A Survey on GUI Agents with Foundation Models Enhanced by Reinforcement Learning

基于强化学习增强基础模型的图形用户界面 (GUI) 智能体综述

Jiahao Li, Kaer Huang

李佳豪，黄凯尔

Lenovo Research

联想研究院

jiahaoli0301@gmail.com

Abstract

Graphical User Interface (GUI) agents, driven by Multi-modal Large Language Models (MLLMs), have emerged as a promising paradigm for enabling intelligent interaction with digital systems. This paper provides a structured summary of recent advances in GUI agents, focusing on architectures enhanced by Reinforcement Learning (RL). We first formalize GUI agent tasks as Markov Decision Processes and discuss typical execution environments and evaluation metrics. We then review the modular architecture of (M)LLM-based GUI agents, covering Perception, Planning, and Acting modules, and trace their evolution through representative works. Furthermore, we categorize GUI agent training methodologies into Prompt-based, Supervised Fine-Tuning (SFT)-based, and RL-based approaches, highlighting the progression from simple prompt engineering to dynamic policy learning via RL. Our summary illustrates how recent innovations in multimodal perception, decision reasoning, and adaptive action generation have significantly improved the generalization and robustness of GUI agents in complex real-world environments. We conclude by identifying key challenges and future directions for building more capable and reliable GUI agents.

摘要

由多模态大语言模型 (MLLM) 驱动的图形用户界面 (GUI) 智能体，已成为实现与数字系统进行智能交互的一种有前景的范式。本文对 GUI 智能体的最新进展进行了系统性总结，重点关注由强化学习 (RL) 增强的架构。我们首先将 GUI 智能体任务形式化为马尔可夫决策过程，并讨论典型的执行环境和评估指标。然后，我们回顾了基于 (M)LLM 的 GUI 智能体的模块化架构，涵盖感知、规划和执行模块，并通过代表性工作追溯其发展历程。此外，我们将 GUI 智能体的训练方法分为基于提示、基于监督微调 (SFT) 和基于强化学习的方法，强调了从简单的提示工程到通过强化学习进行动态策略学习的发展过程。我们的总结表明，多模态感知、决策推理和自适应动作生成方面的最新创新如何显著提高了 GUI 智能体在复杂现实环境中的泛化能力和鲁棒性。最后，我们确定了构建更强大、更可靠的 GUI 智能体的关键挑战和未来方向。

Contents

目录

1 Introduction

1 引言

With the continuous advancements of LLMs in the fields of Natural Language Processing (NLP) and Computer Vision (CV), LLMs have demonstrated impressive performance in various downstream tasks after being trained on large-scale corpora using the "next token prediction" paradigm [23][30][21] . LLMs have progressively evolved from a simple conversational chatbot [24][16][5][6] to autonomous agents [1][26][19] capable of planning [29] , tool use [22] , and memory management [17][18] to perform and complete complex tasks. Building on this foundation, a series of works have emerged that utilize LLMs to interact with digital systems [28][8] [8] [36] [13] [20] [14] [3] [4] [34] [10] [25] (e.g., smartphones or computers), where LLMs interact with digital devices through Graphical User Interfaces (GUIs) in the same way as humans.

随着大语言模型 (LLMs) 在自然语言处理 (NLP) 和计算机视觉 (CV) 领域的不断发展，LLMs 在使用“下一个词预测”范式在大规模语料库上进行训练后，在各种下游任务中表现出了令人瞩目的性能 [23][30][21] 。LLMs 已逐渐从简单的对话式聊天机器人 [24][16][5][6] 发展成为能够进行规划 [29] 、工具使用 [22] 和内存管理 [17][18] 以执行和完成复杂任务的自主智能体 [1][26][19] 。在此基础上，出现了一系列利用 LLMs 与数字系统 [28][8] [8] [36] [13] [20] [14] [3] [4] [34] [10] [25] (如智能手机或计算机) 进行交互的研究，其中 LLMs 通过图形用户界面 (GUIs) 以与人类相同的方式与数字设备进行交互。

This paradigm holds significant research and development potential, as GUIs are integral to nearly all digital devices, smartphones, tablets, desktops, and even televisions, used in daily human activities such as work, learning,

and leisure. In the era of rapid AI development, using LLM-based agents to control digital devices to fulfill human user needs can significantly enhance the user experience, offering more convenient and efficient services, and demonstrating great value potential. However, previous studies based on traditional rule-based and RL methods have struggled with tasks resembling human interactions [9] , limiting their applicability. Recent advances in LLMs and MLLMs have significantly bolstered their capabilities in semantic comprehension and cognitive reasoning. Agents built on (M)LLMs can effectively interpret and integrate textual and visual input, enabling them to devise detailed strategies to tackle complex tasks. These breakthroughs also provide opportunities to address previously mentioned challenges in handling complex tasks, making it possible for agents to autonomously complete user tasks in GUIs.

这种范式具有巨大的研发潜力，因为 GUI 几乎是所有数字设备 (智能手机、平板电脑、台式机，甚至电视) 的组成部分，在人类日常的工作、学习和休闲等活动中都会使用。在人工智能快速发展的时代，使用基于 LLM 的智能体来控制数字设备以满足人类用户的需求，可以显著提升用户体验，提供更便捷、高效的服务，展现出巨大的价值潜力。然而，以往基于传统规则和强化学习方法的研究在处理类似人类交互的任务时遇到了困难 [9] ，限制了它们的适用性。最近 LLMs 和多模态大语言模型 (MLLMs) 的进展显著增强了它们在语义理解和认知推理方面的能力。基于 (M)LLMs 构建的智能体可以有效地解释和整合文本和视觉输入，使它们能够制定详细的策略来应对复杂任务。这些突破也为解决前面提到的处理复杂任务的挑战提供了机会，使智能体能够在 GUI 中自主完成用户任务。

This paper aims to summarize recent learning experiences and provide an overview of the current novel and influential approaches for GUI agents. Specifically, it covers the foundational concepts of GUI agents, the classification of execution environments, architectural frameworks, and key methodological categories.

本文旨在总结近期的学习经验，并概述当前针对 GUI 智能体的新颖且有影响力的方法。具体而言，它涵盖了 GUI 智能体的基本概念、执行环境的分类、架构框架和关键方法类别。

2 Problem Formulation

2 问题描述

A GUI agent refers to an agent that autonomously interacts with digital devices (such as smartphones or desktop computers) through GUIs. It makes decisions and generates plans based on user-defined task instructions and the current screen context, executing actions through interaction with clickable or typeable elements to achieve user goals.

GUI 智能体是指通过 GUI 自主与数字设备 (如智能手机或台式计算机) 进行交互的智能体。它根据用户定义的任务指令和当前屏幕上下文做出决策并生成计划，通过与可点击或可输入元素进行交互来执行操作，以实现用户目标。

The interaction process between GUI agents and their environments is commonly formalized as a Markov Decision Process (MDP), denoted as $M = \{S, A, T, r, \gamma\}$. In this formulation, S represents the state space, defined as the set of all possible screen captures of the digital device. A denotes the set of permissible actions, such as clicking, typing, and other forms of interaction on the screen. The transition function $T : S \times A \times S \to [0, 1]$ defines the probability distribution over the next states given a current state and an action. The reward function

$r : S \times A \rightarrow R$ assigns a scalar feedback value to each state-action pair. γ is the discount factor. The goal of the GUI agent is to learn a decision policy π that maximizes returns. We typically represent $\pi(a \mid s) \in [0, 1]$ as the probability of selecting action a in state s under the policy π .

GUI 代理与其环境之间的交互过程通常被形式化为马尔可夫决策过程 (Markov Decision Process, MDP), 记作 $M = \{S, A, T, r, \gamma\}$ 。在该表述中, S 表示状态空间, 定义为数字设备所有可能屏幕截图的集合。A 表示允许的动作集合, 如点击、输入及其他屏幕交互形式。转移函数 $T : S \times A \times S \rightarrow [0, 1]$ 定义了在给定当前状态和动作下，下一状态的概率分布。奖励函数 $r : S \times A \rightarrow R$ 为每个状态-动作对分配一个标量反馈值。γ 是折扣因子。GUI 代理的目标是学习一个决策策略 π ，以最大化回报。我们通常将 $\pi(a \mid s) \in [0, 1]$ 表示为在策略 π 下，在状态 s 中选择动作 a 的概率。

3 Benchmarks

3 基准测试

The execution environments of GUI agents usually include static environment datasets and dynamic interactive environments. In static settings, the environment remains constant, allowing GUI agents to work within predefined datasets without needing to account for environmental changes in action planning. In dynamic environments, the state may evolve after each action, necessitating that agents observe the updated state to determine subsequent actions. Dynamic environments are generally classified into simulated and real-world settings. Simulated environments are usually cleaner, free from disturbances like pop-up ads, and provide a standardized interactive setting for testing. In real environments, the agent must observe and interact with the digital device like humans. Although this setup more accurately reflects real-world scenarios, it also exposes agents to numerous unpredictable factors.

GUI 代理的执行环境通常包括静态环境数据集和动态交互环境。在静态环境中，环境保持不变，允许 GUI 代理在预定义的数据集内工作，无需在动作规划中考虑环境变化。在动态环境中，状态可能在每次动作后发生变化，代理需要观察更新后的状态以决定后续动作。动态环境通常分为模拟环境和真实环境。模拟环境通常更为纯净，无弹窗广告等干扰，提供标准化的交互测试环境。真实环境中，代理必须像人类一样观察并与数字设备交互。尽管这种设置更准确地反映了现实场景，但也使代理面临众多不可预测因素。

The success rate is the most common metric used to evaluate a GUI agent, reflecting its ability to complete tasks effectively. In addition, execution efficiency is another common evaluation metric, often measured by the number of steps, time, or cost required to complete a task. Notably, for dynamic interactive environments, due to uncertainty from environmental changes, the fixed evaluation metrics mentioned earlier may not be applicable. As a result, human experts are required for manual assessment, which compromises reproducibility and requires significant labor costs. Consequently, recent research has explored leveraging MLLMs for automating the evaluation process, aiming to reduce human involvement. However, due to the hallucination issues in LLMs, ensuring evaluation accuracy while reducing human effort remains a critical research challenge.

成功率是评估 GUI 代理最常用的指标，反映其完成任务的能力。此外，执行效率也是常见的评估指标，通常通过完成任务所需的步骤数、时间或成本来衡量。值得注意的是，对于动态交互环境，由于环境变化带来的不确定性，上述固定评估指标可能不适用。因此，需要人工专家进行手动评估，这影响了可重复性且需要大量人力成本。因此，近期研究探索利用多模态大语言模型 (MLLMs) 自动化评估过程，旨在减少人工参与。然而，由于大型语言模型 (LLMs) 存在幻觉问题，如何在减少人工投入的同时确保评估准确性仍是关键研究挑战。

4 (M)LLM-based GUI Agent Architectures

4 基于 (多模态) 大语言模型的 GUI 代理架构

GUI agents are expected to autonomously operate digital devices operations to meet human task requirements. Typically, GUI agents receive a task instruction and the current screen state as input, and generate a sequence of planned actions to achieve the objective. As depicted in Figure 1.The architecture generally consists of three key modules: Perception, Planning, and Acting.

GUI 代理被期望能够自主操作数字设备以满足人类的任务需求。通常，GUI 代理接收任务指令和当前屏幕状态作为输入，生成一系列规划动作以实现目标。如图 1 所示，该架构通常由感知、规划和执行三个关键模块组成。

4.1 Perception

4.1 感知

The Perception module is responsible for perceiving and understanding the GUI by extracting semantic information from interactive elements like buttons, text boxes, and icons, laying the groundwork for subsequent planning and action execution. For unimodal language models, the GUI perception module mainly relies on accessible APIs provided by operating systems or applications, extracting the type and position of interface elements via the view hierarchy and feeding this symbolic information into the language model. These methods offer structured interface data suitable for model processing, though their effectiveness heavily depends on the developers' proper implementation. However, when dealing with highly dynamic elements such as custom drawing canvases or game environments, the view hierarchy often fails to capture complete visual content, rendering the perception module ineffective. In contrast, for MLLMs, GUI agents can directly perceive the environment using visual information from screen-shots. For example, OmniParser [12] uses existing MLLMs (such as GPT-4V) to convert screen captures into structured representations of User Interface(UI) elements, extracting interface information directly from visual inputs and avoiding reliance on Accessibility APIs. As MLLMs continue to evolve, GUI perception has increasingly incorporated multimodal input sources for richer understanding. MP-GUI [28] introduces three specialized perceivers: a Textual Perceiver (TxP) for extracting text information, a Graphical Perceiver (GaP) for detecting graphical elements such as icons and buttons, and a Spatial Perceiver (SpP) for modeling spatial relationships between elements. By fusing the outputs of these perceivers, the model gains a more comprehensive understanding of the interface. Furthermore, CogAgent [8] introduces a lightweight high-resolution cross-attention module that supplements the original low-resolution large image encoder with a high-resolution small image encoder, enhancing perception of complex GUI interfaces through cross-attention with the underlying Vision-Language Mod-

els(VLMs). Real-world GUI environments are dynamic, requiring perception modules to handle interface changes (e.g., loading animations, pop-ups) and noise (e.g., ads, notifications). UI-Hawk [36] highlights the use of historical screen context in GUI navigation, leveraging history-aware visual encoders and efficient resampling modules to enhance comprehension of changing interfaces. Additionally, the perception of screen visuals may raise privacy concerns due to the presence of sensitive information.

感知模块负责通过提取按钮、文本框和图标等交互元素的语义信息来感知和理解图形用户界面 (GUI)，为后续的规划和动作执行奠定基础。对于单模态语言模型，GUI 感知模块主要依赖操作系统或应用程序提供的可访问 API，通过视图层级提取界面元素的类型和位置，并将这些符号信息输入语言模型。这些方法提供了适合模型处理的结构化界面数据，但其效果在很大程度上依赖于开发者的正确实现。然而，在处理高度动态的元素如自定义绘图画布或游戏环境时，视图层级往往无法捕捉完整的视觉内容，导致感知模块失效。相比之下，对于多模态大语言模型 (MLLMs)，GUI 代理可以直接利用屏幕截图中的视觉信息感知环境。例如，OmniParser [12] 使用现有的 MLLMs(如 GPT-4V) 将屏幕截图转换为用户界面 (UI) 元素的结构化表示，直接从视觉输入中提取界面信息，避免依赖辅助功能 API。随着 MLLMs 的不断发展，GUI 感知越来越多地融合了多模态输入源以实现更丰富的理解。MP-GUI [28] 引入了三种专门的感知器: 文本感知器 (TxP) 用于提取文本信息，图形感知器 (GaP) 用于检测图标和按钮等图形元素，空间感知器 (SpP) 用于建模元素之间的空间关系。通过融合这些感知器的输出，模型获得了对界面的更全面理解。此外， CogAgent [8] 引入了一个轻量级高分辨率交叉注意力模块，利用高分辨率小图像编码器补充原有的低分辨率大图像编码器，通过与底层视觉语言模型 (VLMs) 的交叉注意力增强对复杂 GUI 界面的感知。现实中的 GUI 环境是动态的，感知模块需要处理界面变化 (如加载动画、弹窗) 和噪声 (如广告、通知)。UI-Hawk [36] 强调在 GUI 导航中利用历史屏幕上下文，借助具备历史感知能力的视觉编码器和高效重采样模块提升对变化界面的理解。此外，屏幕视觉的感知可能因包含敏感信息而引发隐私问题。

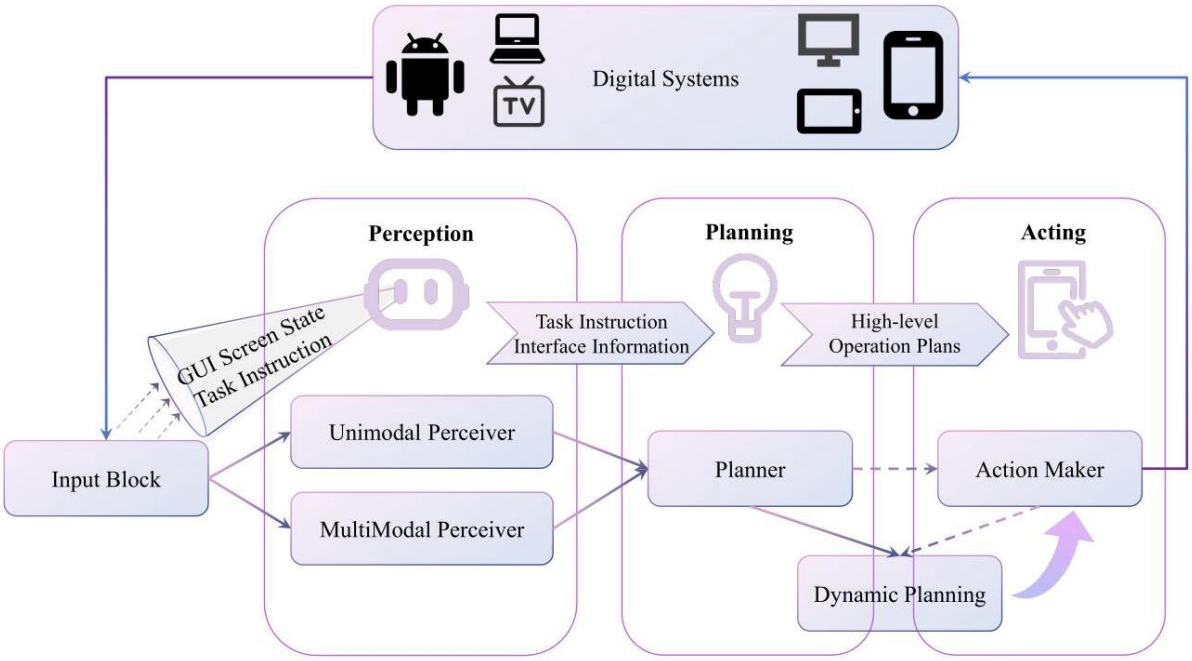

Figure 1: Architecture of GUI agents powered by (Multi-modal) Large Language Models. Comprising three interconnected modules: Perception, Planning, and Acting. The Perception module is responsible for perceiving and understanding the GUI state, the Planning module formulates high-level action strategies, and the Acting

module executes operations, collectively enabling intelligent interaction with the GUI.

图 1: 由 (多模态) 大语言模型驱动的 GUI 代理架构。由感知、规划和执行三个相互连接的模块组成。感知模块负责感知和理解 GUI 状态，规划模块制定高层次的行动策略，执行模块负责操作执行，共同实现对 GUI 的智能交互。

4.2 Planning

4.2 规划

In modern (M)LLM-driven GUI agent architectures, the Planning module is responsible for translating perceived interface information and user task requirements into concrete action plans. With the development of VLMs, recent studies have explored the potential of using these models for planning and reasoning. RL4VLM[32] introduced an approach that integrates reinforcement learning with chain-of-thought (CoT) reasoning in VLMs, where the model first outputs intermediate reasoning steps before the final action, and is fine-tuned using environment-based reward signals. This method employs CoT reasoning to guide the VLM in generating intermediate reasoning steps, leading to the final text-based action output. To further refine the planning process, Mobile-Agent-v2[25] introduced an additional module within their multi-agent framework to generate more detailed execution plans. The system decouples the roles of the Planner, Decision Maker, and Reflector: the Planner outlines task progression to streamline navigation based on prior actions. The Reflector reviews execution outcomes after each step, providing feedback to dynamically adjust the plan. The recent dynamic planning method, D-PoT [37] , breaks the traditional two-stage process of pre-defined planning followed by execution and feedback, integrating planning and feedback into a unified process: Following each action, the model captures the latest screen state and execution history, continuously revising the action sequence in real time until the task is successfully completed.

在现代 (多模态) 大语言模型驱动的 GUI 代理架构中，规划模块负责将感知到的界面信息和用户任务需求转化为具体的行动计划。随着视觉语言模型 (VLMs) 的发展，近期研究探索了利用这些模型进行规划和推理的潜力。RL4VLM[32] 提出了一种将强化学习与 VLM 中的链式思维 (CoT) 推理相结合的方法，模型先输出中间推理步骤再给出最终动作，并通过基于环境的奖励信号进行微调。该方法利用链式思维推理引导 VLM 生成中间推理步骤，最终输出基于文本的动作。为进一步优化规划过程，Mobile-Agent-v2[25] 在其多代理框架中引入了额外模块以生成更详细的执行计划。系统将规划者、决策者和反思者的角色解耦: 规划者概述任务进展以基于先前动作简化导航，反思者在每步执行后审查结果，提供反馈以动态调整计划。最新的动态规划方法 D-PoT [37] 打破了传统的预定义规划后执行与反馈的两阶段流程，将规划与反馈整合为统一过程: 每次动作后，模型捕获最新屏幕状态和执行历史，实时持续修订动作序列，直至任务成功完成。

4.3 Acting

4.3 执行

The GUI Acting module is responsible for converting high-level operation plans generated by the planning module into executable interactive actions on the interface, such as clicking, swiping, and text input [GUI-Odyssey [11] ; AppAgent [35] . The design of this module directly affects the execution efficiency and task success rate of

agents in real-world environments. UI-R1 [13] introduces a carefully designed unified reward mechanism and integrates multimodal inputs to guide the model in learning generalizable operational strategies for mobile interfaces. This method significantly improves performance in action type recognition and target localization, especially in identifying action types and locating UI elements. Subsequently, dynamic decision-making and history-aware mechanisms were incorporated into the Acting module to better handle long-horizon tasks and environmental changes. For instance, the D-PoT framework [37] incorporates the concept of "Dynamic Planning of Thoughts" into the action generation process. After each action execution, the agent updates its remaining plan based on the latest interface feedback and execution history, enabling "plan-as-you-execute" behavior. This mechanism not only enhances adaptability in execution but also significantly improves the completion rate of complex tasks. To further improve generalization and cross-platform adaptability, GUI-R1 [14] proposed the Group Relative Policy Optimization (GRPO) algorithm, which models shared strategies across tasks, enabling a single model to be applicable across platforms such as Windows, Linux, Android, and Web. Auto-GUI [38] introduces a multi-modal chain-style action generation mechanism, combining visual and textual information to interact directly with the interface, thus avoiding the need for environment parsing or reliance on APIs. Overall, the Acting module is evolving from static mapping to an advanced module equipped with dynamic planning, history awareness, cross-platform adaptability, and end-to-end reasoning capabilities, with reinforcement learning, multimodal fusion, and module coordination becoming its core driving forces.

GUI 执行模块负责将规划模块生成的高层操作计划转换为界面上的可执行交互动作，如点击、滑动和文本输入 [GUI-Odyssey [11] ； AppAgent [35]]。该模块的设计直接影响代理在真实环境中的执行效率和任务成功率。UI-R1 [13] 引入了精心设计的统一奖励机制，并融合多模态输入，引导模型学习适用于移动界面的通用操作策略。该方法显著提升了动作类型识别和目标定位的性能，尤其是在动作类型识别和 UI 元素定位方面。随后，动态决策和历史感知机制被引入执行模块，以更好地应对长时任务和环境变化。例如，D-PoT 框架 [37] 将“动态思维规划”(Dynamic Planning of Thoughts) 概念融入动作生成过程中。每次动作执行后，代理根据最新的界面反馈和执行历史更新剩余计划，实现“边执行边规划”的行为。该机制不仅增强了执行的适应性，还显著提升了复杂任务的完成率。为进一步提升泛化能力和跨平台适应性，GUI-R1 [14] 提出了群体相对策略优化 (Group Relative Policy Optimization, GRPO) 算法，建模任务间共享策略，使单一模型可适用于 Windows、Linux、Android 和 Web 等多个平台。Auto-GUI [38] 引入了多模态链式动作生成机制，结合视觉和文本信息直接与界面交互，避免了环境解析或依赖 API 的需求。总体来看，执行模块正从静态映射向具备动态规划、历史感知、跨平台适应和端到端推理能力的高级模块演进，强化学习、多模态融合和模块协同成为其核心驱动力。

5 (M)LLM-based GUI Agent Taxonomy

5 基于 (多模态) 大语言模型的 GUI 代理分类

With the development of MLLMs, the perception mechanisms of GUI agents have evolved from single-modality (e.g., text) to multimodal (e.g., a combination of vision and language). At the same time, training paradigms have diversified-from simple prompt-based guidance, to supervised fine-tuning, and further to RL-based policy optimization-providing strong technical support for GUI agents in diverse application scenarios. Figure 2 shows the taxonomy of GUI agents enhanced by foundation models.

随着多模态大语言模型 (MLLMs) 的发展，GUI 代理的感知机制已从单一模态 (如文本) 演进为多模态 (如视觉与语言的结合)。同时，训练范式也日益多样化——从简单的基于提示的引导，到监督微调，再到基于强化学习的策略优化——为 GUI 代理在多样化应用场景中提供了强有力的技术支持。图 2 展示了基于基础模型增强的 GUI 代理分类体系。

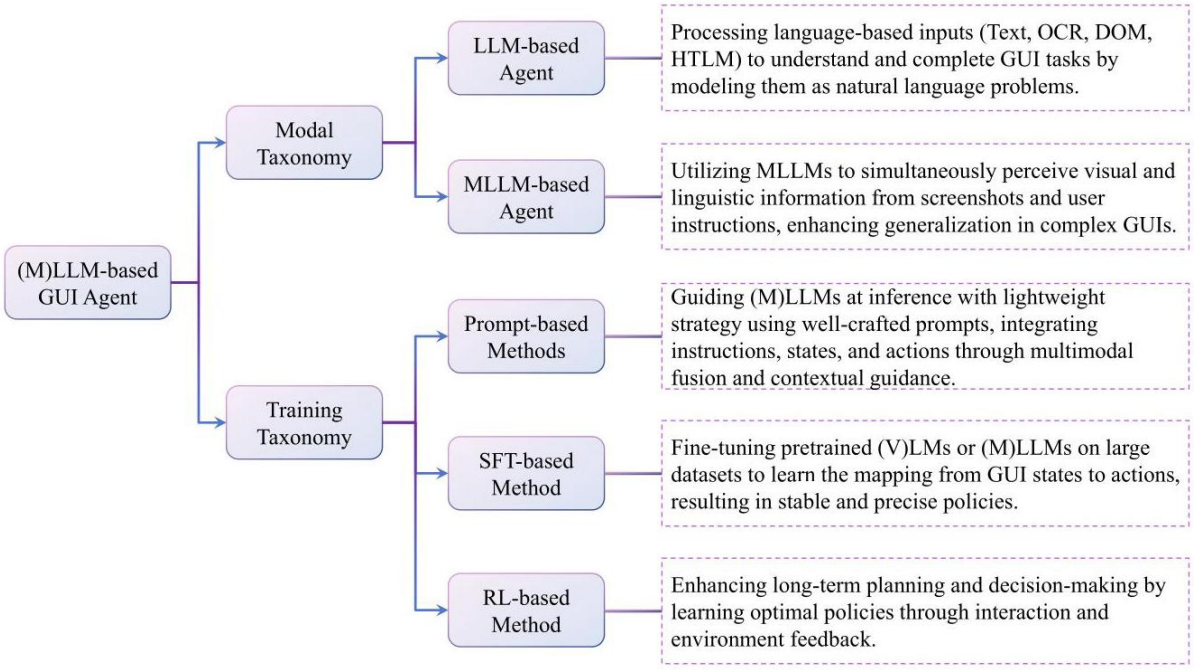

Figure 2: Taxonomy of (M)LLM-based GUI Agents. This figure presents a classification of GUI agents based on their underlying model modalities and training methodologies.

图 2:(多模态) 大语言模型基础的 GUI 代理分类。该图展示了基于模型模态和训练方法对 GUI 代理的分类。

5.1 Modal Taxonomy

5.1 模态分类

5.1.1 LLM-based Agent

5.1.1 基于大语言模型 (LLM) 的代理

Early GUI agents primarily relied on LLMs to perform tasks. These approaches were unimodal, as the agents could only process language-based inputs such as user task descriptions, screen text extracted via OCR, or structured DOM/HTML data to understand and complete tasks. This typically required GUI agents to first convert the screen content into text-based inputs. Therefore, during the perception stage, the screen had to be parsed and translated into a set of object descriptions. The core idea behind such methods is to model GUI tasks as problems of

natural language understanding and generation. The LLM learns to select appropriate actions in different contexts through prompting, fine-tuning, or reinforcement learning. Socratic Models [31] proposed transforming structured environmental data (e.g., HTML trees of web pages) into natural language prompts, from which the LLM generates operation steps, thereby mapping ”language to action”. We-bGPT [15] integrated LLMs with search engine APIs to perform web-based question answering tasks. While LLM-based approaches demand relatively low data and deployment costs, they struggle to directly handle visual signals (e.g., button shapes, colors), have limited capacity to model interface structures, and thus suffer from poor generalization, making them difficult to apply to complex real-world graphical interfaces.

早期的 GUI 代理主要依赖大语言模型 (LLM) 执行任务。这些方法属于单模态，因为代理只能处理基于语言的输入，如用户任务描述、通过 OCR 提取的屏幕文本或结构化的 DOM/HTML 数据，以理解并完成任务。通常，这要求 GUI 代理首先将屏幕内容转换为基于文本的输入。因此，在感知阶段，必须解析屏幕并将其转化为一组对象描述。这类方法的核心思想是将 GUI 任务建模为自然语言理解与生成问题。LLM 通过提示、微调或强化学习，学习在不同上下文中选择合适的动作。Socratic Models [31] 提出将结构化环境数据 (如网页的 HTML 树) 转化为自然语言提示，由 LLM 生成操作步骤，从而实现“语言到动作”的映射。We-bGPT [15] 将 LLM 与搜索引擎 API 结合，执行基于网页的问答任务。尽管基于 LLM 的方法对数据和部署成本要求较低，但它们难以直接处理视觉信号 (如按钮形状、颜色)，对界面结构的建模能力有限，因而泛化能力较差，难以应用于复杂的真实图形界面。

5.1.2 MLLM-based Agent

5.1.2 基于多模态大语言模型 (MLLM) 的代理

GUI interfaces are essentially multimodal interactive systems, consisting of visual components such as buttons, menus, and input boxes, which may not contain easily extractable structured textual information. As a result, traditional LLM-based approaches heavily rely on OCR and DOM structures, making it difficult to generalize to real-world apps or web applications. With the rapid development of MLLMs, GUI agents powered by MLLMs have increasingly become a focus of research in this field. Compared to traditional language-only methods, MLLMs can simultaneously perceive both visual and linguistic information, allowing them to complete tasks directly based on screen captures (or video frames) and textual instructions. This capability significantly enhances agents' generalization and adaptability in complex GUI environments. For example, WebGUM [6] introduced a multimodal task decomposition capability, where the model takes screen images and task instructions as input and performs staged perception, intent reasoning, and action generation, achieving notable results on MiniWoB++ and real-world web tasks. SeeAct [39] introduces a vision-language agent capable of visual context awareness and action localization, using MLLMs like GPT- 4V to reason over screen images and generate semantically relevant click or input actions. UI-TARS [20] proposed an agent architecture that combines native system operations with MLLM perceptual capabilities, emphasizing high-precision recognition and decision-making for complex GUI components in the Android system. Using screen snapshots and task descriptions, the model applies vision-language reasoning for holistic perception and operational planning.

图形用户界面 (GUI) 本质上是多模态交互系统，由按钮、菜单和输入框等视觉组件组成，这些组件可能不包含易于提取的结构化文本信息。因此，传统的基于大语言模型 (LLM) 的方法严重依赖光学字符识别 (OCR) 和文档对象模型 (DOM) 结构，难以推广到实际应用程序或网页应用中。随着多模态大语言模型 (MLLM) 的快速发展，由 MLLM 驱动的 GUI 智能体日益成为该领域的研究焦点。与传统的仅基于语言的方法相比，MLLM 可以同时感知视觉和语言信息，使其能够直接根据屏幕截图 (或视频帧) 和文本指令完成任务。这种能力显著增强了智能体在复杂 GUI 环境中的泛化能力和适应性。例如，WebGUM [6] 引入了多模态任务分解能力，该模型将屏幕图像和任务指令作为输入，进行分阶段感知、意图推理和动作生成，在 MiniWoB++ 和实际网页任务中取得了显著成果。SeeAct [39] 引入了一种能够进行视觉上下文感知和动作定位的视觉语言智能体，使用如 GPT - 4V 这样的 MLLM 对屏幕图像进行推理，并生成语义相关的点击或输入动作。UI - TARS [20] 提出了一种将原生系统操作与 MLLM 感知能力相结合的智能体架构，强调对安卓系统中复杂 GUI 组件进行高精度识别和决策。该模型利用屏幕快照和任务描述，应用视觉语言推理进行整体感知和操作规划。

5.2 Training Taxonomy

5.2 训练分类法

5.2.1 Prompt-based Methods

5.2.1 基于提示的方法

Prompt-based methods offer a lightweight strategy for building GUI agents by harnessing the pretrained strengths of (M)LLMs. Through well-crafted prompts, these models are guided to perform tasks at inference time without parameter tuning or high compute resources. The core of prompt-based methods lies in structured prompt engineering, which integrates user instructions, environment states, and action specifications into model-understandable inputs through multimodal fusion and contextual guidance. Zeng et al.[2022] proposed the Socratic Models framework [31] , which uses multi-round prompts to convert visual or structured data (e.g., HTML trees or OCR-extracted text) into natural language, enabling LLMs to output actions and complete zero-shot multimodal tasks such as video QA and robot planning. Subsequently, researchers extended prompt-based approaches to real-world web automation. WebAgent [7] incorporates HTML summaries and Python subprogram generation in its prompts, breaking long HTML documents into task-relevant segments and generating executable code for end-to-end web automation. With the rapid advancement of MLLMs, prompt-based approaches have further evolved to take screen-shots directly as input. UFO2[34] integrates VLMs and LLMs to construct multimodal inputs (screenshots + Accessibility API control attributes), generating cross-platform action sequences through hierarchical prompts to support automation across Windows, web, and mobile platforms.

基于提示的方法通过利用 (多模态) 大语言模型 ((M)LLM) 的预训练优势，为构建 GUI 智能体提供了一种轻量级策略。通过精心设计的提示，引导这些模型在推理时执行任务，而无需进行参数调整或使用高计算资源。基于提示的方法的核心在于结构化提示工程，它通过多模态融合和上下文引导，将用户指令、环境状态和动作规范整合为模型可理解的输入。曾等人 [2022] 提出了苏格拉底模型框架 [31]，该框架使用多轮提示将视觉或结构化数据 (如 HTML 树或 OCR 提取的文本) 转换为自然语言，使大语言模型能够输出动作并完成视频问答和机器人规划等零样本多模态任务。随后，研究人员将基于提示的方法扩展到实际的网页自动化中。WebAgent [7] 在其提示中纳入了 HTML 摘要和 Python 子程序生成，将长 HTML 文档拆分为与任务相关的片段，并生成可执行代码以实现端到端的网页自动化。随着 MLLM 的快速发展，基于提示的方法进一步发展为直接将屏幕截图作为输入。UFO2[34] 将视觉语言模型 (VLM) 和大语言模型 (LLM) 集成以构建多模态输入 (屏幕截图 + 辅助功能应用程序编程接口控制属性)，通过分层提示生成跨平台动作序列，以支持 Windows、网页和移动平台的自动化操作。

5.2.2 SFT-based Method

5.2.2 基于监督微调 (SFT) 的方法

Supervised Fine-Tuning (SFT) is a widely adopted training paradigm for GUI agents. Unlike zero-shot or few-shot learning, this approach fine-tunes pretrained (V)LMs or MLLMs on large annotated datasets, enabling models to learn the mapping from GUI states to concrete actions, resulting in more stable and precise behavioral policies tailored to specific GUI tasks. SFT offers advantages such as a stable training process and effective specialization for target tasks. Auto-GUI [38] applies supervised fine-tuning on high-quality GUI interaction datasets (including screenshots, task descriptions, action histories, and annotated next-step actions), using GUI states and language tasks as input to predict the next action-such as action type, click coordinates, and text input. This method achieved remarkable performance on several GUI agent benchmarks, such as AITW and Mind2Web. Some efforts focus on building general-purpose GUI understanding models first, and then applying SFT on downstream tasks. For example, GUICourse [4] pretrains general VLMs (like Qwen-VL and MiniCPM-V) on the GUIEnv dataset to enhance OCR and element localization; Falcon-UI [3] is pretrained on the Insight-UI dataset to enhance the model's understanding of GUI environments. This model are then fine-tuned on Android and Web GUI datasets (e.g., AITW, AITZ, Android Control, and Mind2Web), achieving performance comparable to larger models like Qwen2VL; InfiGUIAgent [10] proposes a two-stage supervised fine-tuning framework. The first stage boosts GUI perception and localization, while the second uses synthetic data to teach reasoning and self-correction in complex multistep scenarios. Additionally, many works focus on dataset construction and task diversity. TongUI [33] constructs the GUI-Net dataset, containing 143K interaction records, by automatically crawling GUI operation traces from online tutorials. In conclusion, SFT-based methods continuously improve model robustness and task success rates in real GUI environments by expanding data scale and diversity and adopting staged training processes.

监督微调 (Supervised Fine-Tuning，SFT) 是图形用户界面 (GUI) 代理广泛采用的训练范式。与零样本或少样本学习不同，该方法在大规模标注数据集上对预训练的视觉语言模型 (VLMs) 或多模态大模型 (MLLMs) 进行微调，使模型能够学习从 GUI 状态到具体操作的映射，从而形成更稳定且精确的行为策略，针对特定 GUI 任务进行定制。SFT 具有训练过程稳定和目标任务专门化效果显著的优势。Auto-GUI [38] 在高质量的 GUI 交互数据集 (包括截图、任务描述、操作历史和标注的下一步操作) 上应用监督微调，利用 GUI 状态和语言任务作为输入，预测下一步操作，如操作类型、点击坐标和文本输入。该方法在多个 GUI 代理基准测试中取得了显著表现，如 AITW 和 Mind2Web。一些研究致力于先构建通用的 GUI 理解模型，再在下游任务上应用 SFT。例如，GUICourse [4] 在 GUIEnv 数据集上预训练通用视觉语言模型 (如 Qwen-VL 和 MiniCPM-V)，以增强光学字符识别 (OCR) 和元素定位能力；Falcon-UI [3] 在 Insight-UI 数据集上预训练，以提升模型对 GUI 环境的理解能力。随后，该模型在 Android 和 Web GUI 数据集 (如 AITW、AITZ、Android Control 和 Mind2Web) 上进行微调，性能可与更大型模型如 Qwen2VL 相媲美；InfiGUIAgent [10] 提出了两阶段监督微调框架，第一阶段提升 GUI 感知和定位能力，第二阶段利用合成数据教授复杂多步骤场景中的推理与自我纠正。此外，许多工作关注数据集构建和任务多样性。TongUI [33] 通过自动爬取在线教程中的 GUI 操作轨迹，构建了包含 14.3 万条交互记录的 GUI-Net 数据集。综上所述，基于 SFT 的方法通过扩展数据规模和多样性及采用分阶段训练流程，持续提升模型在真实 GUI 环境中的鲁棒性和任务成功率。

5.2.3 RL-based Method

5.2.3 基于强化学习的方法

RL is primarily used in GUI agent training to enhance long-term planning and decision-making capabilities in complex task scenarios. Recent research highlights RL's crucial role in enabling agents to learn optimal policies via interaction, particularly in handling intricate, multi-step tasks to boost decision-making and task success. Unlike SFT, which depends on human-labeled expert data, RL explores and learns policies through trial with environment feedback. This makes it well-suited for real GUI scenarios with long task sequences, ambiguous goals, and sparse feedback.

强化学习 (RL) 主要用于 GUI 代理训练，以增强其在复杂任务场景中的长期规划和决策能力。近期研究强调了 RL 在通过交互学习最优策略中的关键作用，尤其是在处理复杂多步骤任务时，提升决策质量和任务成功率。与依赖人工标注专家数据的 SFT 不同，RL 通过环境反馈进行试错探索和策略学习，因而非常适合具有长任务序列、目标模糊和反馈稀疏的真实 GUI 场景。

Recently, many works have integrated RL into MLLMs. RL4VLM [32] applies Proximal Policy Optimization (PPO) to VLMs, generating CoT reasoning at each step, converting the output into actions, receiving environment rewards and fine-tuning the VLMs accordingly. To balance data efficiency and cross-platform generalization, GUI-R1 [14] employs RL to enhance the GUI operation capabilities of VLMs in complex real-world scenarios. It incorporates DeepSeek-R1-style "regularized rewards" into GUI action prediction, using a unified action space and the Group Relative Policy Optimization (GRPO) algorithm to train across platforms such as Windows, Linux, MacOS, Android, and Web. For long-horizon tasks spanning multiple apps or APIs, short-term RL often suffers from local optima. Chen et al. [2025] propose LOOP [2] , a memory-efficient variant of PPO that trains IDAs (Intelligent Digital Agents) directly in target environments. It avoids value networks and requires only a single LLM instance in memory to perform end-to-end RL training within the AppWorld environment. Due to the high cost and brittleness of real environment interaction, environment-free RL approaches have emerged. Zheng et al. (2025)

propose VEM (Value Environment Model) [40] , where a value function is trained on offline GUI interaction data to estimate long-term returns for arbitrary state-action pairs, serving as ”virtual rewards” to guide policy optimization. In addition, asynchronous interaction approaches have been incorporated into RL training [27][26] . Trajectories generated by GUI agents in simulated environments are stored in a replay buffer, serving as data for subsequent online model training.

近年来, 许多研究将强化学习 (RL) 整合进多模态大语言模型 (MLLMs)。RL4VLM [32] 将近端策略优化 (Proximal Policy Optimization, PPO) 应用于视觉语言模型 (VLMs)，在每一步生成链式推理 (CoT)，将输出转换为动作，接收环境奖励并据此微调 VLMs。为平衡数据效率与跨平台泛化，GUI-R1 [14] 利用 RL 提升 VLMs 在复杂真实场景中的图形用户界面 (GUI) 操作能力。它将 DeepSeek-R1 风格的“正则化奖励”引入 GUI 动作预测，采用统一动作空间和群体相对策略优化 (Group Relative Policy Optimization, GRPO) 算法，在 Windows、Linux、MacOS、Android 和 Web 等平台上进行训练。针对跨多个应用或 API 的长时任务，短期 RL 常陷入局部最优。陈等人 [2025] 提出 LOOP [2] ，一种内存高效的 PPO 变体，直接在目标环境中训练智能数字代理 (Intelligent Digital Agents, IDAs)。该方法避免了价值网络，仅需单个大语言模型实例即可在 AppWorld 环境中完成端到端的 RL 训练。鉴于真实环境交互成本高且脆弱，出现了无环境 RL 方法。郑等人 (2025) 提出价值环境模型 (Value Environment Model, VEM) [40] ，在离线 GUI 交互数据上训练价值函数，用以估计任意状态-动作对的长期回报，作为“虚拟奖励”指导策略优化。此外，异步交互方法也被引入 RL 训练中 [27][26] 。GUI 代理在模拟环境中生成的轨迹被存储于重放缓冲区，作为后续在线模型训练的数据。

6 Challenges

6 挑战

Despite the rapid progress in (M)LLM-based GUI agents, several challenges remain:

尽管基于 (多模态) 大语言模型的 GUI 代理取得了快速进展，但仍存在若干挑战:

Perception under Dynamic and Noisy Interfaces: Real-world GUIs are often dynamic, featuring pop-ups, advertisements, or frequent layout changes. Although modern GUI Perception module integrate multimodal inputs and history-aware modeling, accurately perceiving and adapting to these unpredictable variations remains a major difficulty.

动态且嘈杂界面下的感知: 真实世界的 GUI 常常动态变化，包含弹窗、广告或频繁的布局调整。尽管现代 GUI 感知模块融合了多模态输入和历史感知建模，准确感知并适应这些不可预测的变化仍是重大难题。

Long-Horizon Planning and Execution: GUI tasks typically require multiple sequential operations. Ensuring consistency across multi-step plans, especially when intermediate states deviate from expectations, challenges current Planning modules. Dynamic replanning strategies [37] have made progress, but long-horizon scalable and reliable reasoning remains open.

长时规划与执行:GUI 任务通常需要多步连续操作。确保多步计划的一致性，尤其是在中间状态偏离预期时，考验当前规划模块。动态重规划策略 [37] 虽有进展，但长时可扩展且可靠的推理仍未解决。

Data Efficiency and Generalization: SFT-based approaches heavily rely on large annotated datasets. Despite advances like GUI-R1[14] using small datasets, achieving strong generalization across different platforms (Windows, Android, Web) with minimal supervision is still an ongoing challenge.

数据效率与泛化: 基于监督微调 (SFT) 的方法高度依赖大规模标注数据。尽管如 GUI-R1[14] 等方法使用小规模数据取得进展，实现跨不同平台 (Windows、Android、Web) 强泛化且监督最小化仍是持续挑战。

Evaluation and Benchmarking: Current benchmarks like Mind2Web and AITW primarily focus on success rates. However, finer-grained evaluations (e.g., plan optimality, robustness under interface drift) and standardized human-in-the-loop assessments are urgently needed to fully measure agent capabilities.

评估与基准测试: 当前如 Mind2Web 和 AITW 等基准主要关注成功率。然而，更细粒度的评估 (如计划最优性、界面漂移下的鲁棒性) 及标准化的人机交互评估亟需建立，以全面衡量代理能力。

7 Conclusions

7 结论

This paper presents a structured summary of recent developments in GUI agents enhanced by foundation models and RL techniques. We first introduced the task formulation, key execution environments, and standard evaluation metrics. Then, we reviewed the modular architecture of GUI agents - covering Perception, Planning, and Acting - along with representative advances in each component. We also discussed three major training paradigms: Prompt-based methods for lightweight deployment, SFT-based methods for domain-specific adaptation, and RL-based methods for dynamic policy learning.

本文系统总结了基于基础模型和强化学习技术增强的 GUI 代理的最新进展。首先介绍了任务定义、关键执行环境及标准评估指标。随后回顾了 GUI 代理的模块化架构——涵盖感知、规划与执行——及各组件的代表性进展。还讨论了三大训练范式: 基于提示的方法实现轻量部署，基于监督微调的方法实现领域适应，以及基于强化学习的方法实现动态策略学习。

Through this review, we observe a clear trend: GUI agents are evolving from static rule-based systems toward adaptive, perception-driven, and reasoning-capable agents, powered by MLLMs and optimized via RL. Despite impressive gains, challenges in perception robustness, long-horizon reasoning, data efficiency, and evaluation remain to be addressed.

通过本综述，我们观察到一个明显趋势:GUI 代理正从静态规则系统向适应性强、感知驱动且具备推理能力的代理演进，这得益于多模态大语言模型的支持和强化学习的优化。尽管取得显著进展，感知鲁棒性、长时推理、数据效率及评估等挑战仍待解决。

Looking forward, integrating better semantic grounding, continual learning, human feedback, and asynchronous interaction will be crucial for building GUI agents that are truly capable of autonomously navigating complex, dynamic digital environments.

展望未来，整合更优的语义基础、持续学习、人类反馈及异步交互，将是构建真正能够自主导航复杂动态数字环境的 GUI 代理的关键。

References

参考文献

[1] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[2] Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive llm agents, 2025.

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, 和 Philipp Krähenbühl。用于长时程交互式大语言模型 (LLM) 代理的强化学习，2025 年。

[3] Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, Yuan Yao, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Guicourse: From general vision language models to versatile gui agents, 2024.

陈文彤，崔军博，胡金义，秦宇佳，方俊杰，赵越，王崇义，刘军，陈贵荣，霍玉鹏，姚远，林彦凯，刘知远，孙茂松。Guicourse: 从通用视觉语言模型到多功能 GUI 代理，2024 年。

[4] Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, Yuan Yao, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Guicourse: From general vision language models to versatile gui agents, 2024.

陈文彤，崔军博，胡金义，秦宇佳，方俊杰，赵越，王崇义，刘军，陈贵荣，霍玉鹏，姚远，林彦凯，刘知远，孙茂松。Guicourse: 从通用视觉语言模型到多功能 GUI 代理，2024 年。

[5] Sumit Kumar Dam, Choong Seon Hong, Yu Qiao, and Chaoning Zhang. A complete survey on llm-based ai chatbots, 2024.

Sumit Kumar Dam，Choong Seon Hong，乔宇，张朝宁。基于大语言模型 (LLM) 的 AI 聊天机器人全面综述，2024 年。

[6] Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixi-ang Shane Gu, and Izzeddin Gur. Multimodal web navigation with instruction-finetuned foundation models, 2024.

古田浩树，李光辉，Ofir Nachum，松尾丰，Aleksandra Faust，Shixi-ang Shane Gu，Izzeddin Gur。基于指令微调基础模型的多模态网页导航，2024 年。

[7] Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis, 2024.

Izzeddin Gur，古田浩树，Austin Huang，Mustafa Safdari，松尾丰，Douglas Eck，Aleksandra Faust。具备规划、长上下文理解和程序合成能力的真实世界网页代理，2024 年。

[8] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents, 2024.

洪文怡，王伟涵，吕庆松，徐嘉政，余文萌，季俊辉，王岩，王子涵，张宇轩，李娟子，徐斌，董宇霄，丁明，唐杰。Cogagent: 面向 GUI 代理的视觉语言模型，2024 年。

[9] Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration, 2018.

Evan Zheran Liu，Kelvin Guu，Panupong Pasupat，史天麟，Percy Liang。基于工作流引导探索的网页界面强化学习，2018 年。

[10] Yuhang Liu, Pengxiang Li, Zishu Wei, Congkai Xie, Xueyu Hu, Xinchen Xu, Shengyu Zhang, Xiaotian Han, Hongxia Yang, and Fei Wu. Infiguiagent: A multimodal generalist gui agent with native reasoning and reflection, 2025.

刘宇航，李鹏翔，魏子书，谢聪凯，胡雪宇，徐新晨，张胜宇，韩晓天，杨红霞，吴飞。Infiguiagent: 具备本地推理与反思能力的多模态通用 GUI 代理，2025 年。

[11] Quanfeng Lu, Wenqi Shao, Zitao Liu, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, Yu Qiao, and Ping Luo. Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices, 2024.

陆全峰，邵文琦，刘子涛，孟凡青，李博轩，陈博彤，黄思远，张凯鹏，乔宇，罗平。Gui Odyssey: 面向移动设备跨应用 GUI 导航的综合数据集，2024 年。

[12] Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. Omniparser for pure vision based gui agent, 2024.

陆亚东，杨建伟，沈业龙，Ahmed Awadallah。Omniparser: 基于纯视觉的 GUI 代理，2024 年。

[13] Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing action prediction of gui agents by reinforcement learning, 2025.

陆正熙，柴宇翔，郭雅轩，尹曦，刘亮，王浩，肖涵，任帅，熊冠京，李宏升。UI-R1: 通过强化学习提升 GUI 代理的动作预测，2025 年。

[14] Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. Gui-r1 : A generalist r1-style vision-language action model for gui agents, 2025.

罗润，王璐，何万伟，夏晓波。GUI-R1: 面向 GUI 代理的通用 R1 风格视觉语言动作模型，2025 年。

[15] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022.

中野礼一郎，Jacob Hilton，Suchir Balaji，Jeff Wu，Long Ouyang，Christina Kim，Christopher Hesse，Shantanu Jain，Vineet Kosaraju，William Saunders，姜旭，Karl Cobbe，Tyna Eloundou，Gretchen Krueger，Kevin Button，Matthew Knight，Benjamin Chess，John Schulman。WebGPT: 基于浏览器辅助的人类反馈问答系统，2022 年。

[16] Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, Xintong Li, Jing Shi, Hongjie Chen, Viet Dac Lai, Zhouhang Xie, Sungchul Kim, Ruiyi Zhang, Tong Yu, Mehrab Tanjim, Nesreen K. Ahmed, Puneet Mathur, Seunghyun Yoon, Lina Yao, Branislav Kveton, Thien Huu Nguyen, Trung Bui, Tianyi Zhou, Ryan A. Rossi, and Franck Dernoncourt. Gui agents: A survey, 2024.

Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, Xintong Li, Jing Shi, Hongjie Chen, Viet Dac Lai, Zhouhang Xie, Sungchul Kim, Ruiyi Zhang, Tong Yu, Mehrab Tanjim, Nesreen K. Ahmed, Puneet Mathur, Seunghyun Yoon, Lina Yao, Branislav Kveton, Thien Huu Nguyen, Trung Bui, Tianyi Zhou, Ryan A. Rossi, 和 Franck Dernoncourt. Gui agents: 一项综述，2024。

[17] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, 和 Michael S. Bernstein. 生成代理: 人类行为的交互模拟，2023。

[18] Chen Qian, Jiahao Li, Yufan Dang, Wei Liu, YiFei Wang, Zihao Xie, Weize Chen, Cheng Yang, Yingli Zhang, Zhiyuan Liu, and Maosong Sun. Iterative experience refinement of software-developing agents, 2024.

陈谦，李佳豪，邓宇凡，刘伟，王一飞，谢子豪，陈伟泽，杨成，张英丽，刘知远，和孙茂松。软件开发代理的迭代经验优化，2024。

[19] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, Proceedings of the 62nd

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15174-15186, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

陈谦，刘伟，刘宏章，陈诺，邓宇凡，李佳豪，杨成，陈伟泽，苏玉生，丛鑫，徐聚元，李大海，刘知远，和孙茂松。ChatDev: 用于软件开发的交流代理。收录于 Lun-Wei Ku, Andre Martins, 和 Vivek Srikumar 编辑的《第 62 届计算语言学协会年会论文集 (第一卷: 长篇论文)》，第 15174-15186 页，泰国曼谷，2024 年 8 月。计算语言学协会。

[20] Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, Xiaojun Xiao, Kai Cai, Chuang Li, Yaowei Zheng, Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng Liu, Feng Lin, Tao Peng, Xin Liu, and Guang Shi. Ui-tars: Pioneering automated gui interaction with native agents, 2025.

秦宇佳，叶一宁，方俊杰，王浩明，梁世豪，田志佐，张俊达，李佳豪，李云欣，黄世觉，钟万军，李宽业，杨佳乐，缪宇，林沃宇，刘龙翔，姜旭，马千里，李景宇，肖晓军，蔡凯，李闯，郑耀伟，金朝林，李晨，周晓，王敏超，陈浩力，李朝建，杨海华，刘海峰，林峰，彭涛，刘鑫，史光。Ui-tars: 开创性的本地代理自动化 GUI 交互，2025。

[21] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

Qwen, :, 安阳，杨宝松，张北辰，惠斌元，郑博，余博文，李成元，刘大义恒，黄飞，魏浩然，林欢，杨健，涂建宏，张建伟，杨建新，杨佳熙，周景仁，林俊阳，党凯，卢克明，包克勤，杨可欣，余乐，李梅，薛明峰，张培，朱琴，门锐，林润基，李天昊，唐天一，夏廷宇，任兴章，任轩成，范阳，苏阳，张一昌，万宇，刘玉琼，崔泽宇，张振儒，邱子涵。Qwen2.5 技术报告，2025。

[22] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, 和 Thomas Scialom。Toolformer: 语言模型自我学习使用工具，2023。

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

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

[24] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay

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

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

[25] Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration, 2024.

Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, 和 Jitao Sang。Mobile-agent-v2: 通过多智能体协作实现高效导航的移动设备操作助手，2024 年。

[26] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. Frontiers of Computer Science, 18(6), March 2024.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, 和 Jirong Wen。基于大型语言模型的自主智能体综述。《计算机科学前沿》，18(6)，2024 年 3 月。

[27] Taiyi Wang, Zhihao Wu, Jianheng Liu, Jianye Hao, Jun Wang, and Kun Shao. Distrl: An asynchronous distributed reinforcement learning framework for on-device control agents, 2025.

Taiyi Wang, Zhihao Wu, Jianheng Liu, Jianye Hao, Jun Wang, 和 Kun Shao。Distrl: 一种用于设备端控制智能体的异步分布式强化学习框架，2025 年。

[28] Ziwei Wang, Weizhi Chen, Leyang Yang, Sheng Zhou, Shengchu Zhao, Hanbei Zhan, Jiongchao Jin, Liangcheng Li, Zirui Shao, and Jiajun Bu. Mp-gui: Modality perception with mllms for gui understanding, 2025.

王紫薇、陈伟智、杨乐阳、周盛、赵盛初、詹汉北、金炯超、李良成、邵子锐和卜佳军。Mp - gui: 利用多模态大语言模型 (MLLMs) 进行图形用户界面 (GUI) 理解的模态感知，2025 年。

[29] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

杰森·魏、王学志、戴尔·舒尔曼斯、马腾·博斯马、布莱恩·伊克特、夏飞、埃德·池、乐存和周丹尼。思维链提示在大语言模型中引发推理，2023 年。

[30] Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions, 2023.

戈库尔·延杜里、拉马林甘·M、切马拉尔·塞尔维·G、苏普丽亚·Y、高塔姆·斯里瓦斯塔瓦、普拉文·库马尔·雷迪·马迪昆塔、迪普蒂·拉杰·G、鲁特维杰·H·贾维里、普拉巴德维·B、王韦正、阿塔纳西奥斯·V·瓦西拉科斯和蒂帕·雷迪·加德卡卢。生成式预训练变换器: 关于使能技术、潜在应用、新兴挑战和未来方向的全面综述，2023 年。

[31] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language, 2022.

安迪·曾、玛丽亚·阿塔里安、布莱恩·伊克特、克日什托夫·乔罗曼斯基、阿德里安·黄、斯特凡·韦尔克、费德里科·通巴里、阿维克·普罗希特、迈克尔·柳、维卡斯·辛德瓦尼、约翰尼·李、文森特·范霍克和皮特·弗洛伦斯。苏格拉底模型: 用语言组合零样本多模态推理，2022 年。

[32] Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, and Sergey Levine. Fine-tuning large vision-language models as decision-making agents via reinforcement learning, 2024.

翟跃翔、白浩、林梓鹏、潘佳怡、童圣邦、周逸飞、阿莱恩·苏尔、谢赛宁、杨立昆、马毅和谢尔盖·莱文。通过强化学习将大型视觉 - 语言模型微调为决策代理，2024 年。

[33] Bofei Zhang, Zirui Shang, Zhi Gao, Wang Zhang, Rui Xie, Xiaojian Ma, Tao Yuan, Xinxiao Wu, Song-Chun Zhu, and Qing Li. Tongui: Building generalized gui agents by learning from multimodal web tutorials, 2025.

张博飞、尚子锐、高志、张旺、谢锐、马晓建、袁涛、吴新晓、朱松纯和李清。Tongui: 通过从多模态网络教程中学习构建通用图形用户界面代理，2025 年。

[34] Chaoyun Zhang, He Huang, Chiming Ni, Jian Mu, Si Qin, Shilin He, Lu Wang, Fangkai Yang, Pu Zhao, Chao Du, Liqun Li, Yu Kang, Zhao Jiang, Suzhen Zheng, Rujia Wang, Jiaxu Qian, Minghua Ma, Jian-Guang Lou, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Ufo2: The desktop agentos, 2025.

张朝云、黄河、倪驰鸣、穆健、秦思、何世林、王璐、杨方凯、赵普、杜超、李立群、康宇、蒋兆、郑素珍、王汝佳、钱佳旭、马明华、楼剑光、林庆伟、萨拉万·拉杰莫汉和张冬梅。Ufo2: 桌面代理操作系统，2025 年。

[35] Chi Zhang, Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users, 2023.

张弛、杨钊、刘家璇、韩雨成、陈鑫、黄泽彪、傅斌和余刚。Appagent: 作为智能手机用户的多模态代理，2023 年。

[36] J. Zhang, Y. Yu, M. Liao, W. Li, J. Wu, and Z. Wei. Ui-hawk: Unleashing the screen stream understanding for gui agents. Preprints, 2024.

张 J、于 Y、廖 M、李 W、吴 J 和魏 Z。Ui - hawk: 释放图形用户界面代理的屏幕流理解能力。预印本，2024 年。

[37] Shaoqing Zhang, Zhuosheng Zhang, Kehai Chen, Xinbei Ma, Muyun Yang, Tiejun Zhao, and Min Zhang. Dynamic planning for llm-based graphical user interface automation, 2024.

张绍清、张卓生、陈克海、马新贝、杨慕云、赵铁军和张敏。基于大语言模型 (LLM) 的图形用户界面自动化动态规划，2024 年。

[38] Zhuosheng Zhang and Aston Zhang. You only look at screens: Multimodal chain-of-action agents, 2024.

张卓生和阿斯顿·张。你只需看屏幕: 多模态行动链代理，2024 年。

[39] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v(ision) is a generalist web agent, if grounded, 2024.

郑博远、苟博宇、吉尔·吉亨、孙欢和苏羽。如果有基础，GPT - 4v(视觉) 是一个通用网络代理，2024 年。

[40] Jiani Zheng, Lu Wang, Fangkai Yang, Chaoyun Zhang, Lingrui Mei, Wenjie Yin, Qing-wei Lin, Dongmei Zhang, Saravan Rajmohan, and Qi Zhang. Vem: Environment-free exploration for training gui agent with value environment model, 2025.

郑佳妮、王璐、杨方凯、张朝云、梅凌锐、尹文杰、林庆伟、张冬梅、萨拉万·拉杰莫汉和张琦。Vem: 使用价值环境模型进行无环境探索以训练图形用户界面代理，2025 年。