding boxes. As users frequently navigate across multiple applications to complete complex tasks, enabling cross-app GUI navigation becomes essential for practical GUI agents.GUI Odyssey [109] addresses this need by introducing a comprehensive dataset specifically designed for

training and evaluating cross-app navigation agents. The GUI Odyssey dataset comprises 7,735 episodes from six mobile devices, covering six types of cross-app tasks, 201 apps, and 1,399 app combinations. By fine-tuning the Qwen-VL model with a history resampling module on this dataset, they developed OdysseyAgent, a multimodal cross-app navigation agent. Extensive experiments show that OdysseyAgent achieves superior accuracy compared to existing models, significantly improving both in-domain and out-of-domain performance on cross-app navigation tasks. Efficiency and scalability are also critical considerations, especially for deploying GUI agents on devices with limited computational resources. TinyClick [110] demonstrates that even compact models can achieve strong performance on GUI automation tasks through effective supervised fine-tuning strategies. Utilizing the Vision-Language Model Florence-2- Base, TinyClick focuses on the primary task of identifying the screen coordinates of UI elements corresponding to user commands. By employing multi-task training and Multimodal Large Language Model-based data augmentation, TinyClick significantly improves model performance while maintaining a compact size of 0.27 billion parameters and minimal latency. MobileAgent [111] combines LoRA and SOP methods to effectively reduce computational overhead through low-rank adaptive supervised fine-tuning, while breaking down complex tasks into subtasks to enhance the model's understanding and execution efficiency. At the same time, this approach does not impose additional burdens on inference speed, significantly improving the model's performance and responsiveness. The performance of agents is often limited by their inability to recover from errors. Agent-R [108] identifies the first error step in an erroneous trajectory and combines it with a correct trajectory to create a corrected path, thus enabling real-time error correction. By training on self-generated corrected trajectories and using an iterative supervised fine-tuning approach, Agent-R dynamically identifies and rectifies errors, gradually enhancing decision-making abilities. Moreover, under a multi-task training strategy, its training outcomes improve significantly. This method offers new directions for developing more intelligent and adaptable GUI agents. Supervised fine-tuning has also been applied to domain-

监督微调已被有效应用于开发更通用且高效的 GUI 代理，通过增强其基础能力和 GUI 知识。开发视觉 GUI 代理的一个基本挑战是仅基于视觉输入实现对屏幕元素的准确交互，这被称为 GUI 定位 (GUI grounding)。SeeClick [105] 通过引入一个完全依赖截图进行任务自动化的视觉 GUI 代理来应对这一挑战，避免了对 HTML 等结构化数据的提取需求，这类数据往往冗长且有时难以获取。鉴于 GUI 定位是关键难题，SeeClick 通过引入 GUI 定位预训练提升了代理的能力。作者还推出了 ScreenSpot，这是首个涵盖移动端、桌面端和网页环境的真实 GUI 定位基准。实验结果表明，通过监督微调提升 GUI 定位能力与下游 GUI 任务性能的提升直接相关。InfiGUIAgent [106] 采用监督微调方法训练，利用参考增强注释 (Reference-Augmented Annotation) 方法充分利用空间信息，在 GUI 元素与文本描述之间建立双向连接，从而增强模型对 GUI 视觉语言的理解。此外，模型还融合了层级推理 (Hierarchical Reasoning) 和期望反射推理 (Expectation-Reflection Reasoning) 能力，使代理能够原生执行复杂推理，提升其定位能力。除了定位，代理还需具备强大的 OCR 能力及对 GUI 组件和交互的全面知识，才能在多样化应用中有效运行。GUICourse [107] 通过提供一套数据集，旨在从通用视觉语言模型 (VLMs) 训练基于视觉的 GUI 代理，解决这些挑战。GUIEnv 数据集通过提供 1000 万网页页面-注释对用于预训练，以及 70 万区域-文本问答对用于监督微调，强化了 OCR 和定位能力。为丰富代理对 GUI 组件和交互的理解，GUIAct 和 GUIChat 数据集提供了大量单步和多步操作指令及包含文本丰富图像和边界框的对话数据。鉴于用户常跨多个应用完成复杂任务，实现跨应用 GUI 导航对实用 GUI 代理至关重要。GUI Odyssey [109] 针对这一需求，推出了专门用于训练和评估跨应用导航代理的综合数据集。GUI Odyssey 数据集包含来自六款移动设备的 7735 个任务片段，涵盖六类跨应用任务、201 款应用及 1399 种应用组合。通过在该数据集上结合历史重采样模块对 Qwen-VL 模型进行微调，开发了 OdysseyAgent，一款多模态跨应用导航代理。大量实验表明，OdysseyAgent 在跨应用导航任务中较现有模型表现出更高准确率，显著提升了域内及域外性能。效率与可扩展性也是关键考量，尤其是在计算资源有限的设备上部署 GUI 代理。TinyClick [110] 展示了即使是紧凑模型，通过有效的监督微调策略也能在 GUI 自动化任务中取得优异表现。TinyClick 基于视觉语言模型 Florence-2-Base，聚焦于识别与用户指令对应的 UI 元素屏幕坐标的核心任务。通过多任务训练和基于多模态大语言模型的数据增强，TinyClick 显著提升了模型性能，同时保持了 2.7 亿参数的紧凑规模和极低延迟。MobileAgent [111] 结合 LoRA 和 SOP 方法，通过低秩自适应监督微调有效降低计算开销，同时将复杂任务拆解为子任务，提升模型理解与执行效率。该方法在不增加推理延迟的前提下，显著提升了模型性能和响应速度。代理性能常受限于其错误恢复能力。Agent-R [108] 识别错误轨迹中的首个错误步骤，并结合正确轨迹生成修正路径，实现实时错误纠正。通过在自生成的修正轨迹上训练并采用迭代监督微调方法，Agent-R 动态识别并修正错误，逐步提升决策能力。此外，在多任务训练策略下，其训练效果显著提升。该方法为开发更智能、更具适应性的 GUI 代理提供了新方向。监督微调也已应用于领域-

specific tasks to address specialized challenges in particular contexts, such as reference resolution and accessibility. In the context of Reference Resolution in GUI Contexts, ReALM [112] formulates reference resolution as a language modeling problem, enabling the model to handle various types of references, including on-screen entities, conversational entities, and background entities. By converting reference resolution into a multiple-choice task for the LLM, ReALM significantly improves the model's ability to resolve references in GUI contexts. For Accessibility and UI Icons Alt-Text Generation, IconDesc [115] addresses the challenge of generating informative alt-text for mobile UI icons, which is essential for users relying on screen readers. Traditional deep learning approaches require extensive datasets and struggle with the diversity and imbalance of icon types. IconDesc introduces a novel method using Large Language Models to autonomously generate alt-text with partial UI data, such as class, resource ID, bounds, and contextual information from parent and sibling nodes. By fine-tuning an off-the-shelf LLM on a small dataset of approximately 1.4k icons, IconDesc demonstrates significant improvements in generating relevant alt-text, aiding developers in enhancing UI accessibility during app development.

针对特定情境中的专业挑战设计的具体任务，例如指代消解和无障碍性。在图形用户界面 (GUI) 环境中的指代消解方面，ReALM [112] 将指代消解问题表述为语言建模问题，使模型能够处理各种类型的指代，包括屏幕上的实体、对话中的实体和背景实体。通过将指代消解转化为大型语言模型 (LLM) 的多项选择任务，ReALM 显著提升了模型在 GUI 环境中解决指代的能力。针对无障碍性和用户界面图标替代文本生成，IconDesc [115] 解决了为移动 UI 图标生成信息丰富的替代文本的挑战，这对于依赖屏幕阅读器的用户至关重要。传统的深度学习方法需要大量数据集，并且难以应对图标类型的多样性和不平衡。IconDesc 引入了一种利用大型语言模型自主生成替代文本的新方法，使用部分 UI 数据，如类别、资源 ID、边界以及来自父节点和兄弟节点的上下文信息。通过在约 1.4 千个图标的小型数据集上微调现成的 LLM，IconDesc 在生成相关替代文本方面表现出显著提升，帮助开发者在应用开发过程中增强 UI 的无障碍性。

TABLE 4: Summary of supervised fine-tuning methods for phone GUI agents

表 4: 手机 GUI 代理的监督微调方法总结

Method	Date	Task Type	Backbone	Size	Contributions
MobileAgent [111] ?	2024.01	General	Qwen	7B	Standard Operating Procedure; Human-machine interaction
SeeClick [105] ?	2024.01	General	Qwen-VL	9.6B	GUI grounding pre-training; ScreenSpot benchmark
ReALM [112]	2024.04	Reference Resolution	FLAN-T5	80M-3B	Formulated reference resolution as language modeling; Improved performance on resolving references
GUICourse [107] ?	2024.06	General	Qwen-VL, Fuyu-8B, MiniCPM-V	N/A	Suite of datasets (GUIEnv, GUIAct, GUIChat); Enhanced OCR and grounding
GUI Odyssey [109] ?	2024.06	General	Qwen-VL	N/A	Cross-app navigation dataset; Agent with history resampling
IconDesc [115]	2024.09	Alt-Text Generation	GPT-3.5	N/A	Generated alt-text for UI icons using partial UI data; Improved accessibility
TinyClick [110] ?	2024.10	General	Florence-2	0.27B	Single-turn agent; Multitask training; MLLM-based data augmentation
InfiGUIAgent [106] ?	2025.01	General	Qwen2-VL	2B	Model-Environment alignment; Self-exploratory Data
Agent-R [108] ?	2025.01	General	LLama-3.1	8B	Self-reflection capabilities; Real-time error correction

方法	日期	任务类型	骨干网络	规模	贡献
MobileAgent [111] ?	2024.01	通用	Qwen	7B	标准操作流程；人机交互
SeeClick [105] ?	2024.01	通用	Qwen-VL	9.6B	图形用户界面 (GUI) 定位预训练；ScreenSpot 基准测试
ReALM [112]	2024.04	指代消解	FLAN-T5	80M-3B	将指代消解表述为语言建模；提升指代消解性能
GUICourse [107] ?	2024.06	通用	Qwen-VL, Fuyu-8B, MiniCPM-V	不适用	数据集套件 (GUIEnv, GUIAct, GUIChat)；增强光学字符识别 (OCR) 和定位能力
GUI Odyssey [109] ?	2024.06	通用	Qwen-VL	不适用	跨应用导航数据集；带历史重采样的代理
IconDesc [115]	2024.09	替代文本生成	GPT-3.5	不适用	利用部分 UI 数据生成界面图标的替代文本；提升无障碍性
TinyClick [110] ?	2024.10	通用	Florence-2	0.27B	单轮代理；多任务训练；基于多模态大模型 (MLLM) 的数据增强
InfiGUIAgent [106] ?	2025.01	通用	Qwen2-VL	2B	模型与环境对齐；自我探索数据
Agent-R [108] ?	2025.01	通用	LLama-3.1	8B	自我反思能力；实时错误纠正

These works collectively demonstrate that supervised fine-tuning is instrumental in advancing GUI agents for phone automation. By addressing specific challenges through targeted datasets and training strategies—whether enhancing GUI grounding, improving OCR and GUI knowledge, enabling cross-app navigation, or optimizing for accessibility-researchers have significantly enhanced the performance and applicability of GUI agents. The advancements summarized in Figure 4 highlight the ongoing efforts and progress in this field, paving the way for more intelligent, versatile, and accessible phone automation solutions capable of handling complex tasks in diverse environment.

这些工作共同表明，监督微调在推动手机自动化图形用户界面 (GUI) 代理的发展中起到了关键作用。通过针对特定挑战采用定向数据集和训练策略——无论是增强 GUI 定位、提升光学字符识别 (OCR) 和 GUI 知识、实现跨应用导航，还是优化无障碍功能——研究人员显著提升了 GUI 代理的性能和适用性。图 4 总结的进展凸显了该领域持续的努力和发展，为更智能、多功能且易于访问的手机自动化解决方案铺平了道路，使其能够在多样化环境中处理复杂任务。

4.2.3 Reinforcement Learning

4.2.3 强化学习

Reinforcement Learning (RL) [278] has emerged as a powerful technique for training agents to interact autonomously with GUIs across various platforms, including phones, web browsers, and desktop environment. Although RL-based approaches for phone GUI agents are relatively few, significant progress has been made in leveraging RL to enhance agent capabilities in dynamic and complex GUI environment. In this section, we discuss RL approaches for GUI agents across different platforms, highlighting their unique challenges, methodologies, and contributions. A summary of notable RL-based methods is presented in Figure 5, which includes specific RL-related features such as the type of RL used (online or offline) and the targeted platform.

强化学习 (Reinforcement Learning, RL)[278] 已成为训练代理自主与各种平台上的 GUI 交互的强大技术，涵盖手机、网页浏览器和桌面环境。尽管基于 RL 的手机 GUI 代理方法相对较少，但在利用 RL 提升代理在动态复杂 GUI 环境中的能力方面已取得显著进展。本节讨论了跨平台 GUI 代理的 RL 方法，重点介绍其独特挑战、方法论和贡献。图 5 总结了若干重要的基于 RL 的方法，包含 RL 类型 (在线或离线) 及目标平台等具体特征。

Phone Agents. Training phone GUI agents using RL presents unique challenges due to the dynamic and complex nature of mobile applications. Agents must adapt to real-world stochasticity and handle the intricacies of interacting with diverse mobile environment. Recent works have addressed these challenges by developing RL frameworks that enable agents to learn from interactions and improve over time. DigiRL [116] andDistRL [117] both tackle the limitations of pre-trained vision-language models (VLMs) in decision-making tasks for device control through GUIs. Recognizing that static demonstrations are insufficient due to the dynamic nature of real-world mobile environment, these works introduce RL approaches to train agents capable of in-the-wild device control. DigiRL proposes an autonomous RL framework that employs a two-stage training process: an initial offline RL phase to initialize the agent using existing data, followed by an offline-to-online RL phase that fine-tunes the model based on its own interactions. By building a scalable Android learning environment with a VLM-based evaluator, DigiRL identifies key design choices for effective RL in mobile GUI domains. The agent learns to handle real-world stochasticity and dynamism, achieving significant improvements over supervised fine-tuning, with a 49.5% absolute increase in success rate on the Android-in-the-Wild dataset. Similarly, DistRL introduces an asynchronous distributed RL framework specifically designed for on-device control agents on mobile devices. To address inefficiencies in online fine-tuning and the challenges posed by dynamic mobile environment, DistRL employs centralized training and decentralized data acquisition. Leveraging an off-policy RL algorithm tailored for distributed and asynchronous data utilization, DistRL improves training efficiency and agent performance by prioritizing significant experiences and encouraging exploration. Experiments show that DistRL achieves a 20% relative improvement in success rate compared to state-of-the-art methods on general Android tasks. Building upon these advancements, AutoGLM [118] extends the application of RL to both phone and web platforms. AutoGLM presents a series of foundation agents based on the ChatGLM model family, aiming to serve as autonomous agents for GUI control. A key insight from this work is the design of an intermediate interface that separates planning and grounding behaviors, allowing for more agile development and enhanced performance. By employing self-evolving online curriculum RL, AutoGLM enables agents to learn from environmental interactions and adapt to dynamic GUI environment. The approach demonstrates impressive success rates on various benchmarks, showcasing the potential of RL in creating versatile GUI agents across platforms.

手机代理。由于移动应用的动态性和复杂性，使用 RL 训练手机 GUI 代理面临独特挑战。代理必须适应现实世界的随机性，并处理与多样移动环境交互的复杂性。近期工作通过开发 RL 框架，使代理能够从交互中学习并持续改进，解决了这些挑战。DigiRL [116] 和 DistRL [117] 均针对预训练视觉语言模型 (VLMs) 在设备控制决策任务中的局限性提出解决方案。鉴于静态示范不足以应对现实移动环境的动态性，这些工作引入 RL 方法训练能够在真实环境中控制设备的代理。DigiRL 提出了一个自主 RL 框架，采用两阶段训练流程: 初始离线 RL 阶段利用现有数据初始化代理，随后离线到在线 RL 阶段基于代理自身交互进行微调。通过构建基于 VLM 评估器的可扩展 Android 学习环境，DigiRL 确定了移动 GUI 领域有效 RL 的关键设计选择。代理学会应对现实世界的随机性和动态性，在 Android-in-the-Wild 数据集上成功率较监督微调提升了 49.5 个百分点。同样，DistRL 引入了专为移动设备上设备控制代理设计的异步分布式 RL 框架。为解决在线微调效率低下及动态移动环境带来的挑战，DistRL 采用集中训练与分散数据采集相结合的方法。利用适合分布式异步数据利用的离策略 RL 算法，DistRL 通过优先处理重要经验和鼓励探索，提高了训练效率和代理性能。实验表明，DistRL 在通用 Android 任务上较最先进方法成功率提升了 20%。基于这些进展，AutoGLM [118] 将 RL 应用扩展至手机和网页平台。AutoGLM 提出了一系列基于 ChatGLM 模型家族的基础代理，旨在作为 GUI 控制的自主代理。该工作的一大关键见解是设计了一个分离规划与定位行为的中间接口，促进了更灵活的开发和性能提升。通过采用自我进化的在线课程 RL，AutoGLM 使代理能够从环境交互中学习并适应动态 GUI 环境。该方法在多个基准测试中展现了令人印象深刻的成功率，展示了 RL 在跨平台创建多功能 GUI 代理中的潜力。

TABLE 5: Summary of reinforcement learning methods for phone GUI agents

表 5: 手机 GUI 代理强化学习方法总结

Method	Date	Platform	RL Type	Backbone	Size
DigiRL [116] □	2024.06	Phone	Online RL	AutoUI-Base	200M
DistRL [117] □	2024.10	Phone	Online RL	T5-based	1.3B
AutoGLM [118] □	2024.11	Phone, Web	Online RL	GLM-4-9B-Base	9B
ScreenAgent [128] □	2024.02	PC OS	N/A	CogAgent	18B
ETO [124] □	2024.03	Web	Offline-to-Online RL	LLaMA-2-7B-Chat	7B
AutoWebGLM [126] □	2024.04	Web	RL (Curriculum Learning, Bootstrapped RL)	ChatGLM3-6B	6B
Agent Q [125] □	2024.08	Web	Offline RL with MCTS	LLaMA-3-70B	70B
GLAINTEL [127] □	2024.11	Web	RL (Offline-to-Online, Hybrid RL)	Flan-T5	0.78B
ReachAgent [120]	2025.02	Phone	Hybrid RL	MobileVLM [63]	N/A
VEM [279] ?	2025.02	Phone	Environment-Free RL	N/A	N/A
Digi-Q [119] ?	2025.02	Phone	O-Function Based RL	N/A	N/A
VSC-RL [121] ?	2025.02	Phone	Variational Subgoal-Conditioned RL	N/A	N/A
UI-R1 [122] ⊠	2025.03	Phone	Rule-Based RL	Qwen2.5-VL	3B

方法	日期	平台	强化学习类型	骨干网络	规模
DigiRL [116] □	2024.06	手机	在线强化学习	AutoUI-Base	200M
DistRL [117] □	2024.10	手机	在线强化学习	基于 T5	1.3B
AutoGLM [118] □	2024.11	手机，网页	在线强化学习	GLM-4-9B-Base	9B
ScreenAgent [128] □	2024.02	PC 操作系统	不适用	CogAgent	18B
ETO [124] □	2024.03	网页	离线到在线强化学习	LLaMA-2-7B-Chat	7B
AutoWebGLM [126] □	2024.04	网页	强化学习 (课程学习，Bootstrapped RL)	ChatGLM3-6B	6B
Agent Q [125] □	2024.08	网页	基于蒙特卡洛树搜索的离线强化学习	LLaMA-3-70B	70B
GLAINTEL [127] □	2024.11	网页	强化学习 (离线到在线，混合强化学习)	Flan-T5	0.78B
ReachAgent [120]	2025.02	手机	混合强化学习	MobileVLM [63]	不适用
VEM [279] ?	2025.02	手机	无环境强化学习	不适用	不适用
Digi-Q [119] ?	2025.02	手机	基于 O 函数的强化学习	不适用	不适用
VSC-RL [121] ?	2025.02	手机	变分子目标条件强化学习	不适用	不适用
UI-R1 [122] ⊠	2025.03	手机	基于规则的强化学习	Qwen2.5-VL	3B

Recent advances have brought several innovative approaches to reinforcement learning for phone GUI agents. ReachAgent [120] decomposes mobile agent tasks into two sub-tasks: page reaching and page operation, utilizing a two-stage fine-tuning strategy. In the first stage, supervised fine-tuning enables the agent to better perform each subtask. In the second stage, reinforcement learning is applied to further optimize the agent's overall task completion capabilities, thereby enhancing its performance in complex tasks. VEM [279] introduces an environment-free RL framework that decouples value estimation from policy optimization using a pretrained Value Environment Model. Unlike traditional RL methods that require costly environment interactions, VEM predicts state-action values directly from offline data, distilling human-like priors about GUI interaction outcomes. This approach avoids compounding errors and enhances resilience to UI changes by focusing on semantic reasoning. Digi-Q [119] presents an approach to train VLM-based action-value Q-functions for device control. Instead of using on-policy RL with actual environment rollouts, Digi-Q trains the Q-function using offline temporal-difference learning on frozen, intermediate-layer features of a VLM. This approach enhances scalability and reduces computational costs compared to fine-tuning the entire VLM. The trained Q-function then uses a Best-of-N policy extraction operator to imitate the best action without requiring environment interaction. VSC-RL [121] addresses the learning inefficiencies in tackling complex sequential decision-making tasks with sparse rewards and long-horizon dependencies. By reformulating vision-language sequential tasks as a variational goal-conditioned RL problem, VSC-RL optimizes the SubGoal Evidence Lower BOund (SGC-ELBO). This approach maximizes subgoal-conditioned return via RL while minimizing the difference with the reference policy. UI-R1 [122] explores how rule-based RL can enhance reasoning capabilities of multimodal large language models for GUI action prediction. Using a small yet high-quality dataset of 136 challenging tasks, UI-R1 introduces a unified rule-based action reward enabling model optimization via Group Relative Policy Optimization (GRPO).

近期进展带来了多种创新方法，用于手机 GUI 代理的强化学习。ReachAgent [120] 将移动代理任务分解为两个子任务: 页面到达和页面操作，采用两阶段微调策略。第一阶段通过监督微调使代理更好地完成各子任务。第二阶段应用强化学习进一步优化代理的整体任务完成能力，从而提升其在复杂任务中的表现。VEM [279] 引入了一种无环境强化学习框架，利用预训练的价值环境模型 (Value Environment Model) 将价值估计与策略优化解耦。不同于传统强化学习方法需要昂贵的环境交互，VEM 直接从离线数据预测状态-动作值，提炼了关于 GUI 交互结果的人类先验知识。该方法避免了误差累积，并通过关注语义推理增强了对 UI 变化的鲁棒性。Digi-Q [119] 提出了一种训练基于视觉语言模型 (VLM) 的动作价值 Q 函数以控制设备的方法。Digi-Q 不使用基于环境实际回滚的在线策略强化学习，而是在冻结的 VLM 中间层特征上通过离线时序差分学习训练 Q 函数。与微调整个 VLM 相比，该方法提升了可扩展性并降低了计算成本。训练好的 Q 函数随后使用 Best-of-N 策略提取算子模仿最佳动作，无需环境交互。VSC-RL [121] 解决了在稀疏奖励和长时依赖的复杂序列决策任务中学习效率低下的问题。通过将视觉语言序列任务重新表述为变分目标条件强化学习问题，VSC-RL 优化子目标证据下界 (SubGoal Evidence Lower BOund, SGC-ELBO)。该方法通过强化学习最大化子目标条件回报，同时最小化与参考策略的差异。UI-R1 [122] 探讨了基于规则的强化学习如何增强多模态大型语言模型在 GUI 动作预测中的推理能力。利用包含 136 个挑战性任务的小而高质量数据集，UI-R1 引入了统一的基于规则的动作奖励，使模型能够通过群体相对策略优化 (Group Relative Policy Optimization, GRPO) 进行优化。

Web Agents. Web navigation tasks involve interacting with complex and dynamic web environment, where agents must interpret web content and perform actions to achieve user-specified goals. RL has been employed to train agents that can adapt to these challenges by learning from interactions and improving decision-making capabilities. ETO [124] (Exploration-based Trajectory Optimization) and Agent Q [125] both focus on enhancing the performance of LLM-based agents in web environment through RL techniques. ETO introduces a learning method that allows agents to learn from their exploration failures by iteratively collecting failure trajectories and using them to create contrastive trajectory pairs for training. By leveraging contrastive learning methods like Direct Preference Optimization (DPO), ETO enables agents to improve performance through an iterative cycle of exploration and training. Experiments on tasks such as WebShop demonstrate that ETO consistently outperforms baselines, highlighting the effectiveness of learning from failures. Agent Q combines guided Monte Carlo Tree Search (MCTS) with a self-critique mechanism and iterative fine-tuning using an off-policy variant of DPO. This framework allows LLM agents to learn from both successful and unsuccessful trajectories, improving generalization in complex, multi-step reasoning tasks. Evaluations on the WebShop environment and real-world booking scenarios show that Agent Q significantly improves success rates, outperforming behavior cloning and reinforcement learning fine-tuned baselines. AutoWebGLM [126] contributes to this domain by developing an LLM-based web navigating agent built upon ChatGLM3-6B. To address the complexity of HTML data and the versatility of web actions, AutoWebGLM introduces an HTML simplification algorithm to represent webpages succinctly. The agent is trained using a hybrid human-AI method to build web browsing data for curriculum training and is further enhanced through reinforcement learning and rejection sampling. AutoWebGLM demonstrates performance superiority on general webpage browsing tasks, achieving practical usability in real-world services. GLAIN-TEL [127] effectively utilizes human experience and the adaptive capabilities of reinforcement learning by integrating human demonstrations with reinforcement learning methods. This approach achieves superior performance in complex product search tasks. Collectively, these works demonstrate how RL techniques can be applied to web agents to improve their ability to navigate and interact with complex web environment. By learning from interactions, failures, and leveraging advanced planning methods, these agents exhibit enhanced reasoning and decision-making capabilities.

网页代理。网页导航任务涉及与复杂且动态的网页环境交互，代理必须理解网页内容并执行操作以实现用户指定的目标。强化学习被用于训练能够适应这些挑战的代理，通过交互学习并提升决策能力。ETO [124](基于探索的轨迹优化) 和 Agent Q [125] 均聚焦于通过强化学习技术提升基于大型语言模型 (LLM) 的代理在网页环境中的表现。ETO 引入了一种学习方法，使代理能够从探索失败中学习，通过迭代收集失败轨迹并利用它们创建对比轨迹对进行训练。借助直接偏好优化 (Direct Preference Optimization, DPO) 等对比学习方法，ETO 通过探索与训练的迭代循环提升代理性能。在 WebShop 等任务上的实验表明, ETO 持续优于基线方法, 凸显了从失败中学习的有效性。Agent Q 结合了引导蒙特卡洛树搜索 (MCTS)、自我批评机制以及使用 DPO 的离策略变体进行迭代微调。该框架使 LLM 代理能够从成功和失败轨迹中学习, 提升在复杂多步推理任务中的泛化能力。在 WebShop 环境和真实预订场景中的评估显示，Agent Q 显著提高了成功率，优于行为克隆和强化学习微调的基线。AutoWebGLM [126] 通过基于 ChatGLM3-6B 构建的 LLM 网页导航代理为该领域做出贡献。为应对 HTML 数据的复杂性和网页操作的多样性，AutoWebGLM 引入了 HTML 简化算法以简洁表示网页。该代理采用人机混合方法训练，构建网页浏览数据用于课程训练，并通过强化学习和拒绝采样进一步提升。AutoWebGLM 在通用网页浏览任务中表现优异，实现了现实服务中的实用性。GLAIN-TEL [127] 通过整合人类示范与强化学习方法，有效利用人类经验和强化学习的自适应能力，在复杂产品搜索任务中取得了优异表现。总体来看，这些工作展示了强化学习技术如何应用于网页代理，提升其在复杂网页环境中的导航和交互能力。通过从交互、失败中学习并利用先进的规划方法，这些代理展现出增强的推理和决策能力。

PC OS Agents. In desktop environment, agents face the challenge of interacting with complex software applications and operating systems, requiring precise control actions and understanding of GUI elements. RL approaches in this domain focus on enabling agents to perform multistep tasks and adapt to the intricacies of desktop GUIs. ScreenAgent [128] constructs an environment where a Vision Language Model (VLM) agent interacts with a real computer screen via the VNC protocol. By observing screenshots and manipulating the GUI through mouse and keyboard actions, the agent operates within an automated control pipeline that includes planning, acting, and reflecting phases. This design allows the agent to continuously interact with the environment and complete multi-step tasks. ScreenAgent introduces the ScreenAgent Dataset, which collects screen-shots and action sequences for various daily computer tasks. The trained model demonstrates computer control capabilities comparable to GPT-4V and exhibits precise UI positioning capabilities, highlighting the potential of RL in desktop GUI automation. AssistGUI [129] develops an LLM-based reinforcement learning framework called Actor-Critic Embodied Agent (ACE). This framework automates desktop GUI through visual analysis, reasoning, and action generation, significantly improving task success rates. Additionally, it introduces a novel benchmarking framework to evaluate a model's ability to complete complex tasks on desktop platforms using mouse and keyboard operations. This advancement offers a new direction for future research in desktop GUI automation.

PC 操作系统代理。在桌面环境中，代理面临与复杂软件应用和操作系统交互的挑战，要求精确的控制操作和对 GUI 元素的理解。该领域的强化学习 (RL) 方法侧重于使代理能够执行多步骤任务并适应桌面 GUI 的复杂性。ScreenAgent [128] 构建了一个环境，使视觉语言模型 (VLM) 代理通过 VNC 协议与真实计算机屏幕交互。通过观察屏幕截图并通过鼠标和键盘操作操控 GUI，代理在包含规划、执行和反思阶段的自动控制流程中运行。该设计使代理能够持续与环境交互并完成多步骤任务。ScreenAgent 引入了 ScreenAgent 数据集，收集了各种日常计算机任务的屏幕截图和操作序列。训练后的模型展示了与 GPT-4V 相当的计算机控制能力，并表现出精确的 UI 定位能力，凸显了强化学习在桌面 GUI 自动化中的潜力。AssistGUI [129] 开发了基于大型语言模型 (LLM) 的强化学习框架，称为 Actor-Critic Embodied Agent(ACE)。该框架通过视觉分析、推理和动作生成实现桌面 GUI 自动化，显著提升了任务成功率。此外，它引入了一个新颖的基准测试框架，用于评估模型在桌面平台上通过鼠标和键盘操作完成复杂任务的能力。这一进展为未来桌面 GUI 自动化研究提供了新方向。

Reinforcement Learning has proven to be a valuable approach for training GUI agents across various platforms, enabling them to learn from interactions with dynamic environment and improve their performance over time. By leveraging RL techniques, these agents can adapt to real-world stochasticity, handle complex decision-making tasks, and exhibit enhanced autonomy in phone, web, and desktop environment. The works discussed in this section showcase the progress made in developing intelligent and versatile GUI agents through RL, paving the way for enhanced automation and user interaction across diverse platforms.

强化学习已被证明是训练跨多平台 GUI 代理的有效方法，使其能够从与动态环境的交互中学习并随着时间提升性能。通过利用强化学习技术，这些代理能够适应现实世界的随机性，处理复杂的决策任务，并在手机、网页和桌面环境中展现出更强的自主性。本节讨论的工作展示了通过强化学习开发智能且多功能 GUI 代理的进展，为各类平台上的自动化和用户交互提升铺平了道路。

5 DATASETS AND BENCHMARKS

5 数据集与基准测试

The rapid evolution of mobile technology has transformed smartphones into indispensable tools for communication, productivity, and entertainment. This shift has spurred a growing interest in developing intelligent agents capable of automating tasks and enhancing user interactions with mobile devices. These agents rely on a deep understanding of GUIs and the ability to interpret and execute instructions effectively. However, the development of such agents presents significant challenges, including the need for diverse datasets, standardized benchmarks, and robust evaluation methodologies.

移动技术的快速发展使智能手机成为通信、生产力和娱乐的不可或缺工具。这一转变激发了开发能够自动化任务并增强用户与移动设备交互的智能代理的兴趣。这些代理依赖于对 GUI 的深刻理解以及有效解释和执行指令的能力。然而，开发此类代理面临重大挑战，包括对多样化数据集、标准化基准和稳健评估方法的需求。

Datasets serve as the backbone for training and testing phone GUI agents, offering rich annotations and task diversity to enable these agents to learn and adapt to complex environment. Complementing these datasets, benchmarks provide structured environment and evaluation metrics, allowing researchers to assess agent performance in a consistent and reproducible manner. Together, datasets and benchmarks form the foundation for advancing the

capabilities of GUI-based agents.

数据集是训练和测试手机 GUI 代理的基础，提供丰富的注释和任务多样性，使代理能够学习并适应复杂环境。与数据集相辅相成的是基准测试，它们提供结构化环境和评估指标，使研究人员能够以一致且可复现的方式评估代理性能。数据集与基准测试共同构成了推动基于 GUI 代理能力提升的基石。

This section delves into the key datasets and benchmarks that have shaped the field. Subsection 5.1 reviews notable datasets that provide the training data necessary for enabling agents to perform tasks such as language grounding, UI navigation, and multimodal interaction. Subsection 5.2 discusses benchmarks that facilitate the evaluation of agent performance, focusing on their contributions to reproducibility, generalization, and scalability. Through these resources, researchers and developers gain the tools needed to push the boundaries of intelligent phone automation, moving closer to creating agents that can seamlessly assist users in their daily lives.

本节深入探讨塑造该领域的关键数据集和基准测试。5.1 小节回顾了提供训练数据的著名数据集，这些数据集支持代理执行语言基础、UI 导航和多模态交互等任务。5.2 小节讨论了促进代理性能评估的基准测试，重点介绍其在可复现性、泛化性和可扩展性方面的贡献。通过这些资源，研究人员和开发者获得了推动智能手机自动化边界的工具，朝着创建能够无缝辅助用户日常生活的代理迈进。

5.1 Datasets

5.1 数据集

The development of phone automation and GUI-based agents has been significantly propelled by the availability of diverse and richly annotated datasets. These datasets provide the foundation for training and evaluating models that can understand and interact with mobile user interfaces using natural language instructions. In this subsection, we review several key datasets, highlighting their unique contributions and how they collectively advance the field. Table 6 summarizes these datasets, providing an overview of their characteristics.

手机自动化和基于 GUI 的代理的发展得益于多样且丰富注释的数据集的可用性。这些数据集为训练和评估能够理解并使用自然语言指令与移动用户界面交互的模型提供了基础。本小节回顾了若干关键数据集，突出其独特贡献及其如何共同推动该领域发展。表 6 总结了这些数据集，概述了它们的特征。

Rico [130] is the largest dataset from the early stage of GUI automation development, providing a solid foundation for understanding modern mobile interfaces and developing GUI agents. It includes various types of data, such as UI screenshots, view hierarchies, and UI metadata, offering valuable references for researchers and developers. Based on this, subsequent studies like RICO Semantics [131], GUI-WORLD [139], and MobileViews [142] have emerged, expanding the types and coverage of datasets and driving the growth of GUI agent research. Among them, MobileViews is currently the largest GUI dataset.

Rico [130] 是 GUI 自动化早期阶段最大的数据集，为理解现代移动界面和开发 GUI 代理提供了坚实基础。它包含多种类型的数据，如 UI 截图、视图层级和 UI 元数据，为研究人员和开发者提供了宝贵参考。在此基础上，后续研究如 RICO Semantics [131]、GUI-WORLD [139] 和 MobileViews [142] 相继出现，扩展了数据集的类型和覆盖范围，推动了 GUI 代理研究的发展。其中，MobileViews 目前是最大的 GUI 数据集。

Early efforts in dataset creation focused on mapping natural language instructions to UI actions. PixelHelp [132] pioneered this area by introducing a problem of grounding natural language instructions to mobile UI action sequences. It decomposed the task into action phrase extraction and grounding, enabling models to interpret instructions like "Turn on flight mode" and execute corresponding UI actions. Building on this, UGIF [136] extended the challenge to a multilingual and multimodal setting. UGIF addressed cross-modal and cross-lingual retrieval and grounding, providing a dataset with instructions in English and UI interactions across multiple languages, thus highlighting the complexities of multilingual UI instruction following.

早期的数据集创建工作集中于将自然语言指令映射到 UI 操作。PixelHelp [132] 在该领域开创先河，提出了将自然语言指令定位到移动 UI 操作序列的问题。它将任务分解为动作短语提取和定位，使模型能够理解“开启飞行模式”等指令并执行相应的 UI 操作。在此基础上，UGIF [136] 将挑战扩展到多语言和多模态环境。UGIF 解决了跨模态和跨语言的检索与定位问题，提供了包含英文指令和多语言 UI 交互的数据集，凸显了多语言 UI 指令执行的复杂性。

Addressing task feasibility and uncertainty, MoTIF [133] introduced a dataset that includes natural language commands which may not be satisfiable within the given UI context. By incorporating feasibility annotations and followup questions, MoTIF encourages research into how agents can recognize and handle infeasible tasks, enhancing robustness in interactive environment.

针对任务可行性和不确定性，MoTIF [133] 引入了一个包含自然语言命令的数据集，这些命令在给定的 UI 环境中可能无法实现。通过加入可行性注释和后续问题，MoTIF 鼓励研究代理如何识别和处理不可行任务，从而增强交互环境中的鲁棒性。

For advancing UI understanding through pre-training, UIBert [134] proposed a Transformer-based model that jointly learns from image and text representations of UIs. By introducing novel pre-training tasks that leverage the correspondence between different UI features, UIBert demonstrated improvements across multiple downstream UI tasks, setting a foundation for models that require a deep understanding of GUI layouts and components.

为了通过预训练推进 UI 理解，UIBert [134] 提出了一种基于 Transformer 的模型，联合学习 UI 的图像和文本表示。通过引入利用不同 UI 特征对应关系的新型预训练任务，UIBert 在多个下游 UI 任务中表现出提升，为需要深入理解图形用户界面 (GUI) 布局和组件的模型奠定了基础。

In the realm of multimodal dialogues and interactions, Meta-GUI [135] proposed a GUI-based task-oriented dialogue system. This work collected dialogues paired with GUI operation traces, enabling agents to perform tasks through conversational interactions and direct GUI manipulations. It bridges the gap between language understanding and action execution within mobile applications.

在多模态对话与交互领域，Meta-GUI [135] 提出了一种基于 GUI 的任务导向对话系统。该工作收集了配对的对话与 GUI 操作轨迹，使代理能够通过对话交互和直接 GUI 操作完成任务，弥合了移动应用中语言理解与动作执行之间的鸿沟。

Recognizing the need for large-scale datasets to train more generalizable agents, several works introduced extensive datasets capturing a wide range of device interactions. Android In The Wild (AITW) [137] released a dataset containing hundreds of thousands of episodes with human demonstrations of device interactions. It presents challenges where agents must infer actions from visual appearances and handle precise gestures. Building upon AITW, Android In The Zoo (AITZ) [138] provided fine-grained semantic annotations using the Chain-of-Action-Thought (CoAT) paradigm, enhancing agents' ability to reason and make decisions in GUI navigation tasks.

鉴于训练更具泛化能力的代理需要大规模数据集，若干工作引入了涵盖广泛设备交互的大型数据集。Android In The Wild (AITW) [137] 发布了包含数十万条人类设备交互演示的集，提出了代理需从视觉外观推断动作并处理精确手势的挑战。在此基础上，Android In The Zoo (AITZ) [138] 采用行动思维链 (Chain-of-Action-Thought, CoAT) 范式提供了细粒度语义注释，增强了代理在 GUI 导航任务中的推理和决策能力。

To address the complexities of cross-application navigation, GUI Odyssey [109] introduced a dataset specifically designed for training and evaluating agents that navigate across multiple apps. By covering diverse apps, tasks, and devices, GUI Odyssey enables the development of agents capable of handling real-world scenarios that involve integrating multiple applications and transferring context between them.

为解决跨应用导航的复杂性，GUI Odyssey [109] 引入了专门用于训练和评估跨多应用导航代理的数据集。通过涵盖多样的应用、任务和设备，GUI Odyssey 支持开发能够处理涉及多应用集成及上下文传递的真实场景的代理。

Understanding how data scale affects agent performance, AndroidControl [140] studied the impact of training data size on computer control agents. By collecting demonstrations with both high-level and low-level instructions across numerous apps, this work analyzed in-domain and out-of-domain generalization, providing insights into the scalability of fine-tuning approaches for device control agents.

关注数据规模对代理性能的影响，AndroidControl [140] 研究了训练数据量对计算机控制代理的影响。通过收集涵盖众多应用的高层和低层指令演示，该工作分析了域内和域外泛化，提供了设备控制代理微调方法可扩展性的见解。

Focusing on detailed annotations to enhance agents' understanding of UI elements, AMEX [141] introduced a comprehensive dataset with multi-level annotations. It includes GUI interactive element grounding, functionality descriptions, and complex natural language instructions with stepwise GUI-action chains. AMEX aims to align agents more closely with human users by providing fundamental knowledge and understanding of the mobile GUI environment from multiple levels, thus facilitating the training of agents with a deeper understanding of page layouts and UI element functionalities.

聚焦于细致注释以提升代理对 UI 元素的理解，AMEX [141] 引入了一个包含多层次注释的综合数据集。其内容包括 GUI 交互元素定位、功能描述以及带有逐步 GUI 动作链的复杂自然语言指令。AMEX 旨在通过多层次提供移动 GUI 环境的基础知识和理解，使代理更贴近人类用户，从而促进具备更深页面布局和 UI 元素功能理解的代理训练。

TABLE 6: Summary of datasets for phone GUI agents. "Actions" refers to the number of distinct actions available; "Demos" refers to the number of demonstration sequences; "Apps" refers to the number of applications covered; "Instr." refers to the number of natural language instructions; "Avg. Steps" refers to the average number of steps per task.

表 6: 手机 GUI 代理数据集汇总。"Actions"指可用的不同动作数量;"Demos"指演示序列数量;"Apps"指涵盖的应用数量；"Instr."指自然语言指令数量；"Avg. Steps"指每个任务的平均步骤数。

Dataset	Date	Screenshots	UI Trees	Actions	Demos	Apps	Instr.	Avg. Steps	Contributions
Rico [130] ?	2017.10	✓	✓	N/A	10,811	9,772	N/A	N/A	Large-scale mobile dataset
PixelHelp [132] ?	2020.05	✓	✓	4	187	4	187	4.2	Grounding instruc tions to actions
MoTIF [133] ?	2021.04	✓	✓	6	4,707	125	276	4.5	Interactive visual environment
UIBert [134] ?	2021.07	✓	✓	N/A	N/A	N/A	16,660	1	Pre-training task
Meta-GUI [135] ?	2022.05	✘	✓	7	4,684	11	1,125	5.3	Multi-turn dialogues
UGIF [136] ?	2022.11	✓	✓	8	523	12	523	5.3	Multilingual UI-grounded instructions
AITW [137] ?	2023.12	✓	✘	7	715,142	357	30,378	6.5	Large-scale interactions
AITZ [138] ⊠	2024.03	✓	✘	7	18,643	70	2,504	7.5	Chain-of-Action-Thought annotations
GUI Odyssey [109] ⍁	2024.06	✘	✓	9	7,735	201	7,735	15.4	Cross-app navigation
AndroidControl [140] :	2024.07	✓	✓	8	15,283	833	15,283	4.8	UI task scaling law
AMEX [141] □	2024.07	✓	✓	8	2,946	110	2,946	12.8	Multi-level detailed annotations
MobileViews [142] □	2024.09	✓	✓	N/A	N/A	21,053	N/A	N/A	Largest-scale mobile dataset

数据集	日期	截图	界面树	操作	演示	应用	指令	平均步骤	贡献
Rico [130] ?	2017.10	✓	✓	不适用	10,811	9,772	不适用	不适用	大规模移动数据集
PixelHelp [132] ?	2020.05	✓	✓	4	187	4	187	4.2	将指令映射到操作
MoTIF [133] ?	2021.04	✓	✓	6	4,707	125	276	4.5	交互式视觉环境
UIBert [134] ?	2021.07	✓	✓	不适用	不适用	不适用	16,660	1	预训练任务
Meta-GUI [135] ?	2022.05	✘	✓	7	4,684	11	1,125	5.3	多轮对话
UGIF [136] ?	2022.11	✓	✓	8	523	12	523	5.3	多语言界面指令
AITW [137] ?	2023.12	✓	✘	7	715,142	357	30,378	6.5	大规模交互
AITZ [138] ⊠	2024.03	✓	✘	7	18,643	70	2,504	7.5	动作思维链注释
GUI Odyssey [109] ⍁	2024.06	✘	✓	9	7,735	201	7,735	15.4	跨应用导航
AndroidControl [140] :	2024.07	✓	✓	8	15,283	833	15,283	4.8	界面任务规模定律
AMEX [141] □	2024.07	✓	✓	8	2,946	110	2,946	12.8	多层次详细注释
MobileViews [142] □	2024.09	✓	✓	不适用	不适用	21,053	不适用	不适用	最大规模移动数据集

Finally, we should focus on methods for generating, collecting, and annotating high-quality datasets. DreamStruct [280] leverages LLMs to generate data design concept descriptions based on target tasks. It then produces HTML code with target labels, embedding semantic tags within. In the post-processing phase, Bing Search API or DALL-E is used to replace placeholder graphic elements, resulting in the final visual content. This research offers a dataset, DreamUI, which includes 9,774 labeled UI interfaces for reference. OS-Genesis [281] utilizes the method of Reverse Task Synthesis to automatically generate task instructions and corresponding action trajectories from interactions. It then integrates these with a trajectory reward model to produce high-quality and diverse GUI agent data. Learn-by-interact [282] uses LLMs to generate data through interaction with the environment and optimizes this data via backward construction. These high-quality data generation techniques reduce the dependency on manually labeled data, facilitating agents' rapid adaptation to new environments and tasks. Ferret-UI 2 [102]

uses the Set-of-Mark (SoM) visual prompt method to tag each UI component with bounding boxes and numerical labels to assist GPT-40 in recognition. Subsequently, GPT-40 generates question-and-answer task data related to UI components, covering multiple aspects of UI comprehension and thus producing high-quality training data. FedMobileAgent [283] automatically collects data during users' daily mobile usage and employs locally deployed VLM to annotate user actions, thereby generating a high-quality dataset. Furthermore, even in the absence of explicit ground truth annotations, we can infer user intentions through their interactions within the GUI to generate corresponding UI annotations [284]. This approach opens up new directions for the collection and annotation of GUI data.

最后，我们应聚焦于生成、收集和标注高质量数据集的方法。Dream-Struct [280] 利用大型语言模型 (LLMs) 根据目标任务生成数据设计概念描述，随后生成带有目标标签的 HTML 代码，内嵌语义标签。在后处理阶段，使用 Bing 搜索 API 或 DALL-E 替换占位图形元素，形成最终的视觉内容。该研究提供了包含 9,774 个带标签 UI 界面的数据集 DreamUI 供参考。OS-Genesis [281] 采用逆向任务合成 (Reverse Task Synthesis) 方法，从交互中自动生成任务指令及对应动作轨迹，随后结合轨迹奖励模型，产出高质量且多样化的 GUI 代理数据。Learn-by-interact [282] 利用 LLMs 通过与环境交互生成数据，并通过逆向构建优化这些数据。这些高质量数据生成技术减少了对人工标注数据的依赖，促进代理快速适应新环境和任务。Ferret-UI 2 [102] 采用 Set-of-Mark(SoM) 视觉提示方法，为每个 UI 组件标注边界框和数字标签，辅助 GPT-40 识别。随后，GPT-40 生成与 UI 组件相关的问答任务数据，涵盖 UI 理解的多个方面，从而产出高质量训练数据。FedMobileAgent [283] 在用户日常手机使用过程中自动收集数据，并利用本地部署的视觉语言模型 (VLM) 标注用户操作，生成高质量数据集。此外，即使缺乏明确的真实标注，也可通过用户在 GUI 中的交互推断其意图，进而生成相应的 UI 标注 [284]。该方法为 GUI 数据的收集与标注开辟了新方向。

Collectively, these datasets represent significant strides in advancing phone automation and GUI-based agent research. They address various challenges, from language grounding and task feasibility to large-scale device control and cross-app navigation. By providing rich annotations and diverse scenarios, they enable the training and evaluation of more capable, robust, and generalizable agents, moving closer to the goal of intelligent and autonomous phone automation solutions.

总体而言，这些数据集代表了手机自动化和基于 GUI 的代理研究的重要进展。它们解决了从语言基础、任务可行性到大规模设备控制及跨应用导航的多种挑战。通过提供丰富的标注和多样化场景，这些数据集支持训练和评估更强大、稳健且具备泛化能力的代理，推动实现智能自主的手机自动化解决方案的目标。

5.2 Benchmarks

5.2 基准测试

The development of mobile GUI-based agents is not only reliant on the availability of diverse datasets but is also significantly influenced by the presence of robust benchmarks. These benchmarks offer standardized environment, tasks, and evaluation metrics, which are essential for consistently and reproducibly assessing the performance of agents. They enable researchers to compare different models and approaches under identical conditions, thus facilitating collaborative progress. In this subsection, we will review some of the notable benchmarks that have been introduced to evaluate phone GUI agents, highlighting their unique features and contributions. A summary of these benchmarks is provided in Table 7, which allows for a comparative understanding of their characteristics.

移动 GUI 代理的发展不仅依赖于多样化数据集的可用性，还受到健全基准测试的显著影响。这些基准提供了标准化的环境、任务和评估指标，对于持续且可复现地评估代理性能至关重要。它们使研究者能够在相同条件下比较不同模型和方法，促进协同进步。本小节将回顾一些用于评估手机 GUI 代理的著名基准，突出其独特特征和贡献。表 7 总结了这些基准，便于对其特性进行比较理解。

TABLE 7: Summary of benchmarks for phone GUI agents

表 7: 手机 GUI 代理基准测试汇总

Benchmark	Date	Tasks	Task Completion	Action Quality	Resource Efficiency	Task Understanding	Format Compliance	Completion Awareness	**Reward**	Eval Accuracy
MobileEnv [143] ☑	2023.05	74	✓	✘	✘	✘	✘	✘	✓	✘
AutoDroid [50] ⊠	2023.09	N/A	✓	✓	✘	✘	✘	✘	✘	✘
AndroidArena [144] ?	2024.02	N/A	✓	✓	✓	✓	✓	✓	✓	✘
LlamaTouch [145] ?	2024.04	496	✓	✓	✘	✓	✘	✓	✘	✓
B-MoCA [146] ?	2024.04	131	✓	✘	✓	✘	✘	✘	✘	✘
AndroidWorld [147] ?	2024.05	116	✓	✘	✘	✘	✘	✘	✓	✘
MobileAgent Bench [151]	2024.06	100	✓	✓	✓	✘	✘	✘	✓	✓
AUITestAgent [148] ?	2024.07	N/A	✓	✓	✘	✓	✓	✓	✓	✓
VisualAgent Bench [152] ?	2024.08	119	✓	✘	✓	✘	✘	✘	✘	✘
AgentStudio [149] ?	2024.10	205	✓	✓	✘	✓	✘	✓	✓	✓
AndroidLab [150] ?	2024.11	138	✓	✓	✓	✓	✘	✘	✓	✓
A3 [156] ?	2025.01	201	✓	✘	✘	✘	✘	✘	✓	✓
AutoEval [154]	2025.03	93	✓	✘	✘	✘	✘	✘	✓	✓
LearnGUI [155] ?	2025.04	2,353	✓	✘	✘	✘	✘	✘	✓	✘

基准测试	日期	任务	任务完成度	操作质量	资源效率	任务理解	格式合规性	完成意识	**Reward**	评估准确性
MobileEnv [143] ☑	2023.05	74	✓	✘	✘	✘	✘	✘	✓	✘
AutoDroid [50] ⊠	2023.09	不适用	✓	✓	✘	✘	✘	✘	✘	✘
AndroidArena [144] ?	2024.02	不适用	✓	✓	✓	✓	✓	✓	✓	✘
LlamaTouch [145] ?	2024.04	496	✓	✓	✘	✓	✘	✓	✘	✓
B-MoCA [146] ?	2024.04	131	✓	✘	✓	✘	✘	✘	✘	✘
AndroidWorld [147] ?	2024.05	116	✓	✘	✘	✘	✘	✘	✓	✘
MobileAgent Bench [151]	2024.06	100	✓	✓	✓	✘	✘	✘	✓	✓
AUITestAgent [148] ?	2024.07	不适用	✓	✓	✘	✓	✓	✓	✓	✓
VisualAgent Bench [152] ?	2024.08	119	✓	✘	✓	✘	✘	✘	✘	✘
AgentStudio [149] ?	2024.10	205	✓	✓	✘	✓	✘	✓	✓	✓
AndroidLab [150] ?	2024.11	138	✓	✓	✓	✓	✘	✘	✓	✓
A3 [156] ?	2025.01	201	✓	✘	✘	✘	✘	✘	✓	✓
AutoEval [154]	2025.03	93	✓	✘	✘	✘	✘	✘	✓	✓
LearnGUI [155] ?	2025.04	2,353	✓	✘	✘	✘	✘	✘	✓	✘

5.2.1 Evaluation Pipelines

5.2.1 评估流程

Early benchmarks in the field of phone GUI agents focused on creating controlled environment for training and evaluating these agents. MobileEnv [143], for example, introduced a universal platform for the training and evaluation of mobile interactions. It provided an isolated and controllable setting, with support for intermediate instructions and rewards. This emphasis on reliable evaluations and the ability to more naturally reflect real-world usage scenarios was a significant step forward.

早期的手机 GUI 代理领域基准测试侧重于创建受控环境以训练和评估这些代理。例如，MobileEnv [143] 引入了一个用于移动交互训练和评估的通用平台。它提供了一个隔离且可控的环境，支持中间指令和奖励。这种对可靠评估及更自然反映现实使用场景的重视，是一大进步。

To address the challenges presented by the complexities of modern operating systems and their vast action spaces, AndroidArena [144] was developed. This benchmark was designed to evaluate large language model (LLM) agents within a complex Android environment. It introduced scalable and semi-automated methods for benchmark construction, with a particular focus on cross-application collaboration and user constraints such as security concerns.

为应对现代操作系统复杂性及其庞大动作空间带来的挑战，开发了 AndroidArena [144]。该基准旨在评估复杂 Android 环境中的大型语言模型 (LLM) 代理。它引入了可扩展且半自动化的基准构建方法，特别关注跨应用协作和用户约束，如安全性问题。

Current research primarily focuses on the overall task success rate and often overlooks the evaluation of core capabilities such as GUI grounding of agents in real-world scenarios. AgentStudio [149] provides a comprehensive platform that spans the entire development cycle, from environment setup and data collection to agent evaluation and visualization. AgentStudio also introduces three benchmark datasets: GroundUI, IDMBench, and CriticBench. These datasets are designed to evaluate agents' capabilities in GUI grounding, learning from videos, and success detection, respectively. Additionally, it introduces a benchmark suite comprising 205 real-world tasks to comprehensively evaluate agents' practical capabilities from multiple perspectives.

当前研究主要关注整体任务成功率，常忽视对代理在现实场景中 GUI 定位等核心能力的评估。AgentStudio [149] 提供了一个涵盖整个开发周期的综合平台，从环境搭建、数据收集到代理评估和可视化。AgentStudio 还引入了三个基准数据集:GroundUI、IDMBench 和 CriticBench，分别用于评估代理的 GUI 定位、视频学习和成功检测能力。此外，它还推出了包含 205 个真实任务的基准套件，从多角度全面评估代理的实际能力。

Recognizing the limitations in scalability and faithfulness of existing evaluation approaches, LlamaTouch [145] presented a novel testbed. This testbed enabled on-device mobile UI task execution and provided a means for faithful and scalable task evaluation. It introduced fine-grained UI component annotation and a multi-level application state matching algorithm. These features allowed for the accurate detection of critical information in each screen, enhancing the evaluation's accuracy and adaptability to dynamic UI changes.

鉴于现有评估方法在可扩展性和真实性方面的局限，LlamaTouch [145] 提出了一个新型测试平台。该平台支持设备端移动 UI 任务执行，并提供了真实且可扩展的任务评估手段。它引入了细粒度 UI 组件标注和多层次应用状态匹配算法，能够准确检测每个界面中的关键信息，提升评估的准确性和对动态 UI 变化的适应性。

B-MoCA [146] expanded the focus of benchmarking to include mobile device control agents across diverse configurations. By incorporating a randomization feature that could change device configurations such as UI layouts and language settings, B-MoCA was able to more effectively assess agents' generalization performance. It provided a realistic benchmark with 131 practical tasks, highlighting the need for agents to handle a wide range of real-world scenarios.

B-MoCA [146] 扩展了基准测试的范围，涵盖多种配置下的移动设备控制代理。通过引入随机化功能，可改变设备配置如 UI 布局和语言设置，B-MoCA 更有效地评估代理的泛化性能。它提供了包含 131 个实际任务的真实基准，凸显了代理应对多样现实场景的需求。

To provide a dynamic and reproducible environment for autonomous agents, AndroidWorld [147] introduced

an Android environment with 116 programmatic tasks across 20 real-world apps. This benchmark emphasized the importance of ground-truth rewards and the ability to dynamically construct tasks that were parameterized and expressed in natural language. This enabled testing on a much larger and more realistic suite of tasks.

为提供动态且可复现的自主代理环境，AndroidWorld [147] 引入了一个包含 20 个真实应用中 116 个程序化任务的 Android 环境。该基准强调了真实奖励的重要性及动态构建参数化且以自然语言表达任务的能力，从而支持在更大规模且更真实的任务集上进行测试。

For the specific evaluation of mobile LLM agents, Mo-bileAgentBench [151] proposed an efficient and user-friendly benchmark. It addressed challenges in scalability and usability by offering 100 tasks across 10 open-source apps. The benchmark also simplified the extension process for developers and ensured that it was fully autonomous and reliable.

针对移动大型语言模型代理的专门评估，Mo-bileAgentBench [151] 提出了一个高效且用户友好的基准。它通过提供 10 个开源应用中的 100 个任务，解决了可扩展性和易用性挑战。该基准还简化了开发者的扩展流程，确保完全自主且可靠。

In the domain of GUI function testing, AUITestA-gent [148] introduced the first automatic, natural language-driven GUI testing tool for mobile apps. By decoupling interaction and verification into separate modules and employing a multi-dimensional data extraction strategy, it enhanced the automation and accuracy of GUI testing. The practical usability of this tool was demonstrated in real-world deployments.

在 GUI 功能测试领域，AUITestAgent [148] 推出了首个自动化、基于自然语言驱动的移动应用 GUI 测试工具。通过将交互与验证模块解耦，并采用多维数据提取策略，提升了 GUI 测试的自动化和准确性。该工具的实用性已在真实部署中得到验证。

AndroidLab [150] presented a systematic Android agent framework. This framework included an operation environment with different modalities and a reproducible benchmark. Supporting both LLMs and large multimodal models (LMMs), it provided a unified platform for training and evaluating agents. Additionally, it came with an Android Instruction dataset that significantly improved the performance of open-source models.

AndroidLab [150] 提出了一个系统化的 Android 代理框架。该框架包含多模态操作环境和可复现的基准，支持大型语言模型 (LLM) 和大型多模态模型 (LMM)，提供了统一的训练与评估平台。此外，配备的 Android 指令数据集显著提升了开源模型的性能。

LearnGUI [155] offers a novel approach by introducing the first comprehensive benchmark specifically designed for demonstration-based learning in mobile GUI agents. Rather than pursuing universal generalization through larger datasets, it focuses on improving agent performance in unseen scenarios through human demonstrations. The benchmark comprises 2,252 offline tasks and 101 online tasks with high-quality human demonstrations.

LearnGUI [155] 提供了一种新颖方法，推出了首个专为基于示范学习的移动 GUI 代理设计的综合基准。它不追求通过更大数据集实现普适泛化，而是通过人类示范提升代理在未见场景中的表现。该基准包含 2252 个离线任务和 101 个带高质量人类示范的在线任务。

Finally, to evaluate the practical performance of mobile GUI agents in complex real-world environments, VisualA-gentBench [152] constructs a series of cross-domain tasks. This benchmark examines the agents' abili-

ties in dynamic interaction and decision-making and provides abundant training trajectory data to support further performance improvement via behavior cloning. A3 (Android Agent Arena) [156] integrates 201 tasks from 21 widely-used third-party applications, covering common real-world user scenarios. It supports an extended action space compatible with any dataset annotation style. Additionally, the use of business-level LLMs automates task evaluation, reducing the need for manual assessment and enhancing scalability.

最后，为评估移动 GUI 代理在复杂真实环境中的实际表现，VisualAgentBench [152] 构建了一系列跨领域任务。该基准考察代理的动态交互与决策能力，并提供丰富的训练轨迹数据，支持通过行为克隆进一步提升性能。A3(Android Agent Arena)[156] 集成了来自 21 个广泛使用第三方应用的 201 个任务，涵盖常见真实用户场景。它支持兼容任何数据集标注风格的扩展动作空间。此外，采用业务级大型语言模型自动化任务评估，减少人工评估需求，提升可扩展性。

AutoEval [154] addresses the practicality and scalability challenges in mobile agent evaluation by introducing a framework that requires no manual effort to define task reward signals or implement evaluation codes. It employs a Structured Substate Representation to describe UI state changes during agent execution and utilizes a Judge System that can autonomously evaluate agent performance with over 94% accuracy compared to human verification.

AutoEval [154] 通过引入一个无需手动定义任务奖励信号或实现评估代码的框架，解决了移动代理评估中的实用性和可扩展性挑战。它采用结构化子状态表示 (Structured Substate Representation) 来描述代理执行过程中的 UI 状态变化，并利用一个评判系统 (Judge System)，该系统能够以超过 94% 的准确率自主评估代理性能，相较于人工验证。

Collectively, these benchmarks have made substantial contributions to the advancement of phone GUI agents. They have achieved this by providing diverse environment, tasks, and evaluation methodologies. They have addressed various challenges, including scalability, reproducibility, generalization across configurations, and the integration of advanced models like LLMs and LMMs. By facilitating rigorous testing and comparison, they have played a crucial role in driving the development of more capable and robust phone GUI agents.

这些基准测试共同为手机 GUI 代理的发展做出了重要贡献。它们通过提供多样化的环境、任务和评估方法，解决了包括可扩展性、可复现性、跨配置泛化以及集成大型语言模型 (LLMs) 和大型多模态模型 (LMMs) 等多种挑战。通过促进严格的测试和比较，这些基准在推动更强大、更稳健的手机 GUI 代理开发中发挥了关键作用。

5.2.2 Evaluation Metrics

5.2.2 评估指标

Evaluation metrics are crucial for measuring the performance of phone GUI agents, providing quantitative indicators of their effectiveness, efficiency, and reliability. This section categorizes and explains the various metrics used across different benchmarks based on their primary functions.

评估指标对于衡量手机 GUI 代理的性能至关重要，提供了其有效性、效率和可靠性的量化指标。本节根据主要功能对不同基准中使用的各类指标进行分类和说明。

Task Completion Metrics. Task Completion Metrics assess how effectively an agent finishes assigned tasks.

Task Completion Rate indicates the proportion of successfully finished tasks, with AndroidWorld [147] exemplifying its use for real-device assessments. Sub-Goal Success Rate further refines this by examining each sub-goal within a larger task, as employed by AndroidLab [150], making it particularly relevant for complex tasks that require segmentation. End-to-end Task Completion Rate, used by LlamaTouch [145], offers a holistic measure of whether an agent can see an entire multi-step task through to completion without interruption.

任务完成指标。任务完成指标评估代理完成分配任务的效果。任务完成率表示成功完成任务的比例，AndroidWorld [147] 以真实设备评估为例。子目标成功率进一步细化，通过检查大型任务中的每个子目标，如 AndroidLab [150] 所用，特别适用于需要分段的复杂任务。端到端任务完成率由 LlamaTouch [145] 采用，提供了代理是否能完整无间断地完成多步骤任务的整体衡量。

Action Execution Quality Metrics. These metrics evaluate the agent's precision and correctness when performing specific actions. Action Accuracy, adopted by AUITestAgent [148] and AutoDroid [66], compares each executed action to the expected one. Correct Step measures the fraction of accurate steps in an action sequence, whereas Correct Trace quantifies the alignment of the entire action trajectory with the ground truth. Operation Logic checks if the agent follows logical procedures to meet task objectives, as AndroidArena [144] demonstrates. Reasoning Accuracy, highlighted in AUITestA-gent [148], gauges how well the agent logically interprets and responds to task requirements.

动作执行质量指标。这些指标评估代理执行特定动作时的精确性和正确性。动作准确率由 AUITestA-gent [148] 和 AutoDroid [66] 采用，比较每个执行动作与预期动作的匹配度。正确步骤衡量动作序列中准确步骤的比例，而正确轨迹量化整个动作轨迹与真实轨迹的一致性。操作逻辑检查代理是否遵循逻辑流程以达成任务目标，如 AndroidArena [144] 所示。推理准确率，在 AUITestAgent [148] 中强调，衡量代理对任务需求的逻辑理解和响应能力。

Resource Utilization and Efficiency Metrics. These indicators measure how efficiently an agent handles system resources and minimizes redundant operations. Resource Consumption, tracked by AUITestAgent [148] via Completion Tokens and Prompt Tokens, reveals how much computational cost is incurred. Step Efficiency, applied by AUITestAgent and MobileAgentBench [151], compares actual steps to an optimal lower bound, while Reversed Redundancy Ratio, used by AndroidArena [144] and AndroidLab [150], evaluates unnecessary detours in the action path.

资源利用与效率指标。这些指标衡量代理处理系统资源的效率及减少冗余操作的能力。资源消耗由 AUITestAgent [148] 通过完成令牌和提示令牌跟踪，反映计算成本。步骤效率由 AUITestAgent 和 MobileAgentBench [151] 应用，比较实际步骤数与最优下界，而逆冗余率由 AndroidArena [144] 和 AndroidLab [150] 使用，评估动作路径中的不必要绕行。

Task Understanding and Reasoning Metrics. These metrics concentrate on the agent's comprehension and analytical skills. Oracle Accuracy and Point Accuracy, used by AUITestA-gent [148], assess how well the agent interprets task instructions and verification points. Reasoning Accuracy indicates the correctness of the agent's logical deductions during execution, and Nuggets Mining, employed by AndroidArena [144], measures the ability to extract key contextual information from the UI environment.

任务理解与推理指标。这些指标关注代理的理解和分析能力。Oracle 准确率和点准确率由 AUITestAgent [148] 使用，评估代理对任务指令和验证点的理解程度。推理准确率指代理执行过程中逻辑推断的正确性，Nuggets 挖掘由 AndroidArena [144] 采用，衡量从 UI 环境中提取关键信息的能力。

Format and Compliance Metrics. These metrics verify whether the agent operates within expected format constraints. Invalid Format and Invalid Action, for example, are tracked in AndroidArena [144] to confirm that an agent's outputs adhere to predefined structures and remain within permissible action ranges.

格式与合规性指标。这些指标验证代理是否在预期格式约束内操作。例如，AndroidArena [144] 跟踪无效格式和无效动作，确保代理输出符合预定义结构并保持在允许的动作范围内。

Completion Awareness and Reflection Metrics. Such metrics evaluate the agent's recognition of task boundaries and its capacity to learn from prior steps. Awareness of Completion, explored in AndroidArena [144], ensures the agent terminates at the correct time. Reflexion@K measures adaptive learning by examining how effectively the agent refines its performance over multiple iterations.

完成意识与反思指标。这类指标评估代理对任务边界的识别能力及从先前步骤中学习的能力。完成意识在 AndroidArena [144] 中探讨，确保代理在正确时机终止。反思 @K 通过考察代理在多次迭代中如何有效改进表现，衡量其适应性学习能力。

Evaluation Accuracy and Reliability Metrics. These indicators measure the consistency and reliability of the evaluation process. Accuracy, as used in LlamaTouch [145], validates alignment between the evaluation approach and manual verification, ensuring confidence in performance comparisons across agents.

评估准确性与可靠性指标。这些指标衡量评估过程的一致性和可靠性。准确率由 LlamaTouch [145] 使用，验证评估方法与人工验证的一致性，确保不同代理性能比较的可信度。

Reward and Overall Performance Metrics. These metrics combine various performance facets into aggregated scores. Task Reward, employed by AndroidArena [144], provides a single effectiveness measure encompassing several factors. Average Reward, used in MobileEnv [143], further reflects consistent performance across multiple tasks, indicating the agent's stability and reliability.

奖励与整体性能指标。这些指标将多方面性能综合为聚合分数。任务奖励由 AndroidArena [144] 采用，提供涵盖多个因素的单一有效性衡量。平均奖励由 MobileEnv [143] 使用，反映多任务中的稳定表现，指示代理的稳定性和可靠性。

These evaluation metrics together provide a comprehensive framework for assessing various dimensions of phone GUI agents. They cover aspects such as effectiveness, efficiency, reliability, and the ability to adapt and learn. By using these metrics, benchmarks can objectively compare the performance of different agents and systematically measure improvements. This enables researchers to identify strengths and weaknesses in different agent designs and make informed decisions about future development directions.

这些评估指标共同构建了一个全面的框架，用于评估手机 GUI 代理的多维度表现。涵盖了有效性、效率、可靠性以及适应和学习能力等方面。通过使用这些指标，基准测试能够客观比较不同代理的性能，系统性地衡量改进效果，帮助研究者识别不同代理设计的优劣，并为未来发展方向提供科学依据。

6 CHALLENGES AND FUTURE DIRECTIONS

6 挑战与未来方向

Integrating LLMs into phone automation has propelled significant advancements but also introduced numerous challenges. Overcoming these challenges is essential for fully unlocking the potential of intelligent phone GUI agents. This section outlines key issues and possible directions for future work, encompassing dataset development, scaling fine-tuning, lightweight on-device deployment, user-centric adaptation, improving model capabilities, standardizing benchmarks, and ensuring reliability and security.

将大型语言模型 (LLMs) 集成到手机自动化中推动了显著进展，但也带来了诸多挑战。克服这些挑战对于充分释放智能手机图形用户界面 (GUI) 代理的潜力至关重要。本节概述了关键问题及未来工作可能的方向，涵盖数据集开发、微调扩展、轻量级设备端部署、以用户为中心的适应、提升模型能力、标准化基准测试以及确保可靠性和安全性。

Dataset Development and Fine-Tuning Scalability. The performance of LLMs in phone automation heavily depends on datasets that capture diverse, real-world scenarios. Existing datasets often lack the breadth needed for comprehensive coverage. Future efforts should focus on developing large-scale, annotated datasets covering a wide range of applications, user behaviors, languages, and device types [137], [138]. Incorporating multimodal inputs—e.g., screenshots, UI trees, and natural language instructions—can help models better understand complex user interfaces. In addition, VideoGUI [285] proposes using instructional videos to demonstrate complex visual tasks to models, helping them to learn how to transition from an initial state to a target state. Video datasets are expected to evolve into a new form for future GUI datasets. However, scaling fine-tuning to achieve robust out-of-domain performance remains a challenge. As shown by AndroidControl [140], obtaining reliable results for high-level tasks outside the training domain may require one to two orders of magnitude more data than currently feasible. Fine-tuning alone may not suffice. Future directions should explore hybrid training methodologies, unsupervised learning, transfer learning, and auxiliary tasks to improve generalization without demanding prohibitively large datasets.

数据集开发与微调扩展性。LLMs 在手机自动化中的表现高度依赖于能够涵盖多样化真实场景的数据集。现有数据集往往缺乏全面覆盖所需的广度。未来工作应聚焦于开发涵盖广泛应用、用户行为、语言和设备类型的大规模标注数据集 [137]，[138]。引入多模态输入——如截图、UI 树和自然语言指令——有助于模型更好地理解复杂用户界面。此外，VideoGUI[285] 提出利用教学视频向模型展示复杂视觉任务，帮助其学习如何从初始状态过渡到目标状态。视频数据集有望成为未来 GUI 数据集的新形式。然而，实现稳健的域外性能的微调扩展仍是挑战。如 AndroidControl[140] 所示，获得训练域外高层任务的可靠结果可能需要比当前可行数据量多一到两个数量级。仅靠微调可能不足。未来方向应探索混合训练方法、无监督学习、迁移学习及辅助任务，以提升泛化能力，同时避免对庞大数据集的过度依赖。

Lightweight and Efficient On-Device Deployment. Deploying LLMs on mobile devices confronts substantial computational and memory constraints. Current hardware often struggles to support large models with minimal latency and power consumption. Approaches such as model pruning, quantization, and efficient transformer architectures can address these constraints [111]. Recent innovations demonstrate promising progress. Octopus v2 [286] shows that a 2-billion parameter on-device model can outpace GPT-4 in accuracy and latency, while Lightweight Neural App Control [81] achieves substantial speed and accuracy improvements by distributing tasks efficiently. AppVLM [113], a lightweight vision-language model, matches GPT-40 in online task completion success rate while being up to ten times faster, making it practical for real-world deployment. Moreover, specialized hardware accelerators and edge computing solutions can further reduce dependency on the cloud, enhance privacy, and improve responsiveness [52]. Consider leveraging the powerful code generation capabilities of small language models (SLMs) to transform GUI task automation into a code generation problem. This approach fully utilizes the strengths of SLMs, significantly enhancing the efficiency and performance of GUI agents on mobile devices [287], [288].

轻量高效的设备端部署。在移动设备上部署 LLMs 面临显著的计算和内存限制。当前硬件常难以支持大型模型同时实现低延迟和低功耗。模型剪枝、量化及高效 Transformer 架构等方法可缓解这些限制 [111]。近期创新展现出良好进展。Octopus v2[286] 表明，拥有 20 亿参数的设备端模型在准确率和延迟上均优于 GPT-4；而 Lightweight Neural App Control[81] 通过高效任务分配实现了显著的速度和准确率提升。轻量级视觉语言模型 AppVLM[113] 在线任务完成成功率匹配 GPT-4，速度却快十倍，具备实际部署潜力。此外，专用硬件加速器和边缘计算方案可进一步减少对云端的依赖，增强隐私保护并提升响应速度 [52]。可考虑利用小型语言模型 (SLMs) 强大的代码生成能力，将 GUI 任务自动化转化为代码生成问题，充分发挥 SLMs 优势，显著提升移动设备上 GUI 代理的效率和性能 [287]，[288]。

User-Centric Adaptation: Interaction and Personalization. Current agents often rely on extensive human intervention to correct errors or guide task execution, undermining seamless user experiences. Enhancing the agent's ability to understand user intent and reducing manual adjustments is crucial. Future research should improve natural language understanding, incorporate voice commands and gestures, and enable agents to learn continuously from user feedback [51], [52], [61], [289]. Personalization is equally important. One-size-fits-all solutions are insufficient given users' diverse preferences and usage patterns. Agents should quickly adapt to new tasks and user-specific contexts without costly retraining. Integrating manual teaching, zero-shot learning, and few-shot learning can help agents generalize from minimal user input [61], [79], [85], [290], making them more flexible and universally applicable. For example, AdaptAgent [225] is capable of adapting to entirely new domains with as few as two human demonstrations. This not only proves the efficiency of limited human input, but also paves a new path for the development of multi-modal agents with broad adaptability. Similarly, LearnAct [155] demonstrates the power of human demonstrations in mobile GUI agents, using a multi-agent framework to automatically extract knowledge from demonstrations to enhance task completion. It establishes demonstration-based learning as a promising direction for creating more personalized and adaptive mobile agents.

以用户为中心的适应: 交互与个性化。当前代理常依赖大量人工干预以纠正错误或指导任务执行, 影响用户体验的流畅性。提升代理理解用户意图的能力并减少手动调整至关重要。未来研究应加强自然语言理解，融合语音命令和手势操作，使代理能持续从用户反馈中学习 [51]，[52]，[61]，[289]。个性化同样重要。鉴于用户偏好和使用习惯的多样性，一刀切的解决方案难以满足需求。代理应能快速适应新任务和用户特定情境，且无需昂贵的再训练。结合手动教学、零样本学习和少样本学习，有助于代理从极少的用户输入中泛化 [61]，[79]，[85]，[290]，使其更灵活且适用范围更广。例如，AdaptAgent[225] 能够通过仅两次人工示范适应全新领域, 这不仅证明了有限人工输入的高效性, 也为开发具备广泛适应性的多模态代理开辟了新路径。同样，LearnAct[155] 展示了人工示范在移动 GUI 代理中的作用，利用多代理框架自动提取示范知识以提升任务完成度，确立了基于示范的学习作为打造更个性化和适应性强的移动代理的有前景方向。

Advancing Model Capabilities: Grounding, Reasoning, and Beyond. Accurately grounding language instructions in specific UI elements is a major hurdle. Although LLMs excel at language understanding, mapping instructions to precise UI interactions requires improved multimodal grounding. Future work should integrate advanced vision models, large-scale annotations, and more effective fusion techniques [80], [95], [101], [105]. Beyond grounding, improving reasoning, long-horizon planning, and adaptability in complex scenarios remains essential. Agents must handle intricate workflows, interpret ambiguous instructions, and dynamically adjust strategies as contexts evolve. Achieving these goals will likely involve new architectures, memory mechanisms, and inference algorithms that extend beyond current LLM capabilities. Standardizing Evaluation Benchmarks. Objective and reproducible benchmarks are imperative for comparing model performance. Existing benchmarks often target narrow tasks or limited domains, complicating comprehensive evaluations. Unified benchmarks covering diverse tasks, app types, and interaction modalities would foster fair comparisons and encourage more versatile and robust solutions [109], [137], [150], [151]. These benchmarks should provide standardized metrics, scenarios, and evaluation protocols, enabling researchers to identify strengths, weaknesses, and paths for improvement with greater clarity.

提升模型能力: 定位、推理及更广领域。准确地将语言指令定位到具体的 UI 元素是一个主要难题。尽管大型语言模型 (LLMs) 在语言理解方面表现出色，但将指令映射到精确的 UI 交互仍需改进多模态定位。未来工作应整合先进的视觉模型、大规模标注以及更有效的融合技术 [80]，[95]，[101]，[105]。除了定位，提升推理能力、长远规划和复杂场景中的适应性依然至关重要。智能体必须处理复杂的工作流程，解读模糊指令，并随着环境变化动态调整策略。实现这些目标可能需要新的架构、记忆机制和推理算法，超越当前 LLM 的能力。标准化评估基准。客观且可复现的基准对于比较模型性能至关重要。现有基准往往针对狭窄任务或有限领域，难以进行全面评估。涵盖多样任务、应用类型和交互方式的统一基准将促进公平比较，推动更通用且稳健的解决方案 [109]，[137]，[150]，[151]。这些基准应提供标准化的指标、场景和评估协议，使研究者能够更清晰地识别优势、劣势及改进路径。

Ensuring Reliability and Security. As agents gain access to sensitive data and perform critical tasks, reliability and security are paramount. Current systems may be susceptible to adversarial attacks, data breaches, and unintended actions [291]. At the same time, LLM agents are also susceptible to backdoor attacks [292], [293]. Recent research like AEIA-MN [294] has demonstrated that multimodal LLM-powered mobile agents are highly vulnerable to Active Environmental Injection Attacks, where attackers manipulate environmental elements (e.g., notifications) to mislead agents, achieving attack success rates up to 93% in benchmark tests. Robust security protocols, error-handling techniques, and privacy-preserving methods are needed to protect user information and maintain user trust [71], [116]. Employing techniques such as data localization, encrypted communication, and anonymization can effectively protect user privacy while collecting data [283]. FedMABench [153] addresses the

challenges of distributed training using federated learning, providing a comprehensive benchmark for evaluating mobile agents across heterogeneous environments. Continuous monitoring and validation processes can detect vulnerabilities and mitigate risks in real-time [61]. Ensuring that agents behave predictably, respect user privacy, and maintain consistent performance under challenging conditions will be crucial for widespread adoption and long-term sustainability.

确保可靠性与安全性。随着智能体获得访问敏感数据和执行关键任务的能力，可靠性和安全性变得尤为重要。当前系统可能易受对抗攻击、数据泄露和非预期操作的影响 [291]。同时，LLM 智能体也易受后门攻击 [292]，[293]。近期研究如 AEIA-MN[294] 表明，多模态 LLM 驱动的移动智能体高度易受主动环境注入攻击 (Active Environmental Injection Attacks)，攻击者通过操控环境元素 (如通知) 误导智能体，在基准测试中攻击成功率高达 93%。需要健全的安全协议、错误处理技术和隐私保护方法以保障用户信息安全并维护用户信任 [71]，[116]。采用数据本地化、加密通信和匿名化等技术可在数据收集过程中有效保护用户隐私 [283]。FedMABench[153] 针对联邦学习中的分布式训练挑战，提供了评估异构环境下移动智能体的综合基准。持续的监控与验证流程能够实时检测漏洞并降低风险 [61]。确保智能体行为可预测、尊重用户隐私并在严苛条件下保持稳定性能，对于广泛应用和长期可持续发展至关重要。

Addressing these challenges involves concerted efforts in data collection, model training strategies, hardware optimization, user-centric adaptation, improved grounding and reasoning, standardized benchmarks, and strong security measures. By advancing these areas, the next generation of LLM-powered phone GUI agents can become more efficient, trustworthy, and capable, ultimately delivering seamless, personalized, and secure experiences for users in dynamic mobile environment.

应对这些挑战需要在数据收集、模型训练策略、硬件优化、以用户为中心的适应性、改进的定位与推理、标准化基准以及强有力的安全措施等方面协同努力。通过推进这些领域，下一代基于 LLM 的手机 GUI 智能体将更加高效、可信且功能强大，最终为用户在动态移动环境中提供无缝、个性化且安全的体验。

7 CONCLUSION

7 结论

In this paper, we have presented a comprehensive survey of recent developments in LLM-driven phone automation technologies, illustrating how large language models can catalyze a paradigm shift from static script-based approaches to dynamic, intelligent systems capable of perceiving, reasoning about, and operating on mobile GUIs. We examined a variety of frameworks, including single-agent architectures, multi-agent collaborations, and plan-then-act pipelines, demonstrating how each approach addresses specific challenges in task complexity, adaptability, and scalability. In parallel, we analyzed both prompt engineering and training-based techniques (such as supervised fine-tuning and reinforcement learning), underscoring their roles in bridging user intent and device action.

本文全面综述了基于大型语言模型驱动的手机自动化技术的最新进展，展示了大型语言模型如何推动从静态脚本方法向能够感知、推理并操作移动 GUI 的动态智能系统的范式转变。我们考察了多种框架，包括单智能体架构、多智能体协作以及先规划后执行的流程，展示了各方法如何应对任务复杂性、适应性和可扩展性的具体挑战。同时，我们分析了提示工程和基于训练的技术 (如监督微调和强化学习)，强调了它们在连接用户意图与设备操作中的作用。

Beyond clarifying these technical foundations, we also spotlighted emerging research directions and provided a critical appraisal of persistent obstacles. These include ensuring robust dataset coverage, optimizing LLM deployments under resource constraints, meeting real-world demand for user-centric personalization, and maintaining security and reliability in sensitive applications. We further emphasized the need for standardized benchmarks, proposing consistent metrics and evaluation protocols to fairly compare and advance competing designs.

除了阐明这些技术基础，我们还聚焦了新兴研究方向并对持续存在的难题进行了批判性评估。这些难题包括确保数据集的覆盖广度、在资源受限条件下优化 LLM 部署、满足现实世界中以用户为中心的个性化需求，以及在敏感应用中维护安全性和可靠性。我们进一步强调了标准化基准的必要性，提出一致的指标和评估协议，以公平比较和推动竞争设计的发展。

Looking ahead, ongoing refinements in model architectures, on-device inference strategies, and multimodal data integration point to an exciting expansion of what LLM-based phone GUI agents can achieve. We anticipate that future endeavors will see the convergence of broader AI paradigms-such as embodied AI and AGI-into phone automation, thereby enabling agents to handle increasingly complex tasks with minimal human oversight. Overall, this survey not only unifies existing strands of research but also offers a roadmap for leveraging the full potential of large language models in phone GUI automation, guiding researchers toward robust, user-friendly, and secure solutions that can adapt to the evolving needs of mobile ecosystems.

展望未来，模型架构、设备端推理策略和多模态数据整合的持续改进预示着基于 LLM 的手机 GUI 智能体能力的激动人心的扩展。我们预期未来的努力将见证更广泛的人工智能范式——如具身 AI(embodied AI) 和通用人工智能 (AGI)——与手机自动化的融合，从而使智能体能够以最少的人类监督处理日益复杂的任务。总体而言，本综述不仅整合了现有研究脉络，还为充分发挥大型语言模型在手机 GUI 自动化中的潜力提供了路线图，引导研究者开发稳健、用户友好且安全的解决方案，以适应移动生态系统不断演变的需求。

REFERENCES

参考文献

[1] T. Azim and I. Neamtiu, ”Targeted and depth-first exploration for systematic testing of android apps,” in Proceedings of the 2013 ACM SIGPLAN international conference on Object oriented programming systems languages & applications, 2013, pp. 641-660. 1, 3, 4

T. Azim 和 I. Neamtiu，“针对安卓应用的目标导向和深度优先探索系统测试”，载于 2013 年 ACM SIGPLAN 面向对象程序设计系统语言与应用国际会议论文集，2013 年，第 641-660 页。1，3，4

[2] M. Pan, A. Huang, G. Wang, T. Zhang, and X. Li, ”Reinforcement learning based curiosity-driven testing

of android applications," in Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis, 2020, pp. 153-164. 1, 4

M. Pan, A. Huang, G. Wang, T. Zhang, 和 X. Li, "基于强化学习的好奇心驱动安卓应用测试," 载于第 29 届 ACM SIGSOFT 国际软件测试与分析研讨会论文集, 2020, 页 153-164. 1, 4

[3] Y. Koroglu, A. Sen, O. Muslu, Y. Mete, C. Ulker, T. Tanriverdi, and Y. Donmez, "Qbe: Qlearning-based exploration of android applications," in 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST). IEEE, 2018, pp. 105-115. 1,4

Y. Koroglu, A. Sen, O. Muslu, Y. Mete, C. Ulker, T. Tanriverdi, 和 Y. Donmez, "Qbe: 基于 Q 学习的安卓应用探索," 载于 2018 年第 11 届 IEEE 软件测试、验证与确认国际会议 (ICST) 论文集. IEEE, 2018, 页 105-115. 1,4

[4] Y. Li, Z. Yang, Y. Guo, and X. Chen, "Humanoid: A deep learning-based approach to automated black-box android app testing," in 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2019, pp. 1070-1073. 1, 4

Y. Li, Z. Yang, Y. Guo, 和 X. Chen, "Humanoid: 一种基于深度学习的自动黑盒安卓应用测试方法," 载于 2019 年第 34 届 IEEE/ACM 自动化软件工程国际会议 (ASE) 论文集. IEEE, 2019, 页 1070-1073. 1, 4

[5] C. Degott, N. P. Borges Jr, and A. Zeller, "Learning user interface element interactions," in Proceedings of the 28th ACM SIGSOFT international symposium on software testing and analysis, 2019, pp. 296-306. 1, 4

C. Degott, N. P. Borges Jr, 和 A. Zeller, "学习用户界面元素交互," 载于第 28 届 ACM SIGSOFT 国际软件测试与分析研讨会论文集, 2019, 页 296-306. 1, 4

[6] Y. L. Arnatovich and L. Wang, "A systematic literature review of automated techniques for functional gui testing of mobile applications," arXiv preprint arXiv:1812.11470, 2018. 1, 2

Y. L. Arnatovich 和 L. Wang, "移动应用功能性 GUI 测试自动化技术的系统文献综述," arXiv 预印本 arXiv:1812.11470, 2018. 1, 2

[7] P. S. Deshmukh, S. S. Date, P. N. Mahalle, and J. Barot, "Automated gui testing for enhancing user experience (ux): A survey of the state of the art," in International Conference on ICT for Sustainable Development. Springer, 2023, pp. 619-628. 1, 2

P. S. Deshmukh, S. S. Date, P. N. Mahalle, 和 J. Barot, "提升用户体验 (UX) 的自动化 GUI 测试: 现状综述," 载于国际 ICT 可持续发展会议. Springer, 2023, 页 619-628. 1, 2

[8] M. Nass, "On overcoming challenges with gui-based test automation," Ph.D. dissertation, Blekinge Tekniska Högskola, 2024. 1, 2

M. Nass, "克服基于 GUI 的测试自动化挑战," 博士论文, Blekinge Tekniska Högskola, 2024. 1, 2

[9] M. Nass, E. Alégroth, and R. Feldt, ”Why many challenges with gui test automation (will) remain,” Information and Software Technology, vol. 138, p. 106625, 2021. 1, 2

M. Nass, E. Alégroth, 和 R. Feldt, “为何 GUI 测试自动化的诸多挑战依然存在 (或将持续存在),”信息与软件技术, 卷 138, 页 106625, 2021. 1, 2

[10] P. Tramontana, D. Amalfitano, N. Amatucci, and A. R. Fasolino, ”Automated functional testing of mobile applications: a systematic mapping study,” Software Quality Journal, vol. 27, pp. 149-201, 2019. 1, 2

P. Tramontana, D. Amalfitano, N. Amatucci, 和 A. R. Fasolino, “移动应用自动功能测试: 系统映射研究,”软件质量期刊, 卷 27, 页 149-201, 2019. 1, 2

[11] Y. Li, H. Wen, W. Wang, X. Li, Y. Yuan, G. Liu, J. Liu, W. Xu, X. Wang, Y. Sun et al., ”Personal llm agents: Insights and survey about the capability, efficiency and security,” arXiv preprint arXiv:2401.05459, 2024. 1

Y. Li, H. Wen, W. Wang, X. Li, Y. Yuan, G. Liu, J. Liu, W. Xu, X. Wang, Y. Sun 等, “个人大型语言模型代理: 能力、效率与安全性的洞察与综述,”arXiv 预印本 arXiv:2401.05459, 2024. 1

[12] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang, ”Large language model based multi-agents: A survey of progress and challenges,” arXiv preprint arXiv:2402.01680, 2024. 1

T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, 和 X. Zhang, “基于大型语言模型的多代理系统: 进展与挑战综述,”arXiv 预印本 arXiv:2402.01680, 2024. 1

[13] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin et al., ”A survey on large language model based autonomous agents,” Frontiers of Computer Science, vol. 18, no. 6, p. 186345, 2024. 1

L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin 等, “基于大型语言模型的自主代理综述,”计算机科学前沿, 卷 18, 期 6, 页 186345, 2024. 1

[14] H. Jin, L. Huang, H. Cai, J. Yan, B. Li, and H. Chen, ”From llms to llm-based agents for software engineering: A survey of current, challenges and future,” arXiv preprint arXiv:2408.02479, 2024. 1

H. Jin, L. Huang, H. Cai, J. Yan, B. Li, 和 H. Chen, “从大型语言模型到基于大型语言模型的软件工程代理: 现状、挑战与未来综述,”arXiv 预印本 arXiv:2408.02479, 2024. 1

[15] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg et al., ”Sparks of artificial general intelligence: Early experiments with gpt-4,” arXiv preprint arXiv:2303.12712, 2023. 1

S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg 等，“通用人工智能的火花:GPT-4 的早期实验”，arXiv 预印本 arXiv:2303.12712，2023 年。1

[16] X. Huang, W. Liu, X. Chen, X. Wang, H. Wang, D. Lian, Y. Wang, R. Tang, and E. Chen, ”Understanding the planning of llm agents: A survey,” arXiv preprint arXiv:2402.02716, 2024. 1

X. Huang, W. Liu, X. Chen, X. Wang, H. Wang, D. Lian, Y. Wang, R. Tang, 和 E. Chen，“理解大型语言模型代理的规划: 综述”，arXiv 预印本 arXiv:2402.02716，2024 年。1

[17] S. V. Albrecht and P. Stone, ”Autonomous agents modelling other agents: A comprehensive survey and open problems,” Artificial Intelligence, vol. 258, pp. 66-95, 2018. 1

S. V. Albrecht 和 P. Stone，“自主代理对其他代理的建模: 全面综述与未解决问题”，《人工智能》，第 258 卷，第 66-95 页，2018 年。1

[18] G. Anscombe, ”Intention,” 2000. 1

G. Anscombe，“意图”，2000 年。1

[19] D. C. Dennett, ”Précis of the intentional stance,” Behavioral and brain sciences, vol. 11, no. 3, pp. 495-505, 1988. 1

D. C. Dennett，“意向立场的概要”，《行为与脑科学》，第 11 卷第 3 期，第 495-505 页，1988 年。1

[20] Y. Shoham, ”Agent-oriented programming,” Artificial intelligence, vol. 60, no. 1, pp. 51-92, 1993. 1

Y. Shoham，“面向代理的编程”，《人工智能》，第 60 卷第 1 期，第 51-92 页，1993 年。1

[21] D. L. Poole and A. K. Mackworth, Artificial Intelligence: foundations of computational agents. Cambridge University Press, 2010. 1

D. L. Poole 和 A. K. Mackworth，《人工智能: 计算代理的基础》，剑桥大学出版社，2010 年。1

[22] B. Inkster, S. Sarda, V. Subramanian et al., ”An empathy-driven, conversational artificial intelligence agent (wysa) for digital mental well-being: real-world data evaluation mixed-methods study,” JMIR mHealth and uHealth, vol. 6, no. 11, p. e12106, 2018. 1

B. Inkster, S. Sarda, V. Subramanian 等，“一种基于同理心的对话式人工智能代理 (Wysa) 用于数字心理健康: 真实数据评估混合方法研究”，《JMIR mHealth and uHealth》，第 6 卷第 11 期，e12106，2018 年。1

[23] J. Gao, M. Galley, and L. Li, ”Neural approaches to conversational ai,” in The 41st international ACM SIGIR conference on research & development in information retrieval, 2018, pp. 1371-1374. 1

J. Gao, M. Galley, 和 L. Li，“对话式人工智能的神经网络方法”，载于第 41 届国际 ACM SIGIR 信息检索研究与开发会议，2018 年，第 1371-1374 页。1

[24] E. Luger and A. Sellen, ”” like having a really bad pa” the gulf between user expectation and experience of conversational agents,” in Proceedings of the 2016 CHI conference on human factors in computing systems, 2016, pp. 5286-5297. 1

E. Luger 和 A. Sellen，“‘就像有一个非常糟糕的私人助理’——用户对话代理的期望与体验之间的鸿沟”，载于 2016 年 CHI 人机交互大会论文集，2016 年，第 5286-5297 页。1

[25] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, ”Power to the people: The role of humans in interactive machine learning,” AI magazine, vol. 35, no. 4, pp. 105-120, 2014. 1

S. Amershi, M. Cakmak, W. B. Knox, 和 T. Kulesza,“赋权于人: 人类在交互式机器学习中的角色”,《人工智能杂志》，第 35 卷第 4 期，第 105-120 页，2014 年。1

[26] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, ”Deep reinforcement learning from human preferences,” Advances in neural information processing systems, vol. 30, 2017. 1

P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, 和 D. Amodei，“基于人类偏好的深度强化学习”,《神经信息处理系统进展》，第 30 卷，2017 年。1

[27] J. Köhl, R. Kolnaar, and W. J. Ravensberg, ”Mode of action of microbial biological control agents against plant diseases: relevance beyond efficacy,” Frontiers in plant science, vol. 10, p. 845, 2019. 1

J. Köhl, R. Kolnaar, 和 W. J. Ravensberg，“微生物生物防治剂对植物病害的作用机制: 超越效果的相关性”,《植物科学前沿》，第 10 卷，845 页，2019 年。1

[28] A. Radford, ”Improving language understanding by generative pre-training,” 2018. 1, 2, 3, 15, 16

A. Radford，“通过生成式预训练提升语言理解”，2018 年。1, 2, 3, 15, 16

[29] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever et al., ”Language models are unsupervised multitask learners,” OpenAI blog, vol. 1, no. 8, p. 9, 2019. 1, 2, 3, 15, 16

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever 等，“语言模型是无监督的多任务学习者”，OpenAI 博客，第 1 卷第 8 期，第 9 页，2019 年。1, 2, 3, 15, 16

[30] T. B. Brown, ”Language models are few-shot learners,” arXiv preprint arXiv:2005.14165, 2020. 1, 2, 3, 15, 16

T. B. Brown，“语言模型是少样本学习者”，arXiv 预印本 arXiv:2005.14165，2020 年。1, 2, 3, 15, 16

[31] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat et al., ”Gpt-4 technical report,” arXiv preprint arXiv:2303.08774, 2023. 1, 2, 3, 15, 17

J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat 等，“GPT-4 技术报告,”arXiv 预印本 arXiv:2303.08774, 2023. 1, 2, 3, 15, 17

[32] R. Bavishi, E. Elsen, C. Hawthorne, M. Nye, A. Odena, A. Somani, and S. Taşırlar, ”Fuyu-8b: A multimodal architecture for ai agents,” 2023. 1

R. Bavishi, E. Elsen, C. Hawthorne, M. Nye, A. Odena, A. Somani, 和 S. Taşırlar, “Fuyu-8b: 一种用于 AI 代理的多模态架构,”2023. 1

[33] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, ”Voyager: An open-ended embodied agent with large language models,” arXiv preprint arXiv:2305.16291, 2023. 1

G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, 和 A. Anandkumar, “Voyager: 基于大型语言模型的开放式具身代理,”arXiv 预印本 arXiv:2305.16291, 2023. 1

[34] S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou et al., ”Metagpt: Meta programming for multi-agent collaborative framework,” arXiv preprint arXiv:2308.00352, 2023. 1

S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou 等, “MetaGPT: 多代理协作框架的元编程,”arXiv 预印本 arXiv:2308.00352, 2023. 1

[35] G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, ”Camel: Communicative agents for” mind” exploration of large language model society,” Advances in Neural Information Processing Systems, vol. 36, pp. 51991-52008, 2023. 1

G. Li, H. Hammoud, H. Itani, D. Khizbullin, 和 B. Ghanem, “CAMEL: 用于大型语言模型社会‘思维’探索的交流代理,”神经信息处理系统进展, 第 36 卷, 页 51991-52008, 2023. 1

[36] J. S. Park, J. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, ”Generative agents: Interactive simulacra of human behavior,” in Proceedings of the 36th annual acm symposium on user interface software and technology, 2023, pp. 1-22. 1

J. S. Park, J. O’Brien, C. J. Cai, M. R. Morris, P. Liang, 和 M. S. Bernstein, “生成代理: 人类行为的交互模拟,”载于第 36 届 ACM 用户界面软件与技术年会论文集, 2023, 页 1-22. 1

[37] D. A. Boiko, R. MacKnight, and G. Gomes, ”Emergent autonomous scientific research capabilities of large language models,” arXiv preprint arXiv:2304.05332, 2023. 1

D. A. Boiko, R. MacKnight, 和 G. Gomes, “大型语言模型的自主科学研究能力的涌现,”arXiv 预印本 arXiv:2304.05332, 2023. 1

[38] C. Qian, X. Cong, C. Yang, W. Chen, Y. Su, J. Xu, Z. Liu, and M. Sun, ”Communicative agents for software development,” arXiv preprint arXiv:2307.07924, vol. 6, no. 3, 2023. 1

C. Qian, X. Cong, C. Yang, W. Chen, Y. Su, J. Xu, Z. Liu, 和 M. Sun, “用于软件开发的交流代理,”arXiv 预印本 arXiv:2307.07924, 第 6 卷, 第 3 期, 2023. 1

[39] Y. Xia, M. Shenoy, N. Jazdi, and M. Weyrich, ”Towards autonomous system: flexible modular production system enhanced with large language model agents,” in 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2023, pp. 1-8. 1

Y. Xia, M. Shenoy, N. Jazdi, 和 M. Weyrich, “迈向自主系统: 由大型语言模型代理增强的灵活模块化生产系统,”载于 2023 年 IEEE 第 28 届新兴技术与工厂自动化国际会议 (ETFA), IEEE, 2023, 页 1-8. 1

[40] I. Dasgupta, C. Kaeser-Chen, K. Marino, A. Ahuja, S. Babayan, F. Hill, and R. Fergus, ”Collaborating with language models for embodied reasoning,” arXiv preprint arXiv:2302.00763, 2023. 1

I. Dasgupta, C. Kaeser-Chen, K. Marino, A. Ahuja, S. Babayan, F. Hill, 和 R. Fergus, “与语言模型协作进行具身推理,”arXiv 预印本 arXiv:2302.00763, 2023. 1

[41] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong et al., ”Chatdev: Communicative agents for software development,” in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 15174-15186. 1

C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong 等,“ChatDev: 用于软件开发的交流代理,”载于第 62 届计算语言学协会年会论文集 (第一卷: 长文), 2024, 页 15174-15186. 1

[42] Y. Dong, X. Jiang, Z. Jin, and G. Li, ”Self-collaboration code generation via chatgpt,” ACM Transactions on Software Engineering and Methodology, vol. 33, no. 7, pp. 1-38, 2024. 1

Y. Dong, X. Jiang, Z. Jin, 和 G. Li, “通过 ChatGPT 实现自我协作代码生成,”ACM 软件工程与方法学汇刊, 第 33 卷, 第 7 期, 页 1-38, 2024. 1

[43] B. Goertzel, ”Artificial general intelligence: concept, state of the art, and future prospects,” Journal of Artificial General Intelligence, vol. 5, no. 1, p. 1, 2014. 1

B. Goertzel, “通用人工智能 (Artificial General Intelligence，AGI): 概念、现状与未来展望,”通用人工智能杂志, 第 5 卷, 第 1 期, 页 1, 2014. 1

[44] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou et al., ”The rise and potential of large language model based agents: A survey,” arXiv preprint arXiv:2309.07864, 2023. 1, 11, 12

Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou 等，“基于大型语言模型代理的兴起与潜力: 综述”，arXiv 预印本 arXiv:2309.07864，2023 年。1, 11, 12

[45] H. Furuta, Y. Matsuo, A. Faust, and I. Gur, ”Exposing limitations of language model agents in sequential-task compositions on the web,” in ICLR 2024 Workshop on Large Language Model (LLM) Agents, 2024. 1

H. Furuta, Y. Matsuo, A. Faust, 和 I. Gur，“揭示语言模型代理在网页顺序任务组合中的局限性”，发表于 ICLR 2024 大型语言模型 (LLM) 代理研讨会，2024 年。1

[46] W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Dong, M. Ding et al., ”Cogagent: A visual language model for gui agents,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 14281-14290. 1, 3, 10, 11, 19, 20

W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Dong, M. Ding 等，“Cogagent: 面向图形用户界面代理的视觉语言模型”，发表于 IEEE/CVF 计算机视觉与模式识别会议论文集，2024 年，页码 14281-14290。1, 3, 10, 11, 19, 20

[47] B. Zheng, B. Gou, J. Kil, H. Sun, and Y. Su, ”Gpt-4v (ision) is a generalist web agent, if grounded,” arXiv preprint arXiv:2401.01614, 2024. 1, 3, 8, 15

B. Zheng, B. Gou, J. Kil, H. Sun, 和 Y. Su，“GPT-4V(视觉) 是通用型网页代理，前提是有基础支持”，arXiv 预印本 arXiv:2401.01614，2024 年。1, 3, 8, 15

[48] C. Zhang, Z. Yang, J. Liu, Y. Han, X. Chen, Z. Huang, B. Fu, and G. Yu, ”Appagent: Multimodal agents as smartphone users,” arXiv preprint arXiv:2312.13771, 2023. 1, 3, 7, 8, 10, 11, 16, 17, 19

C. Zhang, Z. Yang, J. Liu, Y. Han, X. Chen, Z. Huang, B. Fu, 和 G. Yu，“Appagent: 作为智能手机用户的多模态代理”，arXiv 预印本 arXiv:2312.13771，2023 年。1, 3, 7, 8, 10, 11, 16, 17, 19

[49] Y. Song, Y. Bian, Y. Tang, and Z. Cai, ”Navigating interfaces with ai for enhanced user interaction,” arXiv preprint arXiv:2312.11190, 2023. 1, 16

Y. Song, Y. Bian, Y. Tang, 和 Z. Cai，“利用人工智能导航界面以增强用户交互”，arXiv 预印本 arXiv:2312.11190，2023 年。1, 16

[50] H. Wen, Y. Li, G. Liu, S. Zhao, T. Yu, T. J.-J. Li, S. Jiang, Y. Liu, Y. Zhang, and Y. Liu, ”Autodroid: Llm-powered task automation in android,” in Proceedings of the 30th Annual International Conference on Mobile Computing and Networking, 2024, pp. 543-557. 1, 2, 3,7,8,9,11,16,19,29

H. Wen, Y. Li, G. Liu, S. Zhao, T. Yu, T. J.-J. Li, S. Jiang, Y. Liu, Y. Zhang, 和 Y. Liu，“AutoDroid: 基于大型语言模型的安卓任务自动化”，发表于第 30 届国际移动计算与网络会议论文集，2024 年，页码 543-557。1, 2, 3,7,8,9,11,16,19,29

[51] J. Wang, H. Xu, J. Ye, M. Yan, W. Shen, J. Zhang, F. Huang, and J. Sang, ”Mobile-agent: Autonomous multi-modal mobile device agent with visual perception,” arXiv preprint arXiv:2401.16158, 2024. 1, 2, 3, 7, 8, 16, 18, 19, 31

J. Wang, H. Xu, J. Ye, M. Yan, W. Shen, J. Zhang, F. Huang, 和 J. Sang，“Mobile-agent: 具备视觉感知的自主多模态移动设备代理”，arXiv 预印本 arXiv:2401.16158，2024 年。1, 2, 3, 7, 8, 16, 18, 19, 31

[52] J. Wang, H. Xu, H. Jia, X. Zhang, M. Yan, W. Shen, J. Zhang, F. Huang, and J. Sang, ”Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration,” arXiv preprint arXiv:2406.01014, 2024. 1, 2, 3, 7, 8, 11, 12, 13, 16, 18, 19, 31

J. Wang, H. Xu, H. Jia, X. Zhang, M. Yan, W. Shen, J. Zhang, F. Huang, 和 J. Sang，“Mobile-agent-v2: 通过多代理协作实现高效导航的移动设备操作助手”，arXiv 预印本 arXiv:2406.01014，2024 年。1, 2, 3, 7, 8, 11, 12, 13, 16, 18, 19, 31

[53] H. Wen, H. Wang, J. Liu, and Y. Li, ”Droidbot-gpt: Gpt-powered ui automation for android,” arXiv preprint arXiv:2304.07061, 2023. 2,3,8,9,16,19

H. Wen, H. Wang, J. Liu, 和 Y. Li，“DroidBot-GPT: 基于 GPT 的安卓用户界面自动化”，arXiv 预印本 arXiv:2304.07061，2023 年。2,3,8,9,16,19

[54] Z. Liu, C. Li, C. Chen, J. Wang, B. Wu, Y. Wang, J. Hu, and Q. Wang, ”Vision-driven automated mobile gui testing via multimodal large language model,” arXiv preprint arXiv:2407.03037, 2024. 2, 3, 10, 16, 18, 19

Z. Liu, C. Li, C. Chen, J. Wang, B. Wu, Y. Wang, J. Hu, 和 Q. Wang，“基于视觉驱动的多模态大型语言模型自动化移动图形用户界面测试”，arXiv 预印本 arXiv:2407.03037，2024 年。2, 3, 10, 16, 18, 19

[55] J. Zhang, C. Zhao, Y. Zhao, Z. Yu, M. He, and J. Fan, ”Mobile-experts: A dynamic tool-enabled agent team in mobile devices,” arXiv preprint arXiv:2407.03913, 2024. 2, 3, 8, 14, 16, 18, 19

张杰, 赵晨, 赵阳, 余志, 何明, 范军, “Mobile-experts: 移动设备中的动态工具驱动代理团队,”arXiv 预印本 arXiv:2407.03913, 2024. 2, 3, 8, 14, 16, 18, 19

[56] Y. Lu, J. Yang, Y. Shen, and A. Awadallah, ”Omniparser for pure vision based gui agent,” arXiv preprint arXiv:2408.00203, 2024. 2, 3, 10, 11, 16, 17, 19

陆洋, 杨杰, 沈阳, Awadallah, A., “Omniparser: 基于纯视觉的 GUI 代理,”arXiv 预印本 arXiv:2408.00203, 2024. 2, 3, 10, 11, 16, 17, 19

[57] B. Wang, G. Li, and Y. Li, ”Enabling conversational interaction with mobile ui using large language models,” in Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 2023, pp. 1-17. 3, 8, 9, 16, 19

王博, 李刚, 李阳, “利用大型语言模型实现与移动用户界面的对话交互,”载于 2023 年 CHI 人机交互大会论文集, 2023, 页码 1-17. 3, 8, 9, 16, 19

[58] Y. Guan, D. Wang, Z. Chu, S. Wang, F. Ni, R. Song, L. Li, J. Gu, and C. Zhuang, ”Intelligent virtual assistants with Ilm-based process automation,” arXiv preprint arXiv:2312.06677, 2023. 3, 10

管宇, 王东, 朱志, 王松, 倪飞, 宋睿, 李磊, 顾军, 庄超, “基于 Ilm 的流程自动化智能虚拟助手,”arXiv 预印本 arXiv:2312.06677, 2023. 3, 10

[59] Y. Fan, L. Ding, C.-C. Kuo, S. Jiang, Y. Zhao, X. Guan, J. Yang, Y. Zhang, and X. E. Wang, ”Read anywhere pointed: Layout-aware gui screen reading with tree-of-lens grounding,” 2024. [Online]. Available: https://arxiv.org/abs/2406.19263 3, 10

范勇, 丁磊, 郭志成, 蒋森, 赵勇, 关晓, 杨军, 张勇, 王晓东, ” 随处可读指向: 基于树状透镜定位的布局感知 GUI 屏幕阅读,” 2024. [在线]. 可用: https://arxiv.org/abs/2406.19263 3, 10

[60] A. Yan, Z. Yang, W. Zhu, K. Lin, L. Li, J. Wang, J. Yang, Y. Zhong, J. McAuley, J. Gao et al., ”Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation,” arXiv preprint arXiv:2311.07562, 2023. 3, 10, 16, 17, 19

颜安, 杨志, 朱伟, 林凯, 李磊, 王军, 杨军, 钟勇, 麦考利, 高健等, ” 奇境中的 GPT-4V: 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, 2023. 3, 10, 16, 17, 19

[61] S. Lee, J. Choi, J. Lee, M. H. Wasi, H. Choi, S. Y. Ko, S. Oh, and I. Shin, ”Explore, select, derive, and recall: Augmenting llm with human-like memory for mobile task automation,” arXiv preprint arXiv:2312.03003, 2023. 3, 7, 10, 11, 16, 19, 31, 32

李胜, 崔俊, 李军, Wasi M. H., 崔浩, 高胜尧, 吴胜, 申一, ” 探索、选择、推导与回忆: 用类人记忆增强大型语言模型以实现移动任务自动化,” arXiv 预印本 arXiv:2312.03003, 2023. 3, 7, 10, 11, 16, 19, 31, 32

[62] Q. Wu, D. Gao, K. Q. Lin, Z. Wu, X. Guo, P. Li, W. Zhang, H. Wang, and M. Z. Shou, ”Gui action narrator: Where and when did that action take place?” arXiv preprint arXiv:2406.13719, 2024. 3, 10, 18, 19

吴强, 高东, 林启强, 吴志, 郭翔, 李鹏, 张伟, 王浩, 寿明哲, ”GUI 动作叙述者: 该动作发生的时间与地点？ ” arXiv 预印本 arXiv:2406.13719, 2024. 3, 10, 18, 19

[63] Q. Wu, W. Xu, W. Liu, T. Tan, J. Liu, A. Li, J. Luan, B. Wang, and S. Shang, ”Mobilevlm: A vision-language model for better intra-and inter-ui understanding,” arXiv preprint arXiv:2409.14818, 2024. 3, 10, 25

吴强, 徐伟, 刘伟, 谭涛, 刘军, 李安, 栾军, 王斌, 尚松, ”MobileVLM: 一种提升界面内外理解的视觉-语言模型,” arXiv 预印本 arXiv:2409.14818, 2024. 3, 10, 25

[64] Y. Li, C. Zhang, W. Yang, B. Fu, P. Cheng, X. Chen, L. Chen, and Y. Wei, ”Appagent v2: Advanced agent for flexible mobile interactions,” arXiv preprint arXiv:2408.11824, 2024. 3, 11, 18, 19

李阳, 张超, 杨伟, 傅斌, 程鹏, 陈晓, 陈磊, 魏勇, ”AppAgent v2: 灵活移动交互的高级代理,” arXiv 预印本 arXiv:2408.11824, 2024. 3, 11, 18, 19

[65] W. Jiang, Y. Zhuang, C. Song, X. Yang, J. T. Zhou, and C. Zhang, ”Appagentx: Evolving gui agents as proficient smartphone users,” arXiv preprint arXiv:2503.02268, 2025. 3, 11

姜伟, 庄勇, 宋超, 杨翔, 周建涛, 张超, ”AppAgentX: 作为熟练智能手机用户的进化 GUI 代理,” arXiv 预印本 arXiv:2503.02268, 2025. 3, 11

[66] Z. Zhang and A. Zhang, ”You only look at screens: Multimodal chain-of-action agents,” arXiv preprint arXiv:2309.11436, 2023. 3, 7, 10, 11, 19, 20, 30

张志, 张安, ” 你只看屏幕: 多模态动作链代理,” arXiv 预印本 arXiv:2309.11436, 2023. 3, 7, 10, 11, 19, 20, 30

[67] G. Baechler, S. Sunkara, M. Wang, F. Zubach, H. Mansoor, V. Etter, V. Cărbune, J. Lin, J. Chen, and A. Sharma, ”Screenai: A vision-language model for ui and infographics understanding,” arXiv preprint arXiv:2402.04615, 2024. 3, 7, 11, 19, 20, 22

Baechler G., Sunkara S., Wang M., Zubach F., Mansoor H., Etter V., Cărbune V., Lin J., Chen J., Sharma A., ”ScreenAI: 用于界面和信息图理解的视觉-语言模型,” arXiv 预印本 arXiv:2402.04615, 2024. 3, 7, 11, 19, 20, 22

[68] J. Wang, H. Xu, X. Zhang, M. Yan, J. Zhang, F. Huang, and J. Sang, ”Mobile-agent-v: Learning mobile device operation through video-guided multi-agent collaboration,” arXiv preprint arXiv:2502.17110, 2025. 3, 11, 19

王军, 徐浩, 张晓, 颜明, 张军, 黄飞, 桑军, ”Mobile-Agent-V: 通过视频引导的多代理协作学习移动设备操作,” arXiv 预印本 arXiv:2502.17110, 2025. 3, 11, 19

[69] P. Cheng, Z. Wu, Z. Wu, A. Zhang, Z. Zhang, and G. Liu, ”Os-kairos: Adaptive interaction for mllm-powered gui agents,” arXiv preprint arXiv:2503.16465, 2025. 3

程鹏, 吴志, 吴志, 张安, 张志, 刘刚, ”OS-Kairos: 面向多模态大型语言模型驱动 GUI 代理的自适应交互,” arXiv 预印本 arXiv:2503.16465, 2025. 3

[70] Y. Sun, S. Zhao, T. Yu, H. Wen, S. Va, M. Xu, Y. Li, and C. Zhang, ”Gui-xplore: Empowering generalizable gui agents with one exploration,” arXiv preprint arXiv:2503.17709, 2025. 3

孙阳, 赵爽, 余涛, 文浩, Va S., 徐明, 李阳, 张超, ”GUI-Xplore: 通过一次探索赋能通用 GUI 代理,” arXiv 预印本 arXiv:2503.17709, 2025. 3

[71] X. Ma, Z. Zhang, and H. Zhao, ”Coco-agent: A comprehensive cognitive mllm agent for smartphone gui automation,” in Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 9097-9110. 3, 11, 20, 21, 32

X. Ma, Z. Zhang, 和 H. Zhao, “Coco-agent: 用于智能手机 GUI 自动化的综合认知大语言模型 (mllm) 代理,”载于 2024 年计算语言学协会 (ACL) 研究成果, 2024, 页码 9097-9110. 3, 11, 20, 21, 32

[72] Z. Song, Y. Li, M. Fang, Z. Chen, Z. Shi, and Y. Huang, ”Mmac-copilot: Multi-modal agent collaboration operating system copilot,” arXiv preprint arXiv:2404.18074, 2024. 3, 12, 13

Z. Song, Y. Li, M. Fang, Z. Chen, Z. Shi, 和 Y. Huang, “Mmac-copilot: 多模态代理协作操作系统副驾驶,”arXiv 预印本 arXiv:2404.18074, 2024. 3, 12, 13

[73] W. Tan, W. Zhang, X. Xu, H. Xia, Z. Ding, B. Li, B. Zhou, J. Yue, J. Jiang, Y. Li et al., ”Cradle: Empowering foundation agents towards general computer control,” in NeurIPS 2024 Workshop on Open-World Agents. 3, 12, 14

W. Tan, W. Zhang, X. Xu, H. Xia, Z. Ding, B. Li, B. Zhou, J. Yue, J. Jiang, Y. Li 等, “Cradle: 赋能基础代理实现通用计算机控制,”载于 NeurIPS 2024 开放世界代理研讨会. 3, 12, 14

[74] Z. Wang, H. Xu, J. Wang, X. Zhang, M. Yan, J. Zhang, F. Huang, J and H. Ji, ”Mobile-agent-e: Self-evolving mobile assistant for complex tasks,” arXiv preprint arXiv:2501.11733, 2025. 3, 14, 18, 19

Z. Wang, H. Xu, J. Wang, X. Zhang, M. Yan, J. Zhang, F. Huang, J 和 H. Ji, “Mobile-agent-e: 面向复杂任务的自我进化移动助手,”arXiv 预印本 arXiv:2501.11733, 2025. 3, 14, 18, 19

[75] T. Huang, C. Yu, W. Shi, Z. Peng, D. Yang, W. Sun, and Y. Shi, ”Promptrpa: Generating robotic process automation on smart-phones from textual prompts,” arXiv preprint arXiv:2404.02475, 2024. 3, 14, 16

T. Huang, C. Yu, W. Shi, Z. Peng, D. Yang, W. Sun, 和 Y. Shi, “Promptrpa: 基于文本提示生成智能手机机器人流程自动化,”arXiv 预印本 arXiv:2404.02475, 2024. 3, 14, 16

[76] Y. Zhou, S. Wang, S. Dai, Q. Jia, Z. Du, Z. Dong, and J. Xu, ”Chop: Mobile operating assistant with constrained high-frequency optimized subtask planning,” arXiv preprint arXiv:2503.03743, 2025. 3, 14

Y. Zhou, S. Wang, S. Dai, Q. Jia, Z. Du, Z. Dong, 和 J. Xu, “Chop: 具有限制性高频优化子任务规划的移动操作助手,”arXiv 预印本 arXiv:2503.03743, 2025. 3, 14

[77] S. Agashe, K. Wong, V. Tu, J. Yang, A. Li, and X. E. Wang, ”Agent s2: A compositional generalist-specialist framework for computer use agents,” arXiv preprint arXiv:2504.00906, 2025. 3, 14

S. Agashe, K. Wong, V. Tu, J. Yang, A. Li, 和 X. E. Wang, “Agent s2: 面向计算机使用代理的组合型通用-专家框架,”arXiv 预印本 arXiv:2504.00906, 2025. 3, 14

[78] X. Zhang, Y. Deng, Z. Ren, S.-K. Ng, and T.-S. Chua, ”Ask-before-plan: Proactive language agents for real-world planning,” arXiv preprint arXiv:2406.12639, 2024. 3, 14

X. Zhang, Y. Deng, Z. Ren, S.-K. Ng, 和 T.-S. Chua,“Ask-before-plan: 面向现实规划的主动语言代理,”arXiv 预印本 arXiv:2406.12639, 2024. 3, 14

[79] P. Sodhi, S. Branavan, Y. Artzi, and R. McDonald, ”Step: Stacked llm policies for web actions,” in First Conference on Language Modeling, 2024. 3, 7, 14, 31

P. Sodhi, S. Branavan, Y. Artzi, 和 R. McDonald, “Step: 用于网页操作的堆叠大语言模型策略,”载于第一届语言建模会议, 2024. 3, 7, 14, 31

[80] B. Gou, R. Wang, B. Zheng, Y. Xie, C. Chang, Y. Shu, H. Sun, and Y. Su, ”Navigating the digital world as humans do: Universal visual grounding for gui agents,” arXiv preprint arXiv:2410.05243, 2024. 3, 7, 8, 15, 31

B. Gou, R. Wang, B. Zheng, Y. Xie, C. Chang, Y. Shu, H. Sun, 和 Y. Su, “像人类一样导航数字世界: 面向 GUI 代理的通用视觉定位,”arXiv 预印本 arXiv:2410.05243, 2024. 3, 7, 8, 15, 31

[81] F. Christianos, G. Papoudakis, T. Coste, J. Hao, J. Wang, and K. Shao, ”Lightweight neural app control,” arXiv preprint arXiv:2410.17883, 2024. 3, 15, 31

F. Christianos, G. Papoudakis, T. Coste, J. Hao, J. Wang, 和 K. Shao, “轻量级神经应用控制,”arXiv 预印本 arXiv:2410.17883, 2024. 3, 15, 31

[82] J. Hoscilowicz, B. Maj, B. Kozakiewicz, O. Tymoshchuk, and A. Janicki, ”Clickagent: Enhancing ui location capabilities of autonomous agents,” arXiv preprint arXiv:2410.11872, 2024. 3, 8, 15

J. Hoscilowicz, B. Maj, B. Kozakiewicz, O. Tymoshchuk, 和 A. Janicki, “Clickagent: 增强自主代理的 UI 定位能力,”arXiv 预印本 arXiv:2410.11872, 2024. 3, 8, 15

[83] Y. Wang, H. Zhang, J. Tian, and Y. Tang, ”Ponder & press: Advancing visual gui agent towards general computer control,” arXiv preprint arXiv:2412.01268, 2024. 3, 15

Y. Wang, H. Zhang, J. Tian, 和 Y. Tang, “Ponder & press: 推动视觉 GUI 代理迈向通用计算机控制,” arXiv 预印本 arXiv:2412.01268, 2024. 3, 15

[84] M. Taeb, A. Swearngin, E. Schoop, R. Cheng, Y. Jiang, and J. Nichols, ”Axnav: Replaying accessibility tests from natural language,” in Proceedings of the CHI Conference on Human Factors in Computing Systems, 2024, pp. 1-16. 3, 16, 19

M. Taeb, A. Swearngin, E. Schoop, R. Cheng, Y. Jiang, 和 J. Nichols, “Axnav: 基于自然语言的无障碍测试重放,”载于 2024 年 CHI 人机交互大会论文集，页码 1-16。3, 16, 19

[85] Y. Song, Y. Bian, Y. Tang, G. Ma, and Z. Cai, ”Visiontasker: Mobile task automation using vision based ui understanding and llm task planning,” in Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology, 2024, pp. 1-17. 3, 10, 11, 18, 19, 31

Y. Song, Y. Bian, Y. Tang, G. Ma, 和 Z. Cai, “Visiontasker: 基于视觉的用户界面理解与大语言模型 (LLM) 任务规划的移动任务自动化，”载于第 37 届 ACM 用户界面软件与技术年会论文集，2024，页码 1-17。3, 10, 11, 18, 19, 31

[86] X. Li, J. Cao, Y. Liu, S.-C. Cheung, and H. Wang, ”Reusedroid: A vlm-empowered android ui test migrator boosted by active feedback,” arXiv preprint arXiv:2504.02357, 2025. 3

X. Li, J. Cao, Y. Liu, S.-C. Cheung, 和 H. Wang, “Reusedroid: 一种由视觉语言模型 (VLM) 驱动并通过主动反馈增强的安卓 UI 测试迁移工具,”arXiv 预印本 arXiv:2504.02357，2025。3

[87] B. F. Demissie, Y. N. Tun, L. K. Shar, and M. Ceccato, ”Vlm-fuzz: Vision language model assisted recursive depth-first search exploration for effective ui testing of android apps,” arXiv preprint arXiv:2504.11675, 2025. 3

B. F. Demissie, Y. N. Tun, L. K. Shar, 和 M. Ceccato,“Vlm-fuzz: 视觉语言模型辅助的递归深度优先搜索探索，用于安卓应用的高效 UI 测试,”arXiv 预印本 arXiv:2504.11675，2025。3

[88] S. Nong, J. Zhu, R. Wu, J. Jin, S. Shan, X. Huang, and W. Xu, ”Mobileflow: A multimodal llm for mobile gui agent,” arXiv preprint arXiv:2407.04346, 2024. 3, 20, 21

S. Nong, J. Zhu, R. Wu, J. Jin, S. Shan, X. Huang, 和 W. Xu, “Mobileflow: 面向移动 GUI 代理的多模态大语言模型,”arXiv 预印本 arXiv:2407.04346，2024。3, 20, 21

[89] K. Q. Lin, L. Li, D. Gao, Z. Yang, S. Wu, Z. Bai, W. Lei, L. Wang, and M. Z. Shou, ”Showui: One vision-language-action model for gui visual agent,” arXiv preprint arXiv:2411.17465, 2024. 3, 20, 21

K. Q. Lin, L. Li, D. Gao, Z. Yang, S. Wu, Z. Bai, W. Lei, L. Wang, 和 M. Z. Shou, “Showui: 用于 GUI 视觉代理的单一视觉-语言-动作模型,”arXiv 预印本 arXiv:2411.17465，2024。3, 20, 21

[90] Y. Xu, Z. Wang, J. Wang, D. Lu, T. Xie, A. Saha, D. Sahoo, T. Yu, and C. Xiong, ”Aguvis: Unified pure vision agents for autonomous gui interaction,” arXiv preprint arXiv:2412.04454, 2024. 3, 20, 21

Y. Xu, Z. Wang, J. Wang, D. Lu, T. Xie, A. Saha, D. Sahoo, T. Yu, 和 C. Xiong, “Aguvis: 用于自主 GUI 交互的统一纯视觉代理,”arXiv 预印本 arXiv:2412.04454，2024。3, 20, 21

[91] D. Luo, B. Tang, K. Li, G. Papoudakis, J. Song, S. Gong, J. Hao, J. Wang, and K. Shao, ”Vimo: A generative visual gui world model for app agent,” arXiv preprint arXiv:2504.13936, 2025. 3

D. Luo, B. Tang, K. Li, G. Papoudakis, J. Song, S. Gong, J. Hao, J. Wang, 和 K. Shao, “Vimo: 面向应用代理的生成式视觉 GUI 世界模型,”arXiv 预印本 arXiv:2504.13936，2025。3

[92] Y. Qin, Y. Ye, J. Fang, H. Wang, S. Liang, S. Tian, J. Zhang, J. Li, Y. Li, S. Huang et al., ”Ui-tars: Pioneering automated gui interaction with native agents,” arXiv preprint arXiv:2501.12326, 2025. 3, 20, 21

Y. Qin, Y. Ye, J. Fang, H. Wang, S. Liang, S. Tian, J. Zhang, J. Li, Y. Li, S. Huang 等, “Ui-tars: 开创性的原生代理自动化 GUI 交互,”arXiv 预印本 arXiv:2501.12326，2025。3, 20, 21

[93] T. Li, G. Li, J. Zheng, P. Wang, and Y. Li, ”Mug: Interactive multimodal grounding on user interfaces,” arXiv preprint arXiv:2209.15099, 2022. 3, 21

T. Li, G. Li, J. Zheng, P. Wang, 和 Y. Li, “Mug: 用户界面上的交互式多模态定位，”arXiv 预印本 arXiv:2209.15099，2022。3, 21

[94] Y. Qian, Y. Lu, A. G. Hauptmann, and O. Riva, ”Visual grounding for user interfaces,” in Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), 2024, pp. 97-107. 3, 20, 21

Y. Qian, Y. Lu, A. G. Hauptmann, 和 O. Riva, “用户界面的视觉定位,”载于 2024 年北美计算语言学协会人类语言技术会议 (工业轨) 论文集，第 6 卷，页码 97-107，2024。3, 20, 21

[95] J. Zhang, Y. Yu, M. Liao, W. Li, J. Wu, and Z. Wei, ”Ui-hawk: Unleashing the screen stream understanding for gui agents,” 2024. 3, 7, 15, 20, 21, 22, 31

J. Zhang, Y. Yu, M. Liao, W. Li, J. Wu, 和 Z. Wei, “Ui-hawk: 释放屏幕流理解能力的 GUI 代理,”2024。3, 7, 15, 20, 21, 22, 31

[96] Y. Yang, Y. Wang, D. Li, Z. Luo, B. Chen, C. Huang, and J. Li, ”Aria-ui: Visual grounding for gui instructions,” arXiv preprint arXiv:2412.16256, 2024. 3, 20, 21

Y. Yang, Y. Wang, D. Li, Z. Luo, B. Chen, C. Huang, 和 J. Li, “Aria-ui:GUI 指令的视觉定位，” arXiv 预印本 arXiv:2412.16256，2024。3, 20, 21

[97] Z. Wu, Z. Wu, F. Xu, Y. Wang, Q. Sun, C. Jia, K. Cheng, Z. Ding, L. Chen, P. P. Liang et al., ”Os-atlas: A foundation action model for generalist gui agents,” arXiv preprint arXiv:2410.23218, 2024. 3, 20, 21

吴志强, 吴志强, 徐飞, 王勇, 孙强, 贾超, 程凯, 丁志, 陈磊, 梁鹏鹏等, ”Os-atlas: 通用图形用户界面代理的基础动作模型,” arXiv 预印本 arXiv:2410.23218, 2024. 3, 20, 21

[98] Z. Wang, W. Chen, L. Yang, S. Zhou, S. Zhao, H. Zhan, J. Jin, L. Li, Z. Shao, and J. Bu, ”Mp-gui: Modality perception with mllms for gui understanding,” arXiv preprint arXiv:2503.14021, 2025. 3, 20, 22

王志, 陈伟, 杨磊, 周松, 赵帅, 詹浩, 金杰, 李磊, 邵志, 卜军, ”Mp-gui: 利用多模态大语言模型 (MLLMs) 进行图形用户界面理解的模态感知,” arXiv 预印本 arXiv:2503.14021, 2025. 3, 20, 22

[99] Z. Wu, P. Cheng, Z. Wu, T. Ju, Z. Zhang, and G. Liu, ”Smoothing grounding and reasoning for mllm-powered gui agents with query-oriented pivot tasks,” arXiv preprint arXiv:2503.00401, 2025. 3

吴志强, 程鹏, 吴志强, 居涛, 张志, 刘刚, ” 基于查询导向枢纽任务的多模态大语言模型驱动图形用户界面代理的平滑定位与推理,” arXiv 预印本 arXiv:2503.00401, 2025. 3

[100] Y. Fan, H. Zhao, R. Zhang, Y. Shen, X. E. Wang, and G. Wu, ”Gui-bee: Align gui action grounding to novel environments via autonomous exploration,” arXiv preprint arXiv:2501.13896, 2025. 3, 20, 21

范阳, 赵浩, 张锐, 沈阳, 王晓恩, 吴刚, ”Gui-bee: 通过自主探索将图形用户界面动作定位对齐到新环境,” arXiv 预印本 arXiv:2501.13896, 2025. 3, 20, 21

[101] K. You, H. Zhang, E. Schoop, F. Weers, A. Swearngin, J. Nichols, Y. Yang, and Z. Gan, ”Ferret-ui: Grounded mobile ui understanding with multimodal llms,” arXiv preprint arXiv:2404.05719, 2024. 3,7,11,15,20,21,31

游凯, 张浩, 埃里克·斯库普, 弗洛里安·韦尔斯, 安德鲁·斯韦恩金, 尼克·乔治, 杨洋, 甘志, ”Ferret-ui: 基于多模态大语言模型的移动用户界面理解,” arXiv 预印本 arXiv:2404.05719, 2024. 3,7,11,15,20,21,31

[102] Z. Li, K. You, H. Zhang, D. Feng, H. Agrawal, X. Li, M. P. S. Moorthy, J. Nichols, Y. Yang, and Z. Gan, ”Ferret-ui 2: Mastering universal user interface understanding across platforms,” 2024. [Online]. Available: https://arxiv.org/abs/2410.18967 3, 20, 22, 28

李志, 游凯, 张浩, 冯东, 阿格拉瓦尔, 李翔, 莫尔西, 尼克·乔治, 杨洋, 甘志, ”Ferret-ui 2: 跨平台通用用户界面理解的掌握,” 2024. [在线]. 可获取: https://arxiv.org/abs/2410.18967 3, 20, 22, 28

[103] A. Burns, K. Saenko, and B. A. Plummer, ”Tell me what’s next: Textual foresight for generic ui representations,” arXiv preprint arXiv:2406.07822, 2024. 3, 20, 22

伯恩斯, 萨恩科, 普拉默, ” 告诉我接下来是什么：通用用户界面表示的文本前瞻,” arXiv 预印本 arXiv:2406.07822, 2024. 3, 20, 22

[104] Q. Chen, D. Pitawela, C. Zhao, G. Zhou, H.-T. Chen, and Q. Wu, ”Webvln: Vision-and-language navigation on websites,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 2, 2024, pp. 1165-1173. 3, 20, 22

陈强, 皮塔韦拉, 赵超, 周刚, 陈宏涛, 吴强, ”Webvln: 网站上的视觉与语言导航,” 载于 AAAI 人工智能会议论文集, 第 38 卷第 2 期, 2024, 页 1165-1173. 3, 20, 22

[105] K. Cheng, Q. Sun, Y. Chu, F. Xu, Y. Li, J. Zhang, and Z. Wu, ”Seeclick: Harnessing gui grounding for advanced visual gui agents,” arXiv preprint arXiv:2401.10935, 2024. 3, 5, 7, 16, 23, 24, 31

程凯, 孙强, 褚阳, 徐飞, 李阳, 张杰, 吴志强, ”Seeclick: 利用图形用户界面定位推动高级视觉图形用户界面代理,” arXiv 预印本 arXiv:2401.10935, 2024. 3, 5, 7, 16, 23, 24, 31

[106] Y. Liu, P. Li, Z. Wei, C. Xie, X. Hu, X. Xu, S. Zhang, X. Han, H. Yang, and F. Wu, ”Infiguiagent: A multimodal generalist gui agent with native reasoning and reflection,” arXiv preprint arXiv:2501.04575, 2025. 3, 23, 24

刘洋, 李鹏, 魏志, 谢超, 胡翔, 徐翔, 张帅, 韩翔, 杨浩, 吴飞, ”Infiguiagent: 具备本地推理与反思能力的多模态通用图形用户界面代理,” arXiv 预印本 arXiv:2501.04575, 2025. 3, 23, 24

[107] W. Chen, J. Cui, J. Hu, Y. Qin, J. Fang, Y. Zhao, C. Wang, J. Liu, G. Chen, Y. Huo et al., ”Guicourse: From general vision language models to versatile gui agents,” arXiv preprint arXiv:2406.11317, 2024. 3, 5, 16, 23, 24

陈伟, 崔军, 胡军, 秦阳, 方军, 赵阳, 王超, 刘军, 陈刚, 霍阳等, ”Guicourse: 从通用视觉语言模型到多功能图形用户界面代理,” arXiv 预印本 arXiv:2406.11317, 2024. 3, 5, 16, 23, 24

[108] S. Yuan, Z. Chen, Z. Xi, J. Ye, Z. Du, and J. Chen, ”Agent-r: Training language model agents to reflect via iterative self-training,” arXiv preprint arXiv:2501.11425, 2025. 3, 23, 24

袁帅, 陈志, 席志, 叶军, 杜志, 陈军, ”Agent-r: 通过迭代自我训练训练语言模型代理进行反思,” arXiv 预印本 arXiv:2501.11425, 2025. 3, 23, 24

[109] Q. Lu, W. Shao, Z. Liu, F. Meng, B. Li, B. Chen, S. Huang, K. Zhang, Y. Qiao, and P. Luo, ”Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices,” arXiv preprint arXiv:2406.08451, 2024. 3, 5, 16, 23, 24, 27, 28, 32

Q. Lu, W. Shao, Z. Liu, F. Meng, B. Li, B. Chen, S. Huang, K. Zhang, Y. Qiao, 和 P. Luo, “Gui odyssey: 面向移动设备跨应用 GUI 导航的综合数据集,”arXiv 预印本 arXiv:2406.08451, 2024. 3, 5, 16, 23, 24, 27, 28, 32

[110] P. Pawlowski, K. Zawistowski, W. Lapacz, M. Skorupa, A. Wiacek, S. Postansque, and J. Hoscilowicz, ”Tinyclick: Single-turn agent for empowering gui automation,” arXiv preprint arXiv:2410.11871, 2024. 3, 5, 16, 23, 24

P. Pawlowski, K. Zawistowski, W. Lapacz, M. Skorupa, A. Wiacek, S. Postansque, 和 J. Hoscilowicz, “Tinyclick: 赋能 GUI 自动化的单轮代理，”arXiv 预印本 arXiv:2410.11871, 2024. 3, 5, 16, 23, 24

[111] T. Ding, ”Mobileagent: enhancing mobile control via human-machine interaction and sop integration,” arXiv preprint arXiv:2401.04124, 2024. 3, 23, 24, 31

T. Ding, “Mobileagent: 通过人机交互和 SOP 集成增强移动控制，”arXiv 预印本 arXiv:2401.04124, 2024. 3, 23, 24, 31

[112] J. R. A. Moniz, S. Krishnan, M. Ozyildirim, P. Saraf, H. C. Ates, Y. Zhang, H. Yu, and N. Rajshree, ”Realm: Reference resolution as language modeling,” arXiv preprint arXiv:2403.20329, 2024. 3, 23, 24

J. R. A. Moniz, S. Krishnan, M. Ozyildirim, P. Saraf, H. C. Ates, Y. Zhang, H. Yu, 和 N. Rajshree, “Realm: 作为语言建模的引用解析，”arXiv 预印本 arXiv:2403.20329, 2024. 3, 23, 24

[113] G. Papoudakis, T. Coste, Z. Wu, J. Hao, J. Wang, and K. Shao, ”Appvlm: A lightweight vision language model for online app control,” arXiv preprint arXiv:2502.06395, 2025. 3, 31

G. Papoudakis, T. Coste, Z. Wu, J. Hao, J. Wang, 和 K. Shao, “Appvlm: 用于在线应用控制的轻量级视觉语言模型，”arXiv 预印本 arXiv:2502.06395, 2025. 3, 31

[114] G. Dai, S. Jiang, T. Cao, Y. Li, Y. Yang, R. Tan, M. Li, and L. Qiu, ”Advancing mobile gui agents: A verifier-driven approach to practical deployment,” arXiv preprint arXiv:2503.15937, 2025. 3, 20, 21

G. Dai, S. Jiang, T. Cao, Y. Li, Y. Yang, R. Tan, M. Li, 和 L. Qiu, “推进移动 GUI 代理: 基于验证器驱动的实用部署方法，”arXiv 预印本 arXiv:2503.15937, 2025. 3, 20, 21

[115] S. Haque and C. Csallner, ”Infering alt-text for ui icons with large language models during app development,” arXiv preprint arXiv:2409.18060, 2024. 3, 23, 24

S. Haque 和 C. Csallner, “在应用开发过程中利用大型语言模型推断 UI 图标的替代文本，”arXiv 预印本 arXiv:2409.18060, 2024. 3, 23, 24

[116] H. Bai, Y. Zhou, M. Cemri, J. Pan, A. Suhr, S. Levine, and A. Kumar, ”Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning,” arXiv preprint arXiv:2406.11896, 2024. 3, 16, 24, 25, 32

H. Bai, Y. Zhou, M. Cemri, J. Pan, A. Suhr, S. Levine, 和 A. Kumar, “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, 16, 24, 25, 32

[117] T. Wang, Z. Wu, J. Liu, J. Hao, J. Wang, and K. Shao, ”Distrl: An asynchronous distributed reinforcement learning framework for on-device control agents,” arXiv preprint arXiv:2410.14803, 2024. 3, 16, 24, 25

T. Wang, Z. Wu, J. Liu, J. Hao, J. Wang, 和 K. Shao, “Distrl: 用于设备端控制代理的异步分布式强化学习框架，”arXiv 预印本 arXiv:2410.14803, 2024. 3, 16, 24, 25

[118] X. Liu, B. Qin, D. Liang, G. Dong, H. Lai, H. Zhang, H. Zhao, I. L. Iong, J. Sun, J. Wang et al., "Autoglm: Autonomous foundation agents for guis," arXiv preprint arXiv:2411.00820, 2024. 3, 8, 25

X. Liu, B. Qin, D. Liang, G. Dong, H. Lai, H. Zhang, H. Zhao, I. L. Iong, J. Sun, J. Wang 等, "Autoglm: 面向 GUI 的自主基础代理," arXiv 预印本 arXiv:2411.00820, 2024. 3, 8, 25

[119] H. Bai, Y. Zhou, L. E. Li, S. Levine, and A. Kumar, "Digi-q: Learning q-value functions for training device-control agents," arXiv preprint arXiv:2502.15760, 2025. 3, 25, 26

H. Bai, Y. Zhou, L. E. Li, S. Levine, 和 A. Kumar, "Digi-q: 用于训练设备控制代理的 Q 值函数学习," arXiv 预印本 arXiv:2502.15760, 2025. 3, 25, 26

[120] Q. Wu, W. Liu, J. Luan, and B. Wang, "Reachagent: Enhancing mobile agent via page reaching and operation," arXiv preprint arXiv:2502.02955, 2025. 3, 25

Q. Wu, W. Liu, J. Luan, 和 B. Wang, "Reachagent: 通过页面到达和操作增强移动代理," arXiv 预印本 arXiv:2502.02955, 2025. 3, 25

[121] Q. Wu, J. Liu, J. Hao, J. Wang, and K. Shao, "Vsc-rl: Advancing autonomous vision-language agents with variational subgoal-conditioned reinforcement learning," arXiv preprint arXiv:2502.07949, 2025. 3, 25, 26

Q. Wu, J. Liu, J. Hao, J. Wang, 和 K. Shao, "Vsc-rl: 通过变分子目标条件强化学习推进自主视觉语言代理," arXiv 预印本 arXiv:2502.07949, 2025. 3, 25, 26

[122] Z. Lu, Y. Chai, Y. Guo, X. Yin, L. Liu, H. Wang, G. Xiong, and H. Li, "Ui-r1: Enhancing action prediction of gui agents by reinforcement learning," arXiv preprint arXiv:2503.21620, 2025. 3, 25, 26

Z. Lu, Y. Chai, Y. Guo, X. Yin, L. Liu, H. Wang, G. Xiong, 和 H. Li, "Ui-r1: 通过强化学习提升 GUI 代理的动作预测能力," arXiv 预印本 arXiv:2503.21620, 2025. 3, 25, 26

[123] X. Xia and R. Luo, "Gui-r1: A generalist r1-style vision-language action model for gui agents," arXiv preprint arXiv:2504.10458, 2025. 3

X. Xia 和 R. Luo, "Gui-r1: 一种面向 GUI 代理的通用 r1 风格视觉-语言动作模型," arXiv 预印本 arXiv:2504.10458, 2025. 3

[124] Y. Song, D. Yin, X. Yue, J. Huang, S. Li, and B. Y. Lin, "Trial and error: Exploration-based trajectory optimization for llm agents," arXiv preprint arXiv:2403.02502, 2024. 3, 16, 25, 26

Y. Song, D. Yin, X. Yue, J. Huang, S. Li, 和 B. Y. Lin, "试错法: 基于探索的 LLM 代理轨迹优化," arXiv 预印本 arXiv:2403.02502, 2024. 3, 16, 25, 26

[125] P. Putta, E. Mills, N. Garg, S. Motwani, C. Finn, D. Garg, and R. Rafailov, "Agent q: Advanced reasoning and learning for autonomous ai agents," arXiv preprint arXiv:2408.07199, 2024. 3, 25, 26

P. Putta, E. Mills, N. Garg, S. Motwani, C. Finn, D. Garg, 和 R. Rafailov, “Agent q: 面向自主 AI 代理的高级推理与学习,” arXiv 预印本 arXiv:2408.07199, 2024. 3, 25, 26

[126] H. Lai, X. Liu, I. L. Iong, S. Yao, Y. Chen, P. Shen, H. Yu, H. Zhang, X. Zhang, Y. Dong et al., ”Autowebglm: A large language model-based web navigating agent,” in Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 5295-5306. 3, 25, 26

H. Lai, X. Liu, I. L. Iong, S. Yao, Y. Chen, P. Shen, H. Yu, H. Zhang, X. Zhang, Y. Dong 等, “AutoWebGLM: 基于大型语言模型的网页导航代理,” 载于第 30 届 ACM 知识发现与数据挖掘会议论文集，2024，页 5295-5306。3, 25, 26

[127] M. Fereidouni, A. Mosharrof, and A. Siddique, ”Grounded language agent for product search via intelligent web interactions,” in Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U). Association for Computational Linguistics, 2024, p. 63-75. [Online]. Available: http://dx.doi.org/10.18653/v1/2024.customnlp4u-1.73, 25, 26

M. Fereidouni, A. Mosharrof, 和 A. Siddique, “基于智能网页交互的产品搜索的有根语言代理, ”载于首届定制化自然语言处理研讨会论文集: 面向领域、应用、群体或个人的 NLP 定制进展与挑战 (CustomNLP4U)。计算语言学协会，2024，页 63-75。[在线]。可访问:http://dx.doi.org/10.18653/v1/2024.customnlp4u-1.73，25, 26

[128] R. Niu, J. Li, S. Wang, Y. Fu, X. Hu, X. Leng, H. Kong, Y. Chang, and Q. Wang, ”Screenagent: A vision language model-driven computer control agent,” arXiv preprint arXiv:2402.07945, 2024. 3, 25, 26

R. Niu, J. Li, S. Wang, Y. Fu, X. Hu, X. Leng, H. Kong, Y. Chang, 和 Q. Wang, “ScreenAgent: 一种视觉语言模型驱动的计算机控制代理,” arXiv 预印本 arXiv:2402.07945, 2024. 3, 25, 26

[129] D. Gao, L. Ji, Z. Bai, M. Ouyang, P. Li, D. Mao, Q. Wu, W. Zhang, P. Wang, X. Guo et al., ”Assistgui: Task-oriented desktop graphical user interface automation,” arXiv preprint arXiv:2312.13108, 2023. 3, 26

D. Gao, L. Ji, Z. Bai, M. Ouyang, P. Li, D. Mao, Q. Wu, W. Zhang, P. Wang, X. Guo 等, “AssistGUI: 面向任务的桌面图形用户界面自动化,” arXiv 预印本 arXiv:2312.13108, 2023. 3, 26

[130] B. Deka, Z. Huang, C. Franzen, J. Hibschman, D. Afergan, Y. Li, J. Nichols, and R. Kumar, ”Rico: A mobile app dataset for building data-driven design applications,” in Proceedings of the 30th annual ACM symposium on user interface software and technology, 2017, pp. 845-854. 3, 27, 28

B. Deka, Z. Huang, C. Franzen, J. Hibschman, D. Afergan, Y. Li, J. Nichols, 和 R. Kumar, “RICO: 用于构建数据驱动设计应用的移动应用数据集,” 载于第 30 届 ACM 用户界面软件与技术年会论文集，2017，页 845-854。3, 27, 28

[131] S. Sunkara, M. Wang, L. Liu, G. Baechler, Y.-C. Hsiao, A. Sharma, J. Stout et al., ”Towards better semantic understanding of mobile interfaces,” arXiv preprint arXiv:2210.02663, 2022. 3, 27

S. Sunkara, M. Wang, L. Liu, G. Baechler, Y.-C. Hsiao, A. Sharma, J. Stout 等，“迈向更好的移动界面语义理解，”arXiv 预印本 arXiv:2210.02663, 2022. 3, 27

[132] Y. Li, J. He, X. Zhou, Y. Zhang, and J. Baldridge, ”Mapping natural language instructions to mobile ui action sequences,” arXiv preprint arXiv:2005.03776, 2020. 3, 9, 27, 28

Y. Li, J. He, X. Zhou, Y. Zhang, 和 J. Baldridge，“将自然语言指令映射到移动 UI 动作序列，”arXiv 预印本 arXiv:2005.03776, 2020. 3, 9, 27, 28

[133] A. Burns, D. Arsan, S. Agrawal, R. Kumar, K. Saenko, and B. A. Plummer, ”Mobile app tasks with iterative feedback (motif): Addressing task feasibility in interactive visual environments,” arXiv preprint arXiv:2104.08560, 2021. 3, 9, 27, 28

A. Burns, D. Arsan, S. Agrawal, R. Kumar, K. Saenko, 和 B. A. Plummer, “带迭代反馈的移动应用任务 (MOTIF): 解决交互式视觉环境中的任务可行性问题，”arXiv 预印本 arXiv:2104.08560, 2021. 3, 9, 27, 28

[134] C. Bai, X. Zang, Y. Xu, S. Sunkara, A. Rastogi, J. Chen et al., ”Uibert: Learning generic multimodal representations for ui understanding,” arXiv preprint arXiv:2107.13731, 2021. 3, 9, 27, 28

C. Bai, X. Zang, Y. Xu, S. Sunkara, A. Rastogi, J. Chen 等，“Uibert: 学习通用多模态表示以理解用户界面”，arXiv 预印本 arXiv:2107.13731，2021。3, 9, 27, 28

[135] L. Sun, X. Chen, L. Chen, T. Dai, Z. Zhu, and K. Yu, ”Meta-gui: Towards multi-modal conversational agents on mobile gui,” arXiv preprint arXiv:2205.11029, 2022. 3, 27, 28

L. Sun, X. Chen, L. Chen, T. Dai, Z. Zhu, 和 K. Yu，“Meta-gui: 面向移动 GUI 的多模态对话代理”，arXiv 预印本 arXiv:2205.11029，2022。3, 27, 28

[136] S. G. Venkatesh, P. Talukdar, and S. Narayanan, ”Ugif: Ui grounded instruction following,” arXiv preprint arXiv:2211.07615, 2022. 3, 27, 28

S. G. Venkatesh, P. Talukdar, 和 S. Narayanan，“Ugif: 基于用户界面的指令跟随”，arXiv 预印本 arXiv:2211.07615，2022。3, 27, 28

[137] C. Rawles, A. Li, D. Rodriguez, O. Riva, and T. Lillicrap, ”An-droidinthewild: A large-scale dataset for android device control,” Advances in Neural Information Processing Systems, vol. 36, 2024. 3, 27, 28, 31, 32

C. Rawles, A. Li, D. Rodriguez, O. Riva, 和 T. Lillicrap，“An-droidinthewild: 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》(Advances in Neural Information Processing Systems)，第 36 卷，2024。3, 27, 28, 31, 32

[138] J. Zhang, J. Wu, Y. Teng, M. Liao, N. Xu, X. Xiao, Z. Wei, and D. Tang, ”Android in the zoo: Chain-of-action-thought for gui agents,” arXiv preprint arXiv:2403.02713, 2024. 3, 27, 28, 31

J. Zhang, J. Wu, Y. Teng, M. Liao, N. Xu, X. Xiao, Z. Wei, 和 D. Tang，“Android in the zoo: 面向 GUI 代理的链式行动思维”，arXiv 预印本 arXiv:2403.02713，2024。3, 27, 28, 31

[139] D. Chen, Y. Huang, S. Wu, J. Tang, L. Chen, Y. Bai, Z. He, C. Wang, H. Zhou, Y. Li et al., ”Gui-world: A dataset for gui-oriented multimodal llm-based agents,” arXiv preprint arXiv:2406.10819, 2024. 3, 27

D. Chen, Y. Huang, S. Wu, J. Tang, L. Chen, Y. Bai, Z. He, C. Wang, H. Zhou, Y. Li 等，“Gui-world: 面向 GUI 的多模态大语言模型 (LLM) 代理数据集”，arXiv 预印本 arXiv:2406.10819，2024。3, 27

[140] W. Li, W. Bishop, A. Li, C. Rawles, F. Campbell-Ajala, D. Tyama-gundlu, and O. Riva, ”On the effects of data scale on computer control agents,” arXiv preprint arXiv:2406.03679, 2024. 3, 6, 7, 27, 28, 31

W. Li, W. Bishop, A. Li, C. Rawles, F. Campbell-Ajala, D. Tyama-gundlu, 和 O. Riva，“数据规模对计算机控制代理影响的研究”，arXiv 预印本 arXiv:2406.03679，2024。3, 6, 7, 27, 28, 31

[141] Y. Chai, S. Huang, Y. Niu, H. Xiao, L. Liu, D. Zhang, P. Gao, S. Ren, and H. Li, ”Amex: Android multi-annotation expo dataset for mobile gui agents,” arXiv preprint arXiv:2407.17490, 2024. 3, 27, 28

Y. Chai, S. Huang, Y. Niu, H. Xiao, L. Liu, D. Zhang, P. Gao, S. Ren, 和 H. Li，“Amex: 面向移动 GUI 代理的安卓多注释展示数据集”，arXiv 预印本 arXiv:2407.17490，2024。3, 27, 28

[142] L. Gao, L. Zhang, S. Wang, S. Wang, Y. Li, and M. Xu, ”Mo-bileviews: A large-scale mobile gui dataset,” arXiv preprint arXiv:2409.14337, 2024. 3, 27, 28

L. Gao, L. Zhang, S. Wang, S. Wang, Y. Li, 和 M. Xu，“Mobileviews: 大规模移动 GUI 数据集”，arXiv 预印本 arXiv:2409.14337，2024。3, 27, 28

[143] D. Zhang, L. Chen, and K. Yu, ”Mobile-env: A universal platform for training and evaluation of mobile interaction,” arXiv preprint arXiv:2305.08144, 2023. 3, 29, 31

D. Zhang, L. Chen, 和 K. Yu，“Mobile-env: 用于移动交互训练与评估的通用平台”，arXiv 预印本 arXiv:2305.08144，2023。3, 29, 31

[144] M. Xing, R. Zhang, H. Xue, Q. Chen, F. Yang, and Z. Xiao, ”Understanding the weakness of large language model agents within a complex android environment,” in Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6061-6072. 3, 29, 30, 31

M. Xing, R. Zhang, H. Xue, Q. Chen, F. Yang, 和 Z. Xiao，“理解复杂安卓环境下大型语言模型代理的弱点”，发表于第 30 届 ACM 知识发现与数据挖掘会议 (KDD) 论文集，2024，页 6061-6072。3, 29, 30, 31

[145] L. Zhang, S. Wang, X. Jia, Z. Zheng, Y. Yan, L. Gao, Y. Li, and M. Xu, ”Llamatouch: A faithful and scalable testbed for mobile ui automation task evaluation,” arXiv preprint arXiv:2404.16054, 2024. 3, 5, 29, 30, 31

L. Zhang, S. Wang, X. Jia, Z. Zheng, Y. Yan, L. Gao, Y. Li, 和 M. Xu，“Llamatouch: 一个真实且可扩展的移动 UI 自动化任务评测平台”，arXiv 预印本 arXiv:2404.16054，2024。3, 5, 29, 30, 31

[146] J. Lee, T. Min, M. An, D. Hahm, H. Lee, C. Kim, and K. Lee, ”Benchmarking mobile device control agents across diverse configurations,” arXiv preprint arXiv:2404.16660, 2024. 3, 29

J. Lee, T. Min, M. An, D. Hahm, H. Lee, C. Kim, 和 K. Lee，“跨多种配置的移动设备控制代理基准测试”，arXiv 预印本 arXiv:2404.16660，2024。3, 29

[147] C. Rawles, S. Clinckemaillie, Y. Chang, J. Waltz, G. Lau, M. Fair, A. Li, W. Bishop, W. Li, F. Campbell-Ajala et al., ”Androidworld: A dynamic benchmarking environment for autonomous agents,” arXiv preprint arXiv:2405.14573, 2024. 3, 29, 30

C. Rawles, S. Clinckemaillie, Y. Chang, J. Waltz, G. Lau, M. Fair, A. Li, W. Bishop, W. Li, F. Campbell-Ajala 等，“Androidworld: 一个用于自主代理的动态基准测试环境，”arXiv 预印本 arXiv:2405.14573, 2024. 3, 29, 30

[148] Y. Hu, X. Wang, Y. Wang, Y. Zhang, S. Guo, C. Chen, X. Wang, and Y. Zhou, ”Auitestagent: Automatic requirements oriented gui function testing,” arXiv preprint arXiv:2407.09018, 2024. 3, 29, 30

Y. Hu, X. Wang, Y. Wang, Y. Zhang, S. Guo, C. Chen, X. Wang, 和 Y. Zhou, “Auitestagent: 面向需求的自动化 GUI 功能测试，”arXiv 预印本 arXiv:2407.09018, 2024. 3, 29, 30

[149] L. Zheng, Z. Huang, Z. Xue, X. Wang, B. An, and S. Yan, ”Agentstudio: A toolkit for building general virtual agents,” arXiv preprint arXiv:2403.17918, 2024. 3, 29

L. Zheng, Z. Huang, Z. Xue, X. Wang, B. An, 和 S. Yan, “Agentstudio: 构建通用虚拟代理的工具包，”arXiv 预印本 arXiv:2403.17918, 2024. 3, 29

[150] Y. Xu, X. Liu, X. Sun, S. Cheng, H. Yu, H. Lai, S. Zhang, D. Zhang, J. Tang, and Y. Dong, ”Androidlab: Training and systematic benchmarking of android autonomous agents,” arXiv preprint arXiv:2410.24024, 2024. 3, 29, 30, 32

Y. Xu, X. Liu, X. Sun, S. Cheng, H. Yu, H. Lai, S. Zhang, D. Zhang, J. Tang, 和 Y. Dong, “Androidlab: 安卓自主代理的训练与系统基准测试，”arXiv 预印本 arXiv:2410.24024, 2024. 3, 29, 30, 32

[151] L. Wang, Y. Deng, Y. Zha, G. Mao, Q. Wang, T. Min, W. Chen, and S. Chen, ”Mobileagentbench: An efficient and user-friendly benchmark for mobile llm agents,” arXiv preprint arXiv:2406.08184, 2024. 3, 29, 30, 32

L. Wang, Y. Deng, Y. Zha, G. Mao, Q. Wang, T. Min, W. Chen, 和 S. Chen, “Mobileagentbench: 高效且用户友好的移动大语言模型代理基准，”arXiv 预印本 arXiv:2406.08184, 2024. 3, 29, 30, 32

[152] X. Liu, T. Zhang, Y. Gu, I. L. Iong, Y. Xu, X. Song, S. Zhang, H. Lai, X. Liu, H. Zhao et al., ”Visualagentbench: Towards large multimodal models as visual foundation agents,” arXiv preprint arXiv:2408.06327, 2024. 3, 29, 30

X. Liu, T. Zhang, Y. Gu, I. L. Iong, Y. Xu, X. Song, S. Zhang, H. Lai, X. Liu, H. Zhao 等, “Visualagentbench: 迈向大型多模态模型作为视觉基础代理,” arXiv 预印本 arXiv:2408.06327, 2024. 3, 29, 30

[153] W. Wang, Z. Yu, R. Ye, J. Zhang, S. Chen, and Y. Wang, ”Fedmabench: Benchmarking mobile agents on decentralized heterogeneous user data,” arXiv preprint arXiv:2503.05143, 2025. 3, 32

W. Wang, Z. Yu, R. Ye, J. Zhang, S. Chen, 和 Y. Wang, “Fedmabench: 基于去中心化异构用户数据的移动代理基准测试,” arXiv 预印本 arXiv:2503.05143, 2025. 3, 32

[154] J. Sun, Z. Hua, and Y. Xia, ”Autoeval: A practical framework for autonomous evaluation of mobile agents,” arXiv preprint arXiv:2503.02403, 2025. 3, 29, 30

J. Sun, Z. Hua, 和 Y. Xia, “Autoeval: 移动代理自主评估的实用框架,” arXiv 预印本 arXiv:2503.02403, 2025. 3, 29, 30

[155] G. Liu, P. Zhao, L. Liu, Z. Chen, Y. Chai, S. Ren, H. Wang, S. He, and W. Meng, ”Learnact: Few-shot mobile gui agent with a unified demonstration benchmark,” arXiv preprint arXiv:2504.13805, 2025. 3, 11, 19, 29, 30, 31

G. Liu, P. Zhao, L. Liu, Z. Chen, Y. Chai, S. Ren, H. Wang, S. He, 和 W. Meng, ”Learnact: 具有统一演示基准的少样本移动 GUI 代理,” arXiv 预印本 arXiv:2504.13805, 2025. 3, 11, 19, 29, 30, 31

[156] Y. Chai, H. Li, J. Zhang, L. Liu, G. Wang, S. Ren, S. Huang, and H. Li, ”A3: Android agent arena for mobile gui agents,” arXiv preprint arXiv:2501.01149, 2025. 3, 29, 30

Y. Chai, H. Li, J. Zhang, L. Liu, G. Wang, S. Ren, S. Huang, 和 H. Li, ”A3: 用于移动 GUI 代理的 Android 代理竞技场,” arXiv 预印本 arXiv:2501.01149, 2025. 3, 29, 30

[157] B. Wu, Y. Li, M. Fang, Z. Song, Z. Zhang, Y. Wei, and L. Chen, ”Foundations and recent trends in multimodal mobile agents: A survey,” arXiv preprint arXiv:2411.02006, 2024. 2

B. Wu, Y. Li, M. Fang, Z. Song, Z. Zhang, Y. Wei, 和 L. Chen, ” 多模态移动代理的基础与最新趋势: 综述,” arXiv 预印本 arXiv:2411.02006, 2024. 2

[158] S. Wang, W. Liu, J. Chen, W. Gan, X. Zeng, S. Yu, X. Hao, K. Shao, Y. Wang, and R. Tang, ”Gui agents with foundation models: A comprehensive survey,” arXiv preprint arXiv:2411.04890, 2024. 2

S. Wang, W. Liu, J. Chen, W. Gan, X. Zeng, S. Yu, X. Hao, K. Shao, Y. Wang, 和 R. Tang, ” 基于基础模型的 GUI 代理: 全面综述,” arXiv 预印本 arXiv:2411.04890, 2024. 2

[159] C. Zhang, S. He, J. Qian, B. Li, L. Li, S. Qin, Y. Kang, M. Ma, Q. Lin, S. Rajmohan et al., ”Large language model-brained gui agents: A survey,” arXiv preprint arXiv:2411.18279, 2024. 2

C. Zhang, S. He, J. Qian, B. Li, L. Li, S. Qin, Y. Kang, M. Ma, Q. Lin, S. Rajmohan 等, ” 大型语言模型驱动的 GUI 代理: 综述,” arXiv 预印本 arXiv:2411.18279, 2024. 2

[160] P. Kong, L. Li, J. Gao, K. Liu, T. F. Bissyandé, and J. Klein, ”Automated testing of android apps: A systematic literature review,” IEEE Transactions on Reliability, vol. 68, no. 1, pp. 45- 66, 2018. 3, 4

P. Kong, L. Li, J. Gao, K. Liu, T. F. Bissyandé, 和 J. Klein, ” 安卓应用自动化测试: 系统文献综述,” IEEE 可靠性学报, 第 68 卷，第 1 期，页 45-66, 2018. 3, 4

[161] B. Kirubakaran and V. Karthikeyani, ”Mobile application testing-challenges and solution approach through automation,” in 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering. IEEE, 2013, pp. 79-84. 3, 4

B. Kirubakaran 和 V. Karthikeyani, ” 移动应用测试——通过自动化解决挑战与方案,” 2013 年国际模式识别、信息学与移动工程会议论文集. IEEE, 2013, 页 79-84. 3, 4

[162] D. Amalfitano, A. R. Fasolino, P. Tramontana, B. D. Ta, and A. M. Memon, ”Mobiguitar: Automated model-based testing of mobile apps,” IEEE software, vol. 32, no. 5, pp. 53-59, 2014. 3, 4

D. Amalfitano, A. R. Fasolino, P. Tramontana, B. D. Ta, 和 A. M. Memon, ”MobiGuitar: 基于模型的移动应用自动化测试,” IEEE 软件, 第 32 卷，第 5 期，页 53-59, 2014. 3, 4

[163] M. Linares-Vásquez, K. Moran, and D. Poshyvanyk, ”Continuous, evolutionary and large-scale: A new perspective for automated mobile app testing,” in 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2017, pp. 399-410. 3, 4

M. Linares-Vásquez, K. Moran, 和 D. Poshyvanyk, ” 持续、进化与大规模: 移动应用自动化测试的新视角,” 2017 年 IEEE 软件维护与演进国际会议 (ICSME) 论文集. IEEE, 2017, 页 399-410. 3, 4

[164] Y. Zhao, B. Harrison, and T. Yu, ”Dinodroid: Testing android apps using deep q-networks,” ACM Transactions on Software Engineering and Methodology, vol. 33, no. 5, pp. 1-24, 2024. 3, 4

Y. Zhao, B. Harrison, 和 T. Yu, ”DinoDroid: 基于深度 Q 网络的安卓应用测试,” ACM 软件工程与方法学汇刊, 第 33 卷，第 5 期，页 1-24, 2024. 3, 4

[165] G. Hecht, O. Benomar, R. Rouvoy, N. Moha, and L. Duchien, ”Tracking the software quality of android applications along their evolution (t),” in 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2015, pp. 236-247. 4

G. Hecht, O. Benomar, R. Rouvoy, N. Moha, 和 L. Duchien, ” 安卓应用软件质量演进跟踪 (T),” 2015 年第 30 届 IEEE/ACM 自动化软件工程国际会议 (ASE) 论文集. IEEE, 2015, 页 236-247. 4

[166] S. Zein, N. Salleh, and J. Grundy, ”A systematic mapping study of mobile application testing techniques,” Journal of Systems and Software, vol. 117, pp. 334-356, 2016. 4

S. Zein, N. Salleh, 和 J. Grundy, ” 移动应用测试技术的系统映射研究,” 系统与软件学报, 第 117 卷，页 334-356, 2016. 4

[167] C. S. Jensen, M. R. Prasad, and A. Møller, ”Automated testing with targeted event sequence generation,” in Proceedings of the 2013 International Symposium on Software Testing and Analysis, 2013, pp. 67-77. 4

C. S. Jensen, M. R. Prasad, 和 A. Møller, ” 基于目标事件序列生成的自动化测试,” 2013 年国际软件测试与分析研讨会论文集, 2013, 页 67-77. 4

[168] A. Machiry, R. Tahiliani, and M. Naik, ”Dynodroid: An input generation system for android apps,” in Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, 2013, pp. 224-234. 4

A. Machiry, R. Tahiliani, 和 M. Naik, ”Dynodroid: 安卓应用输入生成系统,” 2013 年第九届软件工程基础联合会议论文集, 2013, 页 224-234. 4

[169] D. Amalfitano, A. R. Fasolino, P. Tramontana, S. De Carmine, and A. M. Memon, ”Using gui ripping for automated testing of android applications,” in Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, 2012, pp. 258-261. 4

D. Amalfitano, A. R. Fasolino, P. Tramontana, S. De Carmine, 和 A. M. Memon, “使用 GUI 剥离进行安卓应用的自动化测试,”载于第 27 届 IEEE/ACM 自动化软件工程国际会议论文集, 2012, 页 258-261. 4

[170] P. Ladosz, L. Weng, M. Kim, and H. Oh, ”Exploration in deep reinforcement learning: A survey,” Information Fusion, vol. 85, pp. 1-22, 2022. 4

P. Ladosz, L. Weng, M. Kim, 和 H. Oh, “深度强化学习中的探索: 综述,”信息融合, 第 85 卷, 页 1-22, 2022. 4

[171] J. Fan, Z. Wang, Y. Xie, and Z. Yang, ”A theoretical analysis of deep q-learning,” in Learning for dynamics and control. PMLR, 2020, pp. 486-489. 4

J. Fan, Z. Wang, Y. Xie, 和 Z. Yang, “深度 Q 学习的理论分析,”载于《动力学与控制学习》. PMLR, 2020, 页 486-489. 4

[172] F.-M. Luo, T. Xu, H. Lai, X.-H. Chen, W. Zhang, and Y. Yu, ”A survey on model-based reinforcement learning,” Science China Information Sciences, vol. 67, no. 2, p. 121101, 2024. 4

F.-M. Luo, T. Xu, H. Lai, X.-H. Chen, W. Zhang, 和 Y. Yu, “基于模型的强化学习综述,”中国科学信息科学, 第 67 卷第 2 期, 文章编号 121101, 2024. 4

[173] R. Bridle and E. McCreath, ”Inducing shortcuts on a mobile phone interface,” in Proceedings of the 11th international conference on Intelligent user interfaces, 2006, pp. 327-329. 4

R. Bridle 和 E. McCreath, “在手机界面上引入快捷方式,”载于第 11 届智能用户界面国际会议论文集, 2006, 页 327-329. 4

[174] T. Guerreiro, R. Gamboa, and J. Jorge, ”Mnemonical body shortcuts: improving mobile interaction,” in Proceedings of the 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction, 2008, pp. 1-8. 4

T. Guerreiro, R. Gamboa, 和 J. Jorge, “助记身体快捷方式: 提升移动交互体验,”载于第 15 届欧洲认知工效学会议论文集: 酷交互的人体工学, 2008, 页 1-8. 4

[175] C. Kennedy and S. E. Everett, ”Use of cognitive shortcuts in landline and cell phone surveys,” Public Opinion Quarterly, vol. 75, no. 2, pp. 336-348, 2011. 4

C. Kennedy 和 S. E. Everett, “固定电话和手机调查中认知捷径的使用,”舆论季刊, 第 75 卷第 2 期, 页 336-348, 2011. 4

[176] S. Agostinelli, A. Marrella, and M. Mecella, ”Research challenges for intelligent robotic process automation,” in Business Process Management Workshops: BPM 2019 International Workshops, Vienna, Austria, September 1-6, 2019, Revised Selected Papers 17. Springer, 2019, pp. 12-18. 4

S. Agostinelli, A. Marrella, 和 M. Mecella, “智能机器人流程自动化的研究挑战,”载于业务流程管理研讨会:BPM 2019 国际研讨会论文集, 奥地利维也纳, 2019 年 9 月 1-6 日, 修订精选论文 17. Springer, 2019, 页 12-18. 4

[177] D. Pramod, ”Robotic process automation for industry: adoption status, benefits, challenges and research agenda,” Benchmarking: an international journal, vol. 29, no. 5, pp. 1562-1586, 2022. 4

D. Pramod, “工业机器人流程自动化: 采用现状、收益、挑战及研究议程,”国际基准测试期刊, 第 29 卷第 5 期, 页 1562-1586, 2022. 4

[178] R. Syed, S. Suriadi, M. Adams, W. Bandara, S. J. Leemans, C. Ouyang, A. H. Ter Hofstede, I. Van De Weerd, M. T. Wynn, and H. A. Reijers, ”Robotic process automation: contemporary themes and challenges,” Computers in Industry, vol. 115, p. 103162, 2020. 4

R. Syed, S. Suriadi, M. Adams, W. Bandara, S. J. Leemans, C. Ouyang, A. H. Ter Hofstede, I. Van De Weerd, M. T. Wynn, 和 H. A. Reijers, “机器人流程自动化: 当代主题与挑战,”工业计算机, 第 115 卷, 文章编号 103162, 2020. 4

[179] J. Clarke, J. Proudfoot, A. Whitton, M.-R. Birch, M. Boyd, G. Parker, V. Manicavasagar, D. Hadzi-Pavlovic, A. Fogarty et al., ”Therapeutic alliance with a fully automated mobile phone and web-based intervention: secondary analysis of a randomized controlled trial,” JMIR mental health, vol. 3, no. 1, p. e4656, 2016. 4

J. Clarke, J. Proudfoot, A. Whitton, M.-R. Birch, M. Boyd, G. Parker, V. Manicavasagar, D. Hadzi-Pavlovic, A. Fogarty 等, “与全自动手机及网络干预的治疗联盟: 随机对照试验的二次分析,”JMIR 心理健康, 第 3 卷第 1 期, 文章 e4656, 2016. 4

[180] T. J.-J. Li, A. Azaria, and B. A. Myers, ”Sugilite: creating multimodal smartphone automation by demonstration,” in Proceedings of the 2017 CHI conference on human factors in computing systems, 2017, pp. 6038-6049. 4

T. J.-J. Li, A. Azaria, 和 B. A. Myers, “Sugilite: 通过演示创建多模态智能手机自动化,”载于 2017 年 CHI 人机交互大会论文集, 2017, 页 6038-6049. 4

[181] S. M. Patel and S. J. Pasha, ”Home automation system (has) using android for mobile phone,” International Journal Of Scientific Engeneering and Technology Research, ISSN, pp. 2319-8885, 2015. 4

S. M. Patel 和 S. J. Pasha, “基于安卓手机的家庭自动化系统 (HAS),”国际科学工程技术研究期刊, ISSN, 页 2319-8885, 2015. 4

[182] M. Asadullah and A. Raza, ”An overview of home automation systems,” in 2016 2nd international conference on robotics and artificial intelligence (ICRAI). IEEE, 2016, pp. 27-31. 4

M. Asadullah 和 A. Raza, “家庭自动化系统概述,”载于 2016 年第 2 届机器人与人工智能国际会议 (ICRAI). IEEE, 2016, 页 27-31. 4

[183] R. Majeed, N. A. Abdullah, I. Ashraf, Y. B. Zikria, M. F. Mushtaq, and M. Umer, ”An intelligent, secure, and smart home automation system,” Scientific Programming, vol. 2020, no. 1, p. 4579291, 2020. 4

R. Majeed, N. A. Abdullah, I. Ashraf, Y. B. Zikria, M. F. Mushtaq, 和 M. Umer, “一种智能、安全且智能的家居自动化系统,”《科学编程》(Scientific Programming), 2020 年第 1 期, 页 4579291, 2020 年. 4

[184] X. Liu, Y. Shi, C. Yu, C. Gao, T. Yang, C. Liang, and Y. Shi, ”Understanding in-situ programming for smart home automation,” Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 7, no. 2, pp. 1-31, 2023. 4

X. Liu, Y. Shi, C. Yu, C. Gao, T. Yang, C. Liang, 和 Y. Shi,“理解智能家居自动化的现场编程,”《ACM 交互式、移动、可穿戴及普适技术会议论文集》(Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies), 第 7 卷第 2 期, 页 1-31, 2023 年. 4

[185] R. K. Kodali, S. C. Rajanarayanan, L. Boppana, S. Sharma, and A. Kumar, ”Low cost smart home automation system using smart phone,” in 2019 IEEE R10 humanitarian technology conference (R10- HTC)(47129). IEEE, 2019, pp. 120-125. 4

R. K. Kodali, S. C. Rajanarayanan, L. Boppana, S. Sharma, 和 A. Kumar, “基于智能手机的低成本智能家居自动化系统,”载于 2019 年 IEEE R10 人道技术会议 (R10-HTC)(47129). IEEE, 2019 年, 页 120-125. 4

[186] R. K. Kodali and K. S. Mahesh, ”Low cost implementation of smart home automation,” in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017, pp. 461-466. 4

R. K. Kodali 和 K. S. Mahesh, “低成本智能家居自动化的实现,”载于 2017 年计算、通信与信息学进展国际会议 (ICACCI). IEEE, 2017 年, 页 461-466. 4

[187] S. Moreira, H. S. Mamede, and A. Santos, ”Process automation using rpa-a literature review,” Procedia Computer Science, vol. 219, pp. 244-254, 2023. 4

S. Moreira, H. S. Mamede, 和 A. Santos, “基于机器人流程自动化 (RPA) 的流程自动化——文献综述,”《计算机科学学报》(Procedia Computer Science), 第 219 卷, 页 244-254, 2023 年. 4

[188] C. Lamberton, D. Brigo, and D. Hoy, ”Impact of robotics, rpa and ai on the insurance industry: challenges and opportunities,” Journal of Financial Perspectives, vol. 4, no. 1, 2017. 4

C. Lamberton, D. Brigo, 和 D. Hoy, “机器人技术、机器人流程自动化 (RPA) 与人工智能 (AI) 对保险行业的影响: 挑战与机遇,”《金融视角杂志》(Journal of Financial Perspectives), 第 4 卷第 1 期, 2017 年. 4

[189] A. Meironke and S. Kuehnel, ”How to measure rpa’s benefits? a review on metrics, indicators, and evaluation methods of rpa benefit assessment,” 2022. 4

A. Meironke 和 S. Kuehnel, “如何衡量机器人流程自动化 (RPA) 的效益? 关于 RPA 效益评估的指标、指标体系及评价方法的综述,”2022 年. 4

[190] A. M. Tripathi, Learning Robotic Process Automation: Create Software robots and automate business processes with the leading RPA tool-UiPath. Packt Publishing Ltd, 2018. 5

A. M. Tripathi, 《学习机器人流程自动化: 使用领先的 RPA 工具 UiPath 创建软件机器人并自动化业务流程》. Packt Publishing Ltd, 2018 年. 5

[191] X. Ling, M. Gao, and D. Wang, ”Intelligent document processing based on rpa and machine learning,” in 2020 Chinese Automation Congress (CAC). IEEE, 2020, pp. 1349-1353. 5

X. Ling, M. Gao, 和 D. Wang, “基于机器人流程自动化 (RPA) 和机器学习的智能文档处理,”载于 2020 年中国自动化大会 (CAC). IEEE, 2020 年, 页 1349-1353. 5

[192] S. Agostinelli, M. Lupia, A. Marrella, and M. Mecella, ”Reactive synthesis of software robots in rpa from user interface logs,” Computers in Industry, vol. 142, p. 103721, 2022. 5

S. Agostinelli, M. Lupia, A. Marrella, 和 M. Mecella, “基于用户界面日志的机器人流程自动化 (RPA) 软件机器人反应式合成,”《工业计算机》(Computers in Industry), 第 142 卷, 页 103721, 2022 年. 5

[193] H. V. Le, S. Mayer, M. Weiß, J. Vogelsang, H. Weingärtner, and N. Henze, ”Shortcut gestures for mobile text editing on fully touch sensitive smartphones,” ACM Transactions on Computer-Human Interaction (TOCHI), vol. 27, no. 5, pp. 1-38, 2020. 5

H. V. Le, S. Mayer, M. Weiß, J. Vogelsang, H. Weingärtner, 和 N. Henze, “全触控智能手机上的移动文本编辑快捷手势,”《ACM 人机交互汇刊》(ACM Transactions on Computer-Human Interaction, TOCHI), 第 27 卷第 5 期, 页 1-38, 2020 年. 5

[194] A. M. Roffarello, A. K. Purohit, and S. V. Purohit, ”Trigger-action programming for wellbeing: Insights from 6590 ios shortcuts,” IEEE Pervasive Computing, 2024. 5

A. M. Roffarello, A. K. Purohit, 和 S. V. Purohit, “面向健康的触发-动作编程: 基于 6590 个 iOS 快捷指令的洞察,”《IEEE 普适计算》(IEEE Pervasive Computing), 2024 年. 5

[195] V. Kepuska and G. Bohouta, ”Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home),” in 2018 IEEE 8th annual computing and communication workshop and

conference (CCWC). IEEE, 2018, pp. 99-103. 5

V. Kepuska 和 G. Bohouta, “下一代虚拟个人助理 (微软 Cortana、苹果 Siri、亚马逊 Alexa 和谷歌 Home),”载于 2018 年第 8 届 IEEE 年度计算与通信研讨会及会议 (CCWC). IEEE, 2018 年, 页 99-103. 5

[196] B. R. Cowan, N. Pantidi, D. Coyle, K. Morrissey, P. Clarke, S. Al-Shehri, D. Earley, and N. Bandeira, ”” what can i help you with?” infrequent users’ experiences of intelligent personal assistants,” in Proceedings of the 19th international conference on human-computer interaction with mobile devices and services, 2017, pp. 1-12. 5

B. R. Cowan, N. Pantidi, D. Coyle, K. Morrissey, P. Clarke, S. Al-Shehri, D. Earley, 和 N. Bandeira, “‘我能帮您做什么?’不常用智能个人助理用户的体验,”载于第 19 届国际移动设备与服务人机交互会议论文集, 2017 年, 页 1-12. 5

[197] D. Anicic, P. Fodor, S. Rudolph, R. Stühmer, N. Stojanovic, and R. Studer, ”A rule-based language for complex event processing and reasoning,” in Web Reasoning and Rule Systems: Fourth International Conference, RR 2010, Bressanone/Brixen, Italy, September 22-24, 2010. Proceedings 4. Springer, 2010, pp. 42-57. 5

D. Anicic, P. Fodor, S. Rudolph, R. Stühmer, N. Stojanovic, 和 R. Studer, “一种用于复杂事件处理和推理的基于规则的语言,”载于《网络推理与规则系统: 第四届国际会议，RR 2010，意大利布雷萨诺内/布里克森，2010 年 9 月 22-24 日》会议论文集 4，施普林格，2010 年，第 42-57 页。5

[198] N. Kang, B. Singh, Z. Afzal, E. M. van Mulligen, and J. A. Kors, ”Using rule-based natural language processing to improve disease normalization in biomedical text,” Journal of the American Medical Informatics Association, vol. 20, no. 5, pp. 876-881, 2013. 5

N. Kang, B. Singh, Z. Afzal, E. M. van Mulligen, 和 J. A. Kors, “利用基于规则的自然语言处理改进生物医学文本中的疾病标准化,”《美国医学信息学会杂志》(Journal of the American Medical Informatics Association), 第 20 卷，第 5 期，第 876-881 页，2013 年。5

[199] N. Karanikolas, E. Manga, N. Samaridi, E. Tousidou, and M. Vas-silakopoulos, ”Large language models versus natural language understanding and generation,” in Proceedings of the 27th PanHellenic Conference on Progress in Computing and Informatics, 2023, pp. 278-290. 5, 7

N. Karanikolas, E. Manga, N. Samaridi, E. Tousidou, 和 M. Vassilakopoulos, “大型语言模型与自然语言理解及生成的比较,”载于《第 27 届泛希腊计算与信息学进展会议论文集》，2023 年，第 278-290 页。5, 7

[200] J. Fu, X. Zhang, Y. Wang, W. Zeng, and N. Zheng, ”Understanding mobile gui: From pixel-words to screen-sentences,” Neurocomput-ing, vol. 601, p. 128200, 2024. 5

J. Fu, X. Zhang, Y. Wang, W. Zeng, 和 N. Zheng, “理解移动图形用户界面: 从像素词到屏幕句子,”《神经计算》(Neurocomputing), 第 601 卷，文章编号 128200，2024 年。5

[201] I. Banerjee, B. Nguyen, V. Garousi, and A. Memon, ”Graphical user interface (gui) testing: Systematic mapping and repository,” Information and Software Technology, vol. 55, no. 10, pp. 1679-1694, 2013. 5

I. Banerjee, B. Nguyen, V. Garousi, 和 A. Memon, “图形用户界面 (GUI) 测试: 系统映射与资源库,”《信息与软件技术》(Information and Software Technology), 第 55 卷，第 10 期，第 1679-1694 页，2013 年。5

[202] C. Chen, T. Su, G. Meng, Z. Xing, and Y. Liu, ”From ui design image to gui skeleton: a neural machine translator to bootstrap mobile gui implementation,” in Proceedings of the 40th International Conference on Software Engineering, 2018, pp. 665-676. 5

C. Chen, T. Su, G. Meng, Z. Xing, 和 Y. Liu, “从用户界面设计图像到 GUI 骨架: 一种用于启动移动 GUI 实现的神经机器翻译器,”载于《第 40 届国际软件工程会议论文集》，2018 年，第 665-676 页。5

[203] J. Brich, M. Walch, M. Rietzler, M. Weber, and F. Schaub, ”Exploring end user programming needs in home automation,” ACM Transactions on Computer-Human Interaction (TOCHI), vol. 24, no. 2, pp. 1-35, 2017. 5

J. Brich, M. Walch, M. Rietzler, M. Weber, 和 F. Schaub, “探索家庭自动化中终端用户编程需求,”《ACM 计算机-人机交互汇刊》(ACM Transactions on Computer-Human Interaction, TOCHI), 第 24 卷，第 2 期，第 1-35 页，2017 年。5

[204] J. Wu, X. Zhang, J. Nichols, and J. P. Bigham, ”Screen parsing: Towards reverse engineering of ui models from screenshots,” in The 34th Annual ACM Symposium on User Interface Software and Technology, 2021, pp. 470-483. 5

J. Wu, X. Zhang, J. Nichols, 和 J. P. Bigham, “屏幕解析: 迈向从截图逆向工程用户界面模型,”载于《第 34 届 ACM 用户界面软件与技术年会》，2021 年，第 470-483 页。5

[205] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhari-wal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., ”Language models are few-shot learners,” Advances in neural information processing systems, vol. 33, pp. 1877-1901, 2020. 5, 7

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell 等, “语言模型是少样本学习者,”《神经信息处理系统进展》(Advances in Neural Information Processing Systems), 第 33 卷，第 1877-1901 页，2020 年。5, 7

[206] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, ”Scaling laws for neural language models,” arXiv preprint arXiv:2001.08361, 2020. 5

J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, 和 D. Amodei, “神经语言模型的规模定律,”arXiv 预印本 arXiv:2001.08361, 2020 年。5

[207] T. Hagendorff, ”Machine psychology: Investigating emergent capabilities and behavior in large language models using psychological methods,” arXiv preprint arXiv:2303.13988, vol. 1, 2023. 5

T. Hagendorff, “机器心理学: 利用心理学方法研究大型语言模型的涌现能力与行为,”arXiv 预印本 arXiv:2303.13988, 第 1 卷，2023 年。5

[208] A. Vaswani, ”Attention is all you need,” Advances in Neural Information Processing Systems, 2017. 7

A. Vaswani, “注意力机制即一切,”《神经信息处理系统进展》，2017 年。7

[209] A. Radford, ”Improving language understanding by generative pre-training,” 2018. 7

A. Radford, “通过生成式预训练提升语言理解,”2018 年。7

[210] J. Devlin, ”Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018. 7

J. Devlin,“BERT: 用于语言理解的深度双向变换器预训练,”arXiv 预印本 arXiv:1810.04805, 2018 年。7

[211] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong et al., ”A survey of large language models,” arXiv preprint arXiv:2303.18223, 2023. 7

W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong 等, “大型语言模型综述,”arXiv 预印本 arXiv:2303.18223, 2023 年。7

[212] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang et al., ”A survey on evaluation of large language models,” ACM Transactions on Intelligent Systems and Technology, vol. 15, no. 3, pp. 1-45, 2024. 7

Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang 等，“大型语言模型评估综述”，《ACM 智能系统与技术汇刊》，第 15 卷，第 3 期，页码 1-45，2024 年。7

[213] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, ”Large language models: A survey,” arXiv preprint arXiv:2402.06196, 2024. 7

S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, 和 J. Gao，“大型语言模型综述”，arXiv 预印本 arXiv:2402.06196，2024 年。7

[214] B. Wang, X. Yue, and H. Sun, ”Can chatgpt defend its belief in truth? evaluating llm reasoning via debate,” arXiv preprint arXiv:2305.13160, 2023. 7

B. Wang, X. Yue, 和 H. Sun，“ChatGPT 能否捍卫其对真理的信念？通过辩论评估大型语言模型推理能力”，arXiv 预印本 arXiv:2305.13160，2023 年。7

[215] L. Yuan, G. Cui, H. Wang, N. Ding, X. Wang, J. Deng, B. Shan, H. Chen, R. Xie, Y. Lin et al., ”Advancing llm reasoning generalists with preference trees,” arXiv preprint arXiv:2404.02078, 2024. 7

L. Yuan, G. Cui, H. Wang, N. Ding, X. Wang, J. Deng, B. Shan, H. Chen, R. Xie, Y. Lin 等，“通过偏好树推进大型语言模型推理通用性”，arXiv 预印本 arXiv:2404.02078，2024 年。7

[216] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, ”Llm-planner: Few-shot grounded planning for embodied agents with large language models,” in Proceedings of the IEEE/CVF International

Conference on Computer Vision, 2023, pp. 2998-3009. 7, 12

C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, 和 Y. Su,“LLM-Planner: 基于大型语言模型的少样本具身代理规划”，载于《IEEE/CVF 国际计算机视觉会议论文集》，2023 年，页码 2998-3009。7, 12

[217] K. Valmeekam, M. Marquez, S. Sreedharan, and S. Kambhampati, ”On the planning abilities of large language models-a critical investigation,” Advances in Neural Information Processing Systems, vol. 36, pp. 75993-76005, 2023. 7

K. Valmeekam, M. Marquez, S. Sreedharan, 和 S. Kambhampati，“大型语言模型的规划能力——一项批判性研究”，《神经信息处理系统进展》，第 36 卷，页码 75993-76005，2023 年。7

[218] W. Talukdar and A. Biswas, ”Improving large language model (llm) fidelity through context-aware grounding: A systematic approach to reliability and veracity,” arXiv preprint arXiv:2408.04023, 2024. 7

W. Talukdar 和 A. Biswas，“通过上下文感知的落地提升大型语言模型 (LLM) 可信度: 一种系统化的可靠性与真实性方法”，arXiv 预印本 arXiv:2408.04023，2024 年。7

[219] R. Koike, M. Kaneko, and N. Okazaki, ”Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 19, 2024, pp. 21258-21266. 7

R. Koike, M. Kaneko, 和 N. Okazaki，“Outfox: 通过对抗生成样本的上下文学习检测大型语言模型生成的论文”，载于《AAAI 人工智能会议论文集》，第 38 卷，第 19 期，2024 年，页码 21258-21266。7

[220] S. Zhang, Z. Zhang, K. Chen, X. Ma, M. Yang, T. Zhao, and M. Zhang, ”Dynamic planning for llm-based graphical user interface automation,” arXiv preprint arXiv:2410.00467, 2024. 8, 12

S. Zhang, Z. Zhang, K. Chen, X. Ma, M. Yang, T. Zhao, 和 M. Zhang，“基于大型语言模型的图形用户界面自动化动态规划”，arXiv 预印本 arXiv:2410.00467，2024 年。8, 12

[221] G. E. Monahan, ”State of the art-a survey of partially observable markov decision processes: theory, models, and algorithms,” Management science, vol. 28, no. 1, pp. 1-16, 1982. 8

G. E. Monahan，“最先进技术——部分可观测马尔可夫决策过程 (POMDP) 的理论、模型与算法综述”，《管理科学》，第 28 卷，第 1 期，页码 1-16，1982 年。8

[222] M. T. Spaan, ”Partially observable markov decision processes,” in Reinforcement learning: State-of-the-art. Springer, 2012, pp. 387-414. 8

M. T. Spaan，“部分可观测马尔可夫决策过程”，载于《强化学习: 最先进技术》，施普林格，2012 年，页码 387-414。8

[223] I. Medhi, K. Toyama, A. Joshi, U. Athavankar, and E. Cutrell, ”A comparison of list vs. hierarchical uis on mobile phones for nonliterate users,” in Human-Computer Interaction-INTERACT 2013: 14th IFIP TC 13

International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part II 14. Springer, 2013, pp. 497-504. 9

I. Medhi, K. Toyama, A. Joshi, U. Athavankar, 和 E. Cutrell，“非识字用户在手机上列表界面与层级界面的比较”，载于《人机交互-INTERACT 2013: 第 14 届 IFIP TC 13 国际会议论文集》，南非开普敦，2013 年 9 月 2-6 日，第二部分 14，施普林格，2013 年，页码 497-504。9

[224] O. J. Räsänen and J. P. Saarinen, ”Sequence prediction with sparse distributed hyperdimensional coding applied to the analysis of mobile phone use patterns,” IEEE transactions on neural networks and learning systems, vol. 27, no. 9, pp. 1878-1889, 2015. 9

O. J. Räsänen 和 J. P. Saarinen，“应用稀疏分布式超维编码的序列预测及其在手机使用模式分析中的应用”，《IEEE 神经网络与学习系统汇刊》，第 27 卷，第 9 期，页码 1878-1889，2015 年。9

[225] G. Verma, R. Kaur, N. Srishankar, Z. Zeng, T. Balch, and M. Veloso, ”Adaptagent: Adapting multimodal web agents with few-shot learning from human demonstrations,” arXiv preprint arXiv:2411.13451, 2024. 10, 11, 31

G. Verma, R. Kaur, N. Srishankar, Z. Zeng, T. Balch, 和 M. Veloso，“AdaptAgent: 通过人类示范的少样本学习适应多模态网络代理”，arXiv 预印本 arXiv:2411.13451，2024 年。10, 11, 31

[226] H. He, W. Yao, K. Ma, W. Yu, Y. Dai, H. Zhang, Z. Lan, and D. Yu, ”Webvoyager: Building an end-to-end web agent with large multimodal models,” arXiv preprint arXiv:2401.13919, 2024. 10

H. He, W. Yao, K. Ma, W. Yu, Y. Dai, H. Zhang, Z. Lan, 和 D. Yu, “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. 10

[227] B. Tang and K. G. Shin, ”Steward: Natural language web automation,” arXiv preprint arXiv:2409.15441, 2024. 10

B. Tang 和 K. G. Shin, “Steward: 自然语言网页自动化,”arXiv 预印本 arXiv:2409.15441, 2024. 10

[228] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao, ”Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v,” 2023. [Online]. Available: https://arxiv.org/abs/2310.11441 10, 17

J. Yang, H. Zhang, F. Li, X. Zou, C. Li, 和 J. Gao, “Set-of-mark 提示释放 GPT-4V 中卓越的视觉定位能力,”2023. [在线]. 可用:https://arxiv.org/abs/2310.11441 10, 17

[229] J. Y. Koh, R. Lo, L. Jang, V. Duvvur, M. C. Lim, P.-Y. Huang, G. Neu-big, S. Zhou, R. Salakhutdinov, and D. Fried, ”Visualwebarena: Evaluating multimodal agents on realistic visual web tasks,” arXiv preprint arXiv:2401.13649, 2024. 10, 17

J. Y. Koh, R. Lo, L. Jang, V. Duvvur, M. C. Lim, P.-Y. Huang, G. Neu-big, S. Zhou, R. Salakhutdinov, 和 D. Fried,“Visualwebarena: 在真实视觉网页任务中评估多模态代理,”arXiv 预印本 arXiv:2401.13649, 2024. 10, 17

[230] R. Bonatti, D. Zhao, F. Bonacci, D. Dupont, S. Abdali, Y. Li, Y. Lu, J. Wagle, K. Koishida, A. Bucker et al., "Windows agent arena: Evaluating multi-modal os agents at scale," arXiv preprint arXiv:2409.08264, 2024. 11

R. Bonatti, D. Zhao, F. Bonacci, D. Dupont, S. Abdali, Y. Li, Y. Lu, J. Wagle, K. Koishida, A. Bucker 等, "Windows 代理竞技场: 大规模评估多模态操作系统代理," arXiv 预印本 arXiv:2409.08264, 2024. 11

[231] Y. Ge, Y. Ren, W. Hua, S. Xu, J. Tan, and Y. Zhang, "Llm as os, agents as apps: Envisioning aios, agents and the aios-agent ecosystem," arXiv e-prints, pp. arXiv-2312, 2023. 11

Y. Ge, Y. Ren, W. Hua, S. Xu, J. Tan, 和 Y. Zhang, "大型语言模型作为操作系统，代理作为应用: 展望 AIOS、代理及 AIOS-代理生态系统," arXiv 电子预印本, 页码 arXiv-2312, 2023. 11

[232] K. Mei, Z. Li, S. Xu, R. Ye, Y. Ge, and Y. Zhang, "Aios: Llm agent operating system," arXiv e-prints, pp. arXiv-2403, 2024. 11

K. Mei, Z. Li, S. Xu, R. Ye, Y. Ge, 和 Y. Zhang, "AIOS: 大型语言模型代理操作系统," arXiv 电子预印本, 页码 arXiv-2403, 2024. 11

[233] Y. Deng, X. Zhang, W. Zhang, Y. Yuan, S.-K. Ng, and T.-S. Chua, "On the multi-turn instruction following for conversational web agents," arXiv preprint arXiv:2402.15057, 2024. 11

Y. Deng, X. Zhang, W. Zhang, Y. Yuan, S.-K. Ng, 和 T.-S. Chua, "关于对话式网页代理的多轮指令跟随," arXiv 预印本 arXiv:2402.15057, 2024. 11

[234] T. Li, G. Li, Z. Deng, B. Wang, and Y. Li, "A zero-shot language agent for computer control with structured reflection," arXiv preprint arXiv:2310.08740, 2023. 12

T. Li, G. Li, Z. Deng, B. Wang, 和 Y. Li, "一种带结构化反思的零样本语言代理用于计算机控制," arXiv 预印本 arXiv:2310.08740, 2023. 12

[235] K. Gandhi, J.-P. Fränken, T. Gerstenberg, and N. Goodman, "Understanding social reasoning in language models with language models," Advances in Neural Information Processing Systems, vol. 36, 2024. 12

K. Gandhi, J.-P. Fränken, T. Gerstenberg, 和 N. Goodman, "用语言模型理解语言模型中的社会推理," 神经信息处理系统进展, 第 36 卷, 2024. 12

[236] Z. Chen, Y. Li, and K. Wang, "Optimizing reasoning abilities in large language models: A step-by-step approach," Authorea Preprints, 2024. 12

Z. Chen, Y. Li, 和 K. Wang, "优化大型语言模型的推理能力: 一步步方法," Authorea 预印本, 2024. 12

[237] A. Plaat, A. Wong, S. Verberne, J. Broekens, N. van Stein, and T. Back, "Reasoning with large language models, a survey," arXiv preprint arXiv:2407.11511, 2024. 12

A. Plaat, A. Wong, S. Verberne, J. Broekens, N. van Stein, 和 T. Back, “大型语言模型推理综述,” arXiv 预印本 arXiv:2407.11511, 2024. 12

[238] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou et al., ”Chain-of-thought prompting elicits reasoning in large language models,” Advances in neural information processing systems, vol. 35, pp. 24824-24837, 2022. 12, 16

J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou 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, 第 35 卷, 页 24824-24837, 2022. 12, 16

[239] J. Y. Koh, S. McAleer, D. Fried, and R. Salakhutdinov, ”Tree search for language model agents,” arXiv preprint arXiv:2407.01476, 2024. 12

J. Y. Koh, S. McAleer, D. Fried, 和 R. Salakhutdinov, “语言模型代理的树搜索,” arXiv 预印本 arXiv:2407.01476, 2024. 12

[240] W. E. Bishop, A. Li, C. Rawles, and O. Riva, ”Latent state estimation helps ui agents to reason,” arXiv preprint arXiv:2405.11120, 2024. 12

W. E. Bishop, A. Li, C. Rawles, 和 O. Riva, “潜在状态估计助力用户界面代理推理,” arXiv 预印本 arXiv:2405.11120, 2024. 12

[241] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, ”Reflexion: Language agents with verbal reinforcement learning,” Advances in Neural Information Processing Systems, vol. 36, 2024. 12

N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, 和 S. Yao, “Reflexion: 基于语言代理的口头强化学习,” 《神经信息处理系统进展》(Advances in Neural Information Processing Systems), 第 36 卷, 2024 年. 12

[242] J. Pan, Y. Zhang, N. Tomlin, Y. Zhou, S. Levine, and A. Suhr, ”Autonomous evaluation and refinement of digital agents,” arXiv preprint arXiv:2404.06474, 2024. 12

J. Pan, Y. Zhang, N. Tomlin, Y. Zhou, S. Levine, 和 A. Suhr, “数字代理的自主评估与优化,” arXiv 预印本 arXiv:2404.06474, 2024 年. 12

[243] P. Duan, C.-Y. Cheng, G. Li, B. Hartmann, and Y. Li, ”Uicrit: Enhancing automated design evaluation with a ui critique dataset,” in Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology, 2024, pp. 1-17. 12

P. Duan, C.-Y. Cheng, G. Li, B. Hartmann, 和 Y. Li, “UICrit: 利用界面批评数据集提升自动化设计评估,” 载于第 37 届年度 ACM 用户界面软件与技术研讨会论文集, 2024 年, 第 1-17 页. 12

[244] N. Patil, D. Bhole, and P. Shete, ”Enhanced ui automator viewer with improved android accessibility evaluation features,” in 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT). IEEE, 2016, pp. 977-983. 12

N. Patil, D. Bhole, 和 P. Shete, “增强型 UI Automator Viewer 及其改进的 Android 无障碍评估功能,” 载于 2016 年国际自动控制与动态优化技术会议 (ICACDOT). IEEE, 2016 年, 第 977-983 页. 12

[245] G. Lodi, ”Xctest introduction,” in Test-Driven Development in Swift: Compile Better Code with XCTest and TDD. Springer, 2021, pp. 13-25. 12

G. Lodi, “XCTest 简介,” 载于《Swift 中的测试驱动开发: 使用 XCTest 和 TDD 编写更优代码》. Springer, 2021 年, 第 13-25 页. 12

[246] S. Singh, R. Gadgil, and A. Chudgor, ”Automated testing of mobile applications using scripting technique: A study on appium,” International Journal of Current Engineering and Technology (IJCET), vol. 4, no. 5, pp. 3627-3630, 2014. 12

S. Singh, R. Gadgil, 和 A. Chudgor, “基于脚本技术的移动应用自动化测试: 以 Appium 为例,”《国际当前工程与技术期刊》(International Journal of Current Engineering and Technology, IJCET), 第 4 卷第 5 期, 第 3627-3630 页, 2014 年. 12

[247] U. Gundecha, Selenium Testing Tools Cookbook. Packt Publishing Ltd, 2015. 12

U. Gundecha, 《Selenium 测试工具手册》. Packt Publishing Ltd, 2015 年. 12

[248] C. Sinclair, ”The role of selenium in mobile application testing.” 12

C. Sinclair, “Selenium 在移动应用测试中的作用.”12

[249] A. Torreno, E. Onaindia, A. Komenda, and M. Štolba, ”Cooperative multi-agent planning: A survey,” ACM Computing Surveys (CSUR), vol. 50, no. 6, pp. 1-32, 2017. 12

A. Torreno, E. Onaindia, A. Komenda, 和 M. Štolba, “协作多智能体规划: 综述,”《ACM 计算机调查》(ACM Computing Surveys, CSUR), 第 50 卷第 6 期, 第 1-32 页, 2017 年. 12

[250] A. Dorri, S. S. Kanhere, and R. Jurdak, ”Multi-agent systems: A survey,” Ieee Access, vol. 6, pp. 28573-28593, 2018. 12

A. Dorri, S. S. Kanhere, 和 R. Jurdak, “多智能体系统: 综述,”《IEEE Access》, 第 6 卷, 第 28573-28593 页, 2018 年. 12

[251] R. Gong, Q. Huang, X. Ma, H. Vo, Z. Durante, Y. Noda, Z. Zheng, S.-C. Zhu, D. Terzopoulos, L. Fei-Fei et al., ”Mindagent: Emergent gaming interaction,” arXiv preprint arXiv:2309.09971, 2023. 12

R. Gong, Q. Huang, X. Ma, H. Vo, Z. Durante, Y. Noda, Z. Zheng, S.-C. Zhu, D. Terzopoulos, L. Fei-Fei 等, “Mindagent: 新兴的游戏交互,” arXiv 预印本 arXiv:2309.09971, 2023. 12

[252] F. Chen, W. Ren et al., ”On the control of multi-agent systems: A survey,” Foundations and Trends® in Systems and Control, vol. 6, no. 4, pp. 339-499, 2019. 12

F. Chen, W. Ren 等, “多智能体系统控制综述,”《系统与控制基础与趋势 ®》, 第 6 卷, 第 4 期, 页 339-499, 2019. 12

[253] Y. Talebirad and A. Nadiri, ”Multi-agent collaboration: Harnessing the power of intelligent llm agents,” arXiv preprint arXiv:2306.03314, 2023. 12

Y. Talebirad 和 A. Nadiri, “多智能体协作: 利用智能大型语言模型 (LLM) 代理的力量,”arXiv 预印本 arXiv:2306.03314, 2023. 12

[254] Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, and C. Wang, ”Autogen: Enabling next-gen llm applications via multi-agent conversation framework,” arXiv preprint arXiv:2308.08155, 2023. 12

Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, 和 C. Wang, “Autogen: 通过多智能体对话框架支持下一代大型语言模型应用,”arXiv 预印本 arXiv:2308.08155, 2023. 12

[255] W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C.-M. Chan, H. Yu, Y. Lu, Y.-H. Hung, C. Qian et al., ”Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors,” in The Twelfth International Conference on Learning Representations, 2023. 12

W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C.-M. Chan, H. Yu, Y. Lu, Y.-H. Hung, C. Qian 等,“Agentverse: 促进多智能体协作与探索新兴行为,”载于第十二届国际学习表征会议, 2023. 12

[256] H. Li, Y. Q. Chong, S. Stepputtis, J. Campbell, D. Hughes, M. Lewis, and K. Sycara, ”Theory of mind for multi-agent collaboration via large language models,” arXiv preprint arXiv:2310.10701, 2023. 12

H. Li, Y. Q. Chong, S. Stepputtis, J. Campbell, D. Hughes, M. Lewis, 和 K. Sycara, “通过大型语言模型实现多智能体协作的心智理论,”arXiv 预印本 arXiv:2310.10701, 2023. 12

[257] Z. Liu, Y. Zhang, P. Li, Y. Liu, and D. Yang, ”A dynamic llm-powered agent network for task-oriented agent collaboration,” in First Conference on Language Modeling, 2024. 12

Z. Liu, Y. Zhang, P. Li, Y. Liu, 和 D. Yang, “面向任务的动态大型语言模型驱动智能体网络协作,”载于首届语言建模会议, 2024. 12

[258] X. Li, S. Wang, S. Zeng, Y. Wu, and Y. Yang, ”A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges,” Vicinagearth, vol. 1, no. 1, p. 9, 2024. 12

X. Li, S. Wang, S. Zeng, Y. Wu, 和 Y. Yang, “基于大型语言模型的多智能体系统综述: 工作流程、基础设施与挑战,”《Vicinagearth》, 第 1 卷, 第 1 期, 页 9, 2024. 12

[259] K.-T. Tran, D. Dao, M.-D. Nguyen, Q.-V. Pham, B. O’Sullivan, and H. D. Nguyen, ”Multi-agent collaboration mechanisms: A survey of llms,” arXiv preprint arXiv:2501.06322, 2025. 12

K.-T. Tran, D. Dao, M.-D. Nguyen, Q.-V. Pham, B. O’Sullivan, 和 H. D. Nguyen, “多智能体协作机制: 大型语言模型综述,”arXiv 预印本 arXiv:2501.06322, 2025. 12

[260] D. Yin, F. Brahman, A. Ravichander, K. Chandu, K.-W. Chang, Y. Choi, and B. Y. Lin, ”Agent lumos: Unified and modular training for open-source language agents,” in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 12380-12403. 14

D. Yin, F. Brahman, A. Ravichander, K. Chandu, K.-W. Chang, Y. Choi, 和 B. Y. Lin,“Agent Lumos: 开源语言智能体的统一模块化训练,”载于第 62 届计算语言学协会年会论文集 (第一卷: 长篇论文), 2024, 页 12380-12403. 14

[261] Y. Zhang, Z. Ma, Y. Ma, Z. Han, Y. Wu, and V. Tresp, ”Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration,” arXiv preprint arXiv:2408.15978, 2024. 14

Y. Zhang, Z. Ma, Y. Ma, Z. Han, Y. Wu, 和 V. Tresp, “Webpilot: 用于网页任务执行的多功能自主多智能体系统，具备策略性探索能力,”arXiv 预印本 arXiv:2408.15978, 2024. 14

[262] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, ”Tree of thoughts: Deliberate problem solving with large language models,” Advances in Neural Information Processing Systems, vol. 36, 2024. 16

S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, 和 K. Narasimhan, “思维树: 利用大型语言模型的深思熟虑问题解决方法,”《神经信息处理系统进展》, 第 36 卷, 2024. 16

[263] W. Chen, X. Ma, X. Wang, and W. W. Cohen, ”Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks,” arXiv preprint arXiv:2211.12588, 2022. 16

W. Chen, X. Ma, X. Wang, 和 W. W. Cohen,“思维程序提示: 将计算与推理分离以解决数值推理任务,” arXiv 预印本 arXiv:2211.12588, 2022. 16

[264] Y. Yang, X. Yang, S. Li, C. Lin, Z. Zhao, C. Shen, and T. Zhang, ”Security matrix for multimodal agents on mobile devices: A systematic and proof of concept study,” arXiv preprint arXiv:2407.09295, 2024. 16

Y. Yang, X. Yang, S. Li, C. Lin, Z. Zhao, C. Shen, 和 T. Zhang,“移动设备多模态智能体的安全矩阵: 系统性研究与概念验证,”arXiv 预印本 arXiv:2407.09295, 2024. 16

[265] Z. Zhang, Y. Yao, A. Zhang, X. Tang, X. Ma, Z. He, Y. Wang, M. Gerstein, R. Wang, G. Liu et al., ”Igniting language intelligence: The hitchhiker’s guide from chain-of-thought reasoning to language agents,” arXiv preprint arXiv:2311.11797, 2023. 16

张志, 姚洋, 张安, 唐晓, 马翔, 何志, 王勇, M. Gerstein, 王锐, 刘刚等, ” 点燃语言智能: 从链式思维推理到语言代理的指南,” arXiv 预印本 arXiv:2311.11797, 2023. 16

[266] J. Li, D. Li, S. Savarese, and S. Hoi, ”Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” in International conference on machine learning. PMLR, 2023, pp. 19730-19742. 17

李杰, 李东, S. Savarese, 和 S. Hoi, ”Blip-2: 利用冻结的图像编码器和大型语言模型启动语言-图像预训练,” 载于国际机器学习会议. PMLR, 2023, 页 19730-19742. 17

[267] Q. Ye, H. Xu, G. Xu, J. Ye, M. Yan, Y. Zhou, J. Wang, A. Hu, P. Shi, Y. Shi et al., ”mplug-owl: Modularization empowers large language models with multimodality,” arXiv preprint arXiv:2304.14178, 2023. 17

叶强, 徐浩, 徐刚, 叶军, 闫明, 周阳, 王军, 胡安, 石鹏, 石勇等, ”mplug-owl: 模块化赋能大型语言模型多模态能力,” arXiv 预印本 arXiv:2304.14178, 2023. 17

[268] W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song et al., ”Cogvlm: Visual expert for pretrained language models,” arXiv preprint arXiv:2311.03079, 2023. 17

王伟, 吕强, 余伟, 洪伟, 齐军, 王勇, 纪军, 杨志, 赵磊, 宋翔等, ”Cogvlm: 预训练语言模型的视觉专家,” arXiv 预印本 arXiv:2311.03079, 2023. 17

[269] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, ”Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond,” arXiv preprint arXiv:2308.12966, 2023. 17

白军, 白帅, 杨帅, 王帅, 谭帅, 王鹏, 林军, 周超, 和周军, ”Qwen-vl: 一款多功能视觉语言模型，支持理解、定位、文本阅读及更多,” arXiv 预印本 arXiv:2308.12966, 2023. 17

[270] H. Liu, C. Li, Q. Wu, and Y. J. Lee, ”Visual instruction tuning,” Advances in neural information processing systems, vol. 36, 2024. 17

刘浩, 李晨, 吴强, 和 Y. J. Lee, ” 视觉指令调优,” 神经信息处理系统进展, 第 36 卷, 2024. 17

[271] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, ”Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution,” arXiv preprint arXiv:2409.12191, 2024. 17

王鹏, 白帅, 谭帅, 王帅, 范志, 白军, 陈凯, 刘翔, 王军, 葛伟, 范勇, 党凯, 杜明, 任翔, 门锐, 刘东, 周超, 周军, 和林军, ”Qwen2-vl: 提升视觉语言模型对任意分辨率世界的感知,” arXiv 预印本 arXiv:2409.12191, 2024. 17

[272] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu et al., ”Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 24 185-24 198. 17

陈志, 吴军, 王伟, 苏伟, 陈刚, 邢帅, 钟明, 张强, 朱晓, 陆磊等, ”Internvl: 扩展视觉基础模型并对齐以支持通用视觉语言任务,” 载于 IEEE/CVF 计算机视觉与模式识别会议论文集, 2024, 页 24185-24198. 17

[273] Z. Chen, W. Wang, H. Tian, S. Ye, Z. Gao, E. Cui, W. Tong, K. Hu, J. Luo, Z. Ma et al., ”How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites,” arXiv preprint arXiv:2404.16821, 2024. 17

陈志, 王伟, 田浩, 叶帅, 高志, 崔恩, 童伟, 胡凯, 罗军, 马志等, ” 我们距离 GPT-4V 有多远? 利用开源套件缩小与商业多模态模型的差距,” arXiv 预印本 arXiv:2404.16821, 2024. 17

[274] L. Zheng, R. Wang, X. Wang, and B. An, ”Synapse: Trajectory-as-exemplar prompting with memory for computer control,” in The Twelfth International Conference on Learning Representations, 2023. 17

郑磊, 王锐, 王翔, 和安斌, ”Synapse: 基于轨迹示例的提示与记忆用于计算机控制,” 载于第十二届国际学习表征会议, 2023. 17

[275] R. Varghese and M. Sambath, ”Yolov8: A novel object detection algorithm with enhanced performance and robustness,” in 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). IEEE, 2024, pp. 1-6. 18

R. Varghese 和 M. Sambath, ”Yolov8: 一种性能和鲁棒性增强的新型目标检测算法,” 载于 2024 年数据工程与智能计算系统进展国际会议 (ADICS). IEEE, 2024, 页 1-6. 18

[276] Y. Du, C. Li, R. Guo, X. Yin, W. Liu, J. Zhou, Y. Bai, Z. Yu, Y. Yang, Q. Dang et al., ”Pp-ocr: A practical ultra lightweight ocr system,” arXiv preprint arXiv:2009.09941, 2020. 18

杜阳, 李晨, 郭锐, 尹翔, 刘伟, 周军, 白勇, 余志, 杨勇, 党强等, ”Pp-ocr: 一款实用的超轻量级光学字符识别系统,” arXiv 预印本 arXiv:2009.09941, 2020. 18

[277] H.-M. Xu, Q. Chen, L. Wang, and L. Liu, ”Attention-driven gui grounding: Leveraging pretrained multimodal large language models without fine-tuning,” arXiv preprint arXiv:2412.10840, 2024. 21

H.-M. Xu, Q. Chen, L. Wang, 和 L. Liu, “基于注意力驱动的 GUI 定位: 利用预训练多模态大型语言模型无需微调,”arXiv 预印本 arXiv:2412.10840, 2024. 21

[278] L. P. Kaelbling, M. L. Littman, and A. W. Moore, ”Reinforcement learning: A survey,” Journal of artificial intelligence research, vol. 4, pp. 237-285, 1996. 24

L. P. Kaelbling, M. L. Littman, 和 A. W. Moore, “强化学习综述,”人工智能研究杂志, 第 4 卷, 页 237-285, 1996. 24

[279] J. Zheng, L. Wang, F. Yang, C. Zhang, L. Mei, W. Yin, Q. Lin, D. Zhang, S. Rajmohan, and Q. Zhang, ”Vem: Environment-free exploration for training gui agent with value environment model,” arXiv preprint arXiv:2502.18906, 2025. 25

J. Zheng, L. Wang, F. Yang, C. Zhang, L. Mei, W. Yin, Q. Lin, D. Zhang, S. Rajmohan, 和 Q. Zhang,“VEM: 基于价值环境模型的无环境探索用于训练 GUI 代理,”arXiv 预印本 arXiv:2502.18906, 2025. 25

[280] Y.-H. Peng, F. Huq, Y. Jiang, J. Wu, X. Y. Li, J. P. Bigham, and A. Pavel, ”Dreamstruct: Understanding slides and user interfaces via synthetic data generation,” in European Conference on Computer Vision. Springer, 2024, pp. 466-485. 28

Y.-H. Peng, F. Huq, Y. Jiang, J. Wu, X. Y. Li, J. P. Bigham, 和 A. Pavel, “DreamStruct: 通过合成数据生成理解幻灯片和用户界面,”欧洲计算机视觉会议论文集. Springer, 2024, 页 466-485. 28

[281] Q. Sun, K. Cheng, Z. Ding, C. Jin, Y. Wang, F. Xu, Z. Wu, C. Jia, L. Chen, Z. Liu et al., ”Os-genesis: Automating gui agent trajectory construction via reverse task synthesis,” arXiv preprint arXiv:2412.19723, 2024.

Q. Sun, K. Cheng, Z. Ding, C. Jin, Y. Wang, F. Xu, Z. Wu, C. Jia, L. Chen, Z. Liu 等, “OS-Genesis: 通过逆向任务合成自动构建 GUI 代理轨迹,”arXiv 预印本 arXiv:2412.19723, 2024. 28

[282] H. Su, R. Sun, J. Yoon, P. Yin, T. Yu, and S. Ö. Arık, ”Learn-by-interact: A data-centric framework for self-adaptive agents in realistic environments,” arXiv preprint arXiv:2501.10893, 2025. 28

H. Su, R. Sun, J. Yoon, P. Yin, T. Yu, 和 S. Ö. Arık, “Learn-by-Interact: 面向现实环境中自适应代理的数据中心框架,”arXiv 预印本 arXiv:2501.10893, 2025. 28

[283] W. Wang, Z. Yu, W. Liu, R. Ye, T. Jin, S. Chen, and Y. Wang, ”Fedmobileagent: Training mobile agents using decentralized self-sourced data from diverse users,” arXiv preprint arXiv:2502.02982, 2025. 28, 32

W. Wang, Z. Yu, W. Liu, R. Ye, T. Jin, S. Chen, 和 Y. Wang, “FedMobileAgent: 利用来自多样用户的去中心化自源数据训练移动代理,”arXiv 预印本 arXiv:2502.02982, 2025. 28, 32

[284] O. Berkovitch, S. Caduri, N. Kahlon, A. Efros, A. Caciularu, and I. Dagan, ”Identifying user goals from ui trajectories,” arXiv preprint arXiv:2406.14314, 2024. 28

O. Berkovitch, S. Caduri, N. Kahlon, A. Efros, A. Caciularu, 和 I. Dagan, “从用户界面轨迹识别用户目标,”arXiv 预印本 arXiv:2406.14314, 2024. 28

[285] K. Q. Lin, L. Li, D. Gao, Q. Wu, M. Yan, Z. Yang, L. Wang, and M. Z. Shou, ”Videogui: A benchmark for gui automation from instructional videos,” arXiv preprint arXiv:2406.10227, 2024. 31

K. Q. Lin, L. Li, D. Gao, Q. Wu, M. Yan, Z. Yang, L. Wang, 和 M. Z. Shou, “VideoGUI: 基于教学视频的 GUI 自动化基准,”arXiv 预印本 arXiv:2406.10227, 2024. 31

[286] W. Chen and Z. Li, ”Octopus v2: On-device language model for super agent,” arXiv preprint arXiv:2404.01744, 2024. 31

W. Chen 和 Z. Li, “Octopus v2: 面向超级代理的设备端语言模型,”arXiv 预印本 arXiv:2404.01744, 2024. 31

[287] H. Wen, S. Tian, B. Pavlov, W. Du, Y. Li, G. Chang, S. Zhao, J. Liu, Y. Liu, Y.-Q. Zhang et al., ”Autodroid-v2: Boosting slm-based gui agents via code generation,” arXiv preprint arXiv:2412.18116, 2024. 31

H. Wen, S. Tian, B. Pavlov, W. Du, Y. Li, G. Chang, S. Zhao, J. Liu, Y. Liu, Y.-Q. Zhang 等,“AutoDroid-v2: 通过代码生成提升基于 SLM 的 GUI 代理,”arXiv 预印本 arXiv:2412.18116, 2024. 31

[288] F. Wang, Z. Zhang, X. Zhang, Z. Wu, T. Mo, Q. Lu, W. Wang, R. Li, J. Xu, X. Tang et al., ”A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness,” arXiv preprint arXiv:2411.03350, 2024. 31

F. Wang, Z. Zhang, X. Zhang, Z. Wu, T. Mo, Q. Lu, W. Wang, R. Li, J. Xu, X. Tang 等, “大型语言模型时代小型语言模型的综合综述: 技术、增强、应用、与大型语言模型的协作及可信性,”arXiv 预印本 arXiv:2411.03350, 2024. 31

[289] F. Huq, J. P. Bigham, and N. Martelaro, ”What’s important here?: Opportunities and challenges of llm in retrieving information from web interface,” R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models. 31

F. Huq, J. P. Bigham, 和 N. Martelaro, “这里重要的是什么? : 大型语言模型在从网页界面检索信息中的机遇与挑战,”R0-FoMo: 大型基础模型中少样本和零样本学习的鲁棒性. 31

[290] W. Li, F.-L. Hsu, W. Bishop, F. Campbell-Ajala, M. Lin, and O. Riva, ”Uinav: A practical approach to train on-device automation agents,” in Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), 2024, pp. 36-51. 31

W. Li, F.-L. Hsu, W. Bishop, F. Campbell-Ajala, M. Lin, 和 O. Riva, “Uinav: 一种实用的设备端自动化代理训练方法,”发表于 2024 年北美计算语言学协会人类语言技术分会会议论文集 (第 6 卷: 产业轨道), 2024 年, 第 36-51 页。31

[291] C. H. Wu, J. Y. Koh, R. Salakhutdinov, D. Fried, and A. Raghu-nathan, ”Adversarial attacks on multi-modal agents,” arXiv preprint arXiv:2406.12814, 2024. 32

C. H. Wu, J. Y. Koh, R. Salakhutdinov, D. Fried, 和 A. Raghunathan, “多模态代理的对抗攻击,”arXiv 预印本 arXiv:2406.12814, 2024 年。32

[292] W. Yang, X. Bi, Y. Lin, S. Chen, J. Zhou, and X. Sun, ”Watch out for your agents! investigating backdoor threats to llm-based agents,” arXiv preprint arXiv:2402.11208, 2024. 32

W. Yang, X. Bi, Y. Lin, S. Chen, J. Zhou, 和 X. Sun, “警惕你的代理! 基于大语言模型 (LLM) 代理的后门威胁调查,”arXiv 预印本 arXiv:2402.11208, 2024 年。32

[293] Y. Wang, D. Xue, S. Zhang, and S. Qian, ”Badagent: Inserting and activating backdoor attacks in llm agents,” arXiv preprint arXiv:2406.03007, 2024. 32

Y. Wang, D. Xue, S. Zhang, 和 S. Qian, “Badagent: 在大语言模型 (LLM) 代理中插入并激活后门攻击,”arXiv 预印本 arXiv:2406.03007, 2024 年。32

[294] Y. Chen, X. Hu, K. Yin, J. Li, and S. Zhang, ”Aeia-mn: Evaluating the robustness of multimodal llm-powered mobile agents against active environmental injection attacks,” arXiv preprint arXiv:2502.13053, 2025. 32

Y. Chen, X. Hu, K. Yin, J. Li, 和 S. Zhang, “AEIA-MN: 评估多模态大语言模型 (LLM) 驱动移动代理对主动环境注入攻击的鲁棒性,”arXiv 预印本 arXiv:2502.13053, 2025 年。32