OS Agents: A Survey on MLLM-based Agents for Computer, Phone and Browser Use

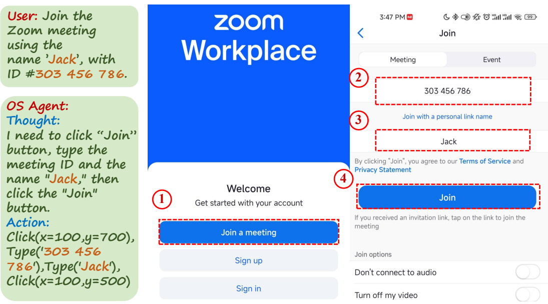
**Xueyu Hu[1, †], Tao Xiong[1, ‡], Biao Yi[1, ‡], Zishu Wei[1, ‡],
Ruixuan Xiao[1], Yurun Chen[1], Jiasheng Ye[2], Meiling Tao[3],
Xiangxin Zhou[4, 5], Ziyu Zhao[1], Yuhuai Li[1], Shengze Xu[6],
Shenzhi Wang[7], Xinchen Xu[1], Shuofei Qiao[1], Zhaokai Wang[8],
Kun Kuang[1], Tieyong Zeng[6], Liang Wang[4, 5], Jiwei Li[1], Yuchen Eleanor Jiang[3],
Wangchunshu Zhou[3], Guoyin Wang[9], Keting Yin[1], Zhou Zhao[1],
Hongxia Yang[10], Fan Wu[8], Shengyu Zhang[1, *], Fei Wu[1]**

[1]Zhejiang University, [2]Fudan University, [3]OPPO AI Center,
[4]University of Chinese Academy of Sciences,
[5]Institute of Automation, Chinese Academy of Sciences,
[6]The Chinese University of Hong Kong, [7]Tsinghua University,
[8]Shanghai Jiao Tong University, [9]01.AI, [10]The Hong Kong Polytechnic University

Correspondence: {huxueyu, sy_zhang}@zju.edu.cn
https://os-agent-survey.github.io/

Abstract

The dream to create AI assistants as capable and versatile as the fictional *J.A.R.V.I.S* from *Iron Man* has long captivated imaginations. With the evolution of (multimodal) large language models ((M)LLMs), this dream is closer to reality, as (M)LLM-based Agents using computers, mobile phones and web browsers by operating within the environments and interfaces (e.g., Graphical User Interface (GUI) and Command Line Interface (CLI)) provided by operating systems (OS) to automate tasks have significantly advanced. This paper presents a comprehensive survey on these advanced agents, designated as **OS Agents**. We begin by elucidating the fundamentals of OS Agents, exploring their key components and capabilities. We then examine methodologies for constructing OS Agents, focusing on domain-specific foundation models and agent frameworks. A detailed review of evaluation metrics and benchmarks highlights how OS Agents are assessed across diverse platforms and tasks. Finally, we discuss current challenges and identify promising directions for future research. An open-source GitHub repository is maintained as a dynamic resource to foster further innovation in this field.



User: Join the Zoom meeting using the name 'Jack', with ID #303 456 786.

OS Agent:
Thought:
I need to click "Join" button, type the meeting ID and the name "Jack," then click the "Join" button.
Action:
Click(x=100,y=700), Type('303 456 786'),Type('Jack'), Click(x=100,y=500)

zoom
Workplace
Welcome
Get started with your account
1
Join a meeting
Sign up
Sign in
3:47 PM
Join
Meeting
Event
2
303 456 786
3
Join with a personal link name
Jack
By clicking "Join", you agree to our Terms of Service and Privacy Statement
4
Join
If you received an invitation link, tap on the link to join the meeting
Join options
Don't connect to audio
Turn off my video

Figure 1: An example of OS Agents automatically joining a Zoom meeting on the user's phone as requested.

1 Introduction

Building a superintelligent AI assistant akin to *J.A.R.V.I.S.*[1] from the Marvel movie *Iron Man*, which assists *Tony Stark* in controlling various systems and automating tasks, has long been a human aspiration. These entities are recognized as **Operating System Agents (OS Agents)**, as they use computers, phones and browsers by operating within the environments and interfaces (e.g., Graphical User Interface (GUI) and Command Line Interface (CLI))) provided by operating systems (OS). OS Agents can complete tasks autonomously and

[†]Project Lead, [‡]Core Contributor, [*]Corresponding Author

[1] J.A.R.V.I.S. stands for "Just A Rather Very Intelligent System", a fictional AI assistant character from the Marvel Cinematic Universe. It appears in Iron Man (2008), The Avengers (2012), and other films, serving as Tony Stark's (Iron Man's) personal assistant and interface for his technology

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7436–7465
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

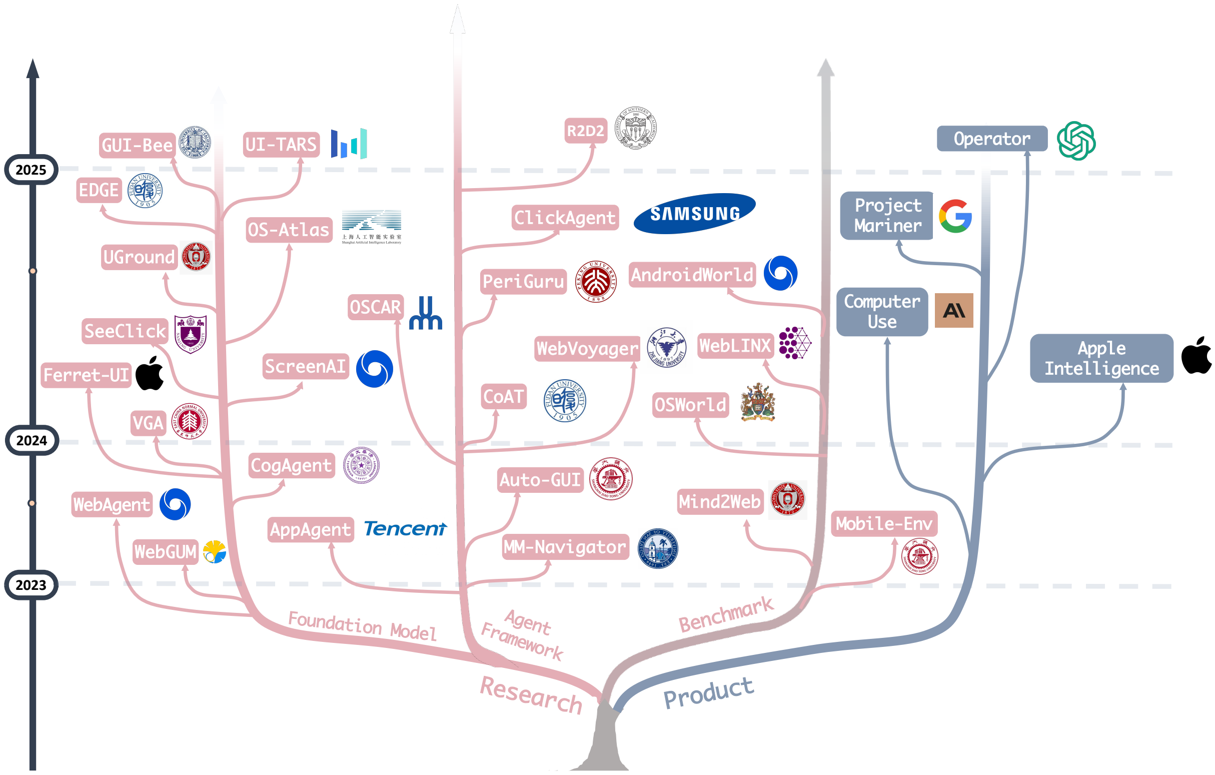


Figure 2: Part of academic research and commercial products of OS Agents in recent years. The figure is adapted from this repo.

have the potential to significantly enhance the lives of billions of users worldwide. Imagine a world where tasks such as online shopping, travel arrangements booking, and other daily activities could be seamlessly performed by these agents, thereby substantially increasing efficiency and productivity. In the past, virtual assistants such as Siri (Inc., 2024), Cortana (Research, 2024), Amazon Alexa (Google, 2024) and Google Assistant(Amazon, 2024) have already offered glimpses into this potential, but limitations in model capabilities such as contextual understanding (Tulshan and Dhage, 2019), have prevented these products from achieving widespread adoption and full functionality.

Fortunately, recent advancements in (multimodal) large language models ((M)LLMs), such as GPT (OpenAI) series models have ushered in a new era of possibilities for OS Agents. These models boast remarkable abilities, enabling OS Agents to better understand complex tasks and use computers, phones and browsers to execute. As illustrated in Figure 2, there has been a surge of OS Agents in both academic research and commercial products. A variety of works have been proposed to construct (M)LLM-based OS Agents by training domain-specific foundation models for OS Agents (Gur et al., 2023; You et al., 2025; Gou et al., 2024; Meng et al., 2024) and designing OS Agent frameworks (Zhang et al., 2023a; Yan et al., 2023; Ma et al., 2023; Zhang et al., 2024e). Meanwhile, a large number of works evaluating OS Agents (Xie et al., 2024b; Rawles et al., 2024a; Xing et al., 2024; Zhou et al., 2023a) have also been introduced. In the industry, notable products include the recently released Operator[2] by OpenAI, Computer Use[3] by Anthropic, Apple Intelligence[4] by Apple, and Project Mariner[5] by Google Deepmind. For instance, Computer Use leverages Claude (Anthropic, 2024b) to interact directly with users' computers, aiming for seamless task automation. Given these advancements and the growing body of work, it has become increasingly important to provide a comprehensive survey that consolidates the current state of research in this area.

We begin by discussing the fundamentals of OS

[2] https://operator.chatgpt.com/

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

[4] https://www.apple.com/apple-intelligence/

[5] https://deepmind.google/technologies/project-mariner/

Agents (§2). Next, we explore two critical aspects of constructing OS Agents (§3): (1) the development of domain-specific foundation models (§3.1); and (2) the building of effective agent frameworks around these models (§3.2). We also review the evaluation metrics (§4.1) and benchmarks (§4.2) commonly used to assess the performance of OS Agents. Additionally, we analyze existing commercial products of OS Agents (§5). Finally, we discuss the challenges and future directions for OS Agents (§6).

2 Fundamental of OS Agents

OS Agents are specialized AI agents that leverage the environment, input and output interfaces provided by operating systems to generally use computers, mobile phones and web browsers in response to user-defined goals. These agents are designed to automate tasks executed within the OS, leveraging the exceptional understanding and generative capabilities of (M)LLMs to enhance user experience and operational efficiency. To achieve this, OS Agents are based on several key components and necessitate some core capabilities discussed in the following.

2.1 Key Component

Environment. The environment for OS Agents refers to the platforms in which they operate, including computers, phones and browsers. OS Agents interact with these diverse environments to perform tasks, gather feedback, and adapt to their unique characteristics. We refer readers to §4.2 for detailed discussion.

Observation Space. The observation space encompasses the information OS Agents can access about the system's state and user activities. Observation includes capturing information from the OS, such as screen images, or textual data, such as the description of the screen and the HTML code in web-based contexts. Further details are elaborated in §3.2.1.

Action Space. The action space defines the set of interactions through which OS Agents manipulate the environment using the input interfaces provided by the OS. These actions can be broadly categorized into input operations, navigation operations and extended operations. A comprehensive discussion can be found in §3.2.4.

2.2 Capability

Understanding. A fundamental capability of OS Agents is comprehending complex OS environments. These environments encompass a diverse array of data formats, including HTML code (Gur et al., 2023; Lai et al., 2024) and GUIs captured in screenshots (Nong et al., 2024; Wu et al., 2024f). The complexity escalates with length code with sparse information, high-resolution interfaces cluttered with minuscule icons, small text, and densely packed elements (He et al., 2024a; Hong et al., 2024; You et al., 2025). Such environments challenge the agents' perceptual abilities and demand advanced contextual comprehension.

Planning. Planning (Huang and Chang, 2023; Zhang et al., 2024i; Huang et al., 2024b) is a crucial capability of OS Agents, enabling them to decompose complex tasks into manageable sub-tasks and devise sequences of actions to achieve specific goals (Wu et al., 2024e; Gao et al., 2023). Planning within operating systems often requires agents to dynamically adjust plans based on environmental feedback and historical actions (Zhang and Zhang, 2023; Wang and Liu, 2024; Kim et al., 2024a).

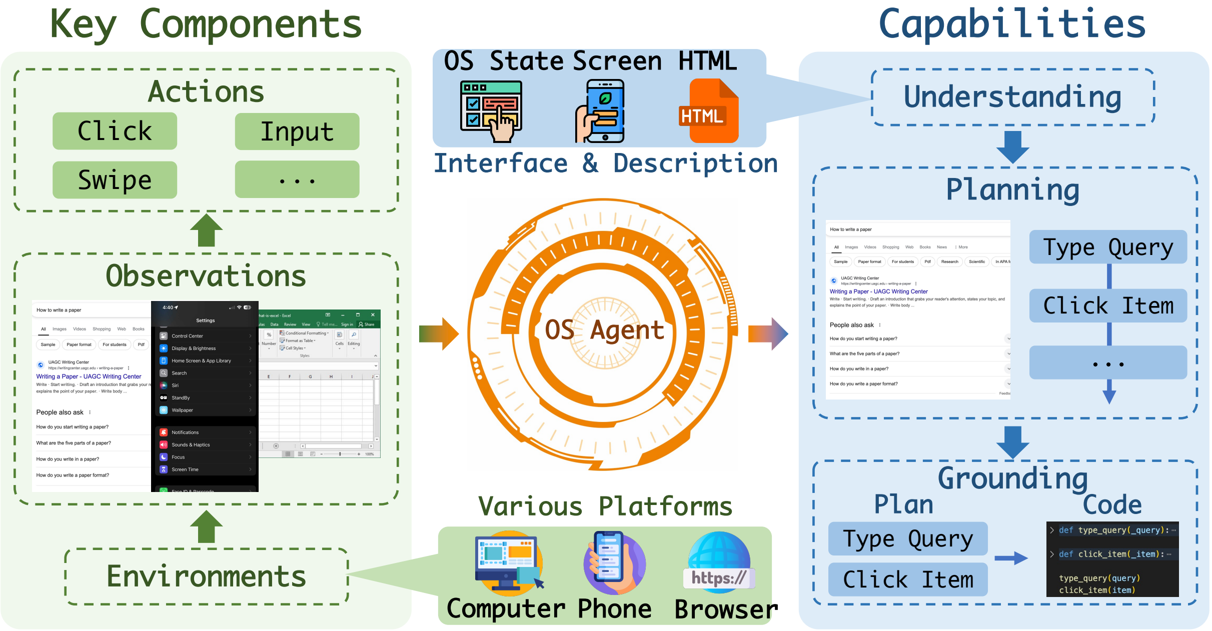
Grounding. Action grounding is another essential capability of OS Agents, referring to the ability to translate textual instructions or plans into executable actions within the operating environment (Zheng et al., 2024a; Wu et al., 2024f). The agent must identify elements on the screen and provide the necessary parameters (e.g., coordinates, input values) to ensure successful execution. While OS environments often contain numerous selectable elements and possible actions, the resulting complexity makes action grounding particularly challenging.

3 Construction of OS Agents

In this section, we discuss effective strategies for constructing OS Agents, including training domain-specific foundation models and designing agent frameworks for OS Agents.

3.1 Foundation Model

The construction of foundation models for OS Agents involves two key components: model architecture and training strategies, including pre-training, supervised finetuning and reinforcement learning. Table 1 in the Appendix summarizes the architecture and training strategies used in the



```
> def type_query(_query):…
> def click_item(_item):…
  type_query(query)
  click_item(item)
```

Figure 3: Fundamentals of OS Agents. Key components that OS Agents based: Environment, Obersevation Space and Action Space. Capabilities of OS Agents: Understanding, Planning and Grounding.

recent foundation models for OS Agents (as of January 2025).

3.1.1 Architecture

A variety of architectures are employed in foundation models for OS Agents. We discuss four common choices on model architectures when building OS Agents as follows.

Existing LLMs. The architecture of existing LLMs can already process user instructions and read HTML code to perceive information contained in user interfaces. Therefore, several works (Liu et al., 2024c; Lai et al., 2024; Patel et al., 2024; Liu et al., 2024a) directly leverage existing open-source LLMs as backbone models without further optimizing architecture and train based on it to develop foundation models for OS Agents.

Existing MLLMs. Although LLMs are capable of handling OS tasks, an inescapable shortcoming is that LLMs can only process textual input, while GUIs are designed for human users that directly perceive vision information (Xu et al., 2024e; Meng et al., 2024). Therefore, existing open-source MLLMs which additionally have the ability to process vision information while preserving the ability for complex natural language processing are introduced (Baechler et al., 2024; Chen et al., 2024d; Pawlowski et al., 2024).

Concatenated MLLMs. Typical architecture of MLLMs consists of an LLM and a vision encoder connected by an adapter network or a cross-attention module. Several works (Kil et al., 2024; Zhang et al., 2023d) have shown that choosing LLMs and vision encoders that are suitable to process OS tasks and concatenating them could be a more suitable approach for constructing foundation models for OS Agents.

Modified MLLMs. Further adjustments have been adopted to architectures of MLLMs to enhance understanding abilities of OS Agents. For instance, most existing MLLMs can only process images of relatively low resolutions, typically 224 × 224, while common resolution of GUI screenshots is much larger. Some works have been proposed to modify MLLMs with specific modules to perceive these features. For example, CogAgent (Hong et al., 2024) introduced additional EVA-CLIP-L high-resolution vision encoder that accepts images of size 1120 × 1120, and added a cross-attention module to connect with the original MLLM.

3.1.2 Pre-training

Continual pre-training are used to enhance the foundation models for OS Agents by expanding their understanding of GUI and facilitating the acquisition of the inherent correlations between visual and textual information. We discuss two important factors: data source and task in pre-training.

Data source. (1) *Publicly available data.* Some studies leverage publicly available datasets to quickly obtain large-scale data for pre-training (Gur et al., 2023; Nong et al., 2024). However, relying solely on publicly available data for pre-training is insufficient to address the complex and diverse tasks required by OS Agents (Gou et al., 2024). Consequently, (2) *Synthetic data.* Researchers incorporate synthetic data into the pre-training process (Cheng et al., 2024a; Chen et al., 2024c).

Task. (1) *Screen grounding.* Many studies have the task of extracting 2D coordinates or bounding boxes of target elements from images based on textual descriptions in pre-training (Wu et al., 2024f; Baechler et al., 2024; Pawlowski et al., 2024). (2) *Screen understanding.* Several studies posit that the foundation models for OS Agents should be capable of extracting semantic information from images, as well as analyzing and interpreting the entire content of the image. (3) *Optical Character Recognition (OCR).* OCR plays a crucial role in handling GUI elements that contain textual content. For instance, Hong et al. (2024) constructed training data during the pre-training stage by using Paddle-OCR to extract text and bounding boxes from GUI screenshots.

3.1.3 Supervised Finetuning

Supervised Fine-Tuning (SFT) has been widely adopted for further enhancing the GUI referring and grounding abilities of the model and making the model fit for navigation tasks.

In order to collect high-quality SFT data, several aspects of work have been proposed. *(1) Rule-Based Data Synthesis.* Several works use tools and specific rules to explore existing web data and extend trajectory data collections. For example, Wu et al. (2024d) adopted breadth-first search (BFS) to cover the app functions and generate action sequences based on the exploration. *(2) Model-Based Data Synthesis.* Several works use (M)LLMs to generate data samples based on webpages or mobile screens from existing datasets. For instance, Zhang et al. (2024f) prompted GPT-4V to generate data samples for GUI referring & grounding and screen summarization tasks. *(3) Model-Based Data Augmentation.* Zhang et al. (2024e) demonstrated that models trained with Chain-of-Action-Thought (CoAT) data, which includes screen description, thinking process about next action, the next action and possible action outcomes, would have better performance on GUI navigation tasks. Therefore, (M)LLMs are employed in several works to construct CoAT data based on existing trajectory data (Xu et al., 2024e; Lai et al., 2024).

3.1.4 Reinforcement Learning

More recently, research has progressed to the "LLMs as agents" paradigm, where LLMs serve as policy models and reinforcement learning is applied to align (M)LLMs with the final objectives. Thil et al. (2024) improved web navigation in LLMs using the Miniwob++ benchmark by fine-tuning T5-based models with hierarchical planning and then integrating these with a multimodal neural network, utilizing both supervised and reinforcement learning. Fereidouni et al. (2024) employs the Flan-T5 architecture and introduce training via Reinforcement Learning. They leveraged human demonstrations through behavior cloning and then further trained the agent with PPO. Reinforcement learning is also introduced to the agents based on vision-only models (Shaw et al., 2023) and MLLMs (Bai et al., 2024; Wang et al., 2024g; Fan et al., 2025a). For instance, Fan et al. (2025a) introduced Q-ICRL, a novel Q-value-incentive in-context reinforcement learning method to optimize exploration efficiency and data quality to improve GUI action grounding.

3.2 Agent Framework

OS Agent frameworks typically consist of four core components: Perception, Planning, Memory, and Action. Table 2 in the Appendix summarizes the technical characteristics of recent OS Agent frameworks.

3.2.1 Perception

Perception is the process through which OS Agents collect and analyze information from the environment. In OS Agents, the perception component needs to observe the current environment and extract relevant information to assist with the agents' planning, action, and memory optimization. Perception can be broadly categorized into two types based on the input modality as follows:

Textual Description of OS. Early works (Ma et al., 2023; Wang et al., 2023a; Lee et al., 2023; Gao et al., 2023) are limited by the fact that LLMs could only process textual input. Therefore, they mainly rely on using tools to convert OS environments into text descriptions, often represented in a structured format, such as HTML, DOM, or accessibility tree.

GUI Screenshot. The emergence of MLLMs enables OS Agents to process visual inputs. Research (Tan et al.; Ma et al., 2024c; Hu et al., 2024b) is increasingly treating GUI screenshots as the perception input for OS Agents, which better aligns with human behavior. To enhance OS Agents' understanding and grounding ability without fine-tuning visual encoders, existing research focuses on using prompting techniques to describe GUI screenshots. These descriptions can generally be categorized into three types: (1) *Visual description.* Most research (Yan et al., 2023; Wang et al., 2024a; Rawles et al., 2024a) uses SoM prompting (Yang et al., 2023) to enhance OS Agents' visual grounding ability. (2) *Semantic description.* Some studies (Pan et al.; Zheng et al., 2024a,b) improve OS Agents' understanding and grounding ability by adding descriptions of these interactive elements. (3) *Dual description.* Dual description combines both visual and semantic information to improve OS Agents' understanding and grounding of the visual environment (Zhang et al., 2023a; Wang and Liu, 2024).

3.2.2 Planning

Planning is the process of developing a sequence of actions to achieve a specific goal based on the current environment (Huang and Chang, 2023; Zhang et al., 2024i; Huang et al., 2024b). It enables OS Agents to break down complex tasks into smaller, manageable sub-tasks and solve them step by step. We categorize existing studies into two key approaches based on whether the planning is fixed or iterates in response to environmental changes: global planning and iterative planning.

Global. OS Agents only generate a global plan once and execute it without making adjustments based on environmental changes. Chain-of-Thought (CoT) (Wei et al., 2023) prompts (M)LLMs to break down complex tasks into reasoning steps, which forms the foundation of global planning in most OS Agents (Fu et al., 2024; Vu et al., 2024). Due to the one-time nature of global planning, research on global planning focuses on fitting the OS Agents' environment and tasks, proposing sufficiently feasible plans from the outset (Wu et al., 2024e; Gao et al., 2023; Agashe et al., 2024).

Iterative. In contrast to global planning, iterative planning allows OS Agents to continuously iterate their plans based on historical actions or changes in the environment, enabling them to adapt to ongoing environmental changes. This methodology is crucial for OS Agents to handle dynamic and unpredictable environments effectively. In specific, ReAct (Yao et al., 2023) builds on the concept of CoT by integrating reasoning with the outcome of actions, making planning more adaptable to changes in the environment. This approach has been widely applied in OS Agents (Zhang et al., 2023a; Ma et al., 2023; He et al., 2024a) for iterative planning. In addition, some studies have proposed iterative planning approaches specifically tailored for OS Agents. For instance, Auto-GUI (Zhang and Zhang, 2023) employs a CoT technique, where a history of past actions is used to generate future plans iteratively after each step.

3.2.3 Memory

Memory module saving useful information serves as one of the core components for OS Agents to perform tasks, adapt to dynamic environments, and continuously optimize their performance during task execution in various complex scenarios.

Memory Sources. Memory can be categorized into Internal Memory and External Memory, serving distinct functions in task execution: immediate information storage and external knowledge support. (1) *Internal Memory.* Internal memory contains information during task completion, such as action history (Zhang and Zhang, 2023), previous screenshots (Zhang et al., 2024e; Rawles et al., 2024b; Wang and Liu, 2024) and state data (Abuelsaad et al., 2024; Tao et al., 2023). (2) *External Memory.* External memory provides long-term knowledge support, primarily enriching an agent's memory capabilities through knowledge bases, external documents, and online information. For instance, some agents dynamically acquire external knowledge by invoking tools (Song et al., 2024a; Reddy et al., 2024), integrating this knowledge into their memory to assist with task execution and decision optimization.

Memory Optimization. Memory optimization can enhance an agent's efficiency in operations and decision-making during complex tasks by effectively managing and utilizing memory resources. In the following, we introduce several key strategies. (1) *Management.* For humans, memory information is constantly processed and abstracted in the brain. Similarly, the memory of OS Agents can be effectively managed to generate higher-level information, consolidate redundant content, and remove irrelevant or outdated information (Tang and Shin, 2024; Wen et al., 2024a). (2) *Growth Experience.*

By revisiting each step of a task, the agent can analyze successes and failures, identify opportunities for improvement, and avoid repeating mistakes in similar scenarios (Kim et al., 2024a; Niu et al., 2024). OS Agents can return to a previous state and choose an alternative path when the current task path proves infeasible or the results do not meet expectations, which is akin to classic search algorithms, enabling the agent to explore multiple potential solutions and find the optimal path (Ma et al., 2023; Li et al., 2024b; Zhu et al., 2024; Li et al., 2024e). (3) *Experience Retrieval.* OS Agents can efficiently plan and execute by retrieving experiences similar to the current task from long-term memory, which helps to reduce redundant operations (Zheng et al., 2023; Deng et al., 2024b; Cho et al., 2024).

3.2.4 Action

The action space defines the interfaces through which (M)LLM-based Agents engage with operating systems, spanning across platforms such as computers, mobile phones, and web browsers. We systematically categorized the action space of OS Agents into input operations, navigation operations, and extended operations.

Input Operations. Input operations encompass interactions via mouse/touch and keyboard (Sun et al., 2022; Fu et al., 2024; Deng et al., 2024a), forming the foundation for OS Agents to interact with digital interfaces.

Navigation Operations. Navigation operations enable OS Agents to traverse targeted platforms and acquire sufficient information for subsequent actions. Navigation operations encompass both basic navigation (Lee et al., 2023; Wang et al., 2024a) and web-specific navigation features (He et al., 2024a).

Extended Operations. Extended Operations provide additional capabilities beyond standard interface interactions, enabling more flexible and powerful agent behaviors. These operations primarily include (1) *code execution* capabilities that allow agents to dynamically extend their action space beyond predefined operations, enabling flexible and customizable control through direct script execution and command interpretation (Wu et al., 2024e; Mei et al., 2024; Tan et al.), and (2) *API integration* features that expand agents' capabilities by accessing external tools and information resources, facilitating interactions with third-party services and specialized functionalities (Wu et al., 2024e; Mei et al., 2024; Tan et al.; Li et al., 2024b).

4 Evaluation of OS Agents

We provide a comprehensive overview of a generic evaluation framework for OS Agents, structured around evaluation metrics and benchmarks. We have listed the recent benchmarks for OS Agents in Table 3 in the Appendix.

4.1 Evaluation Metric

During evaluation, OS Agents provided with task instructions and the current environment input, is expected to execute a sequence of continuous actions until the task is accomplished. By collecting the agent's observations, action outputs, and other environmental information during the process, specific metrics can be calculated. Specifically, the evaluation scope includes both granular step-level evaluations and a more holistic task-level assessment.

Step-level Evaluation. Step-level evaluation centers on a detailed, step-by-step analysis of the planning trajectory, offering a fine-grained evaluation of the actions taken by the agent at each step. In step-level evaluation, the agent's output in response to instruction of each step is directly assessed, with a focus on the accuracy of action grounding and the matching of potential object elements (which refers to the target of the action) (Xu et al., 2024c; Jin et al., 2024; Pasupat et al., 2018).

Task-level Evaluation. Task-level evaluation centers on the final output and evaluates whether the agent reaches the desired final state. The two main criteria are task completion and resource utilization. (1) *Task Completion Metrics.* Task Completion Metrics measure the effectiveness of OS Agents in successfully accomplishing assigned tasks. For instance, *Overall Success Rate (SR)* (Koh et al., 2024a; Zhang and Zhang, 2023; Drouin et al., 2024) provides a straightforward measure of the proportion of tasks that are fully completed. (2) *Efficiency Metrics.* Efficiency Metrics evaluate how efficiently the agent completes assigned tasks, considering factors such as step cost, hardware expenses, and time expenditure. For instance, *Step Ratio* (Chen et al., 2024a; Lee et al., 2024b; Wang et al., 2024c) compares the number of steps taken by the agent to the optimal one (often defined by human performance). A lower step ratio indicates a more efficient and optimized task execution.

4.2 Evaluation Benchmark

To comprehensively evaluate the performance and capabilities of OS Agents, researchers have developed a variety of benchmarks. These benchmarks construct various environments, based on different platforms and settings, and cover a wide range of tasks.

4.2.1 Evaluation Platform

The platform acts as an integrated evaluation environment, specifically encompassing the virtual settings in which benchmarks are performed. Some benchmarks also incorporate multiple platforms at the same time, which places greater demands on the agent's cross-platform transferability. Existing real-world platforms can primarily be categorized into three types: Computer, Phone, and Browser.

Computer. Computer platform is complex due to the diversity of operating systems and applications. Efficient computer benchmarks (Xie et al., 2024b; Wang et al., 2024j; Bonatti et al., 2024) need to handle the wide variety and complexity of real-world computing environments, which span different operating systems, interfaces, and applications.

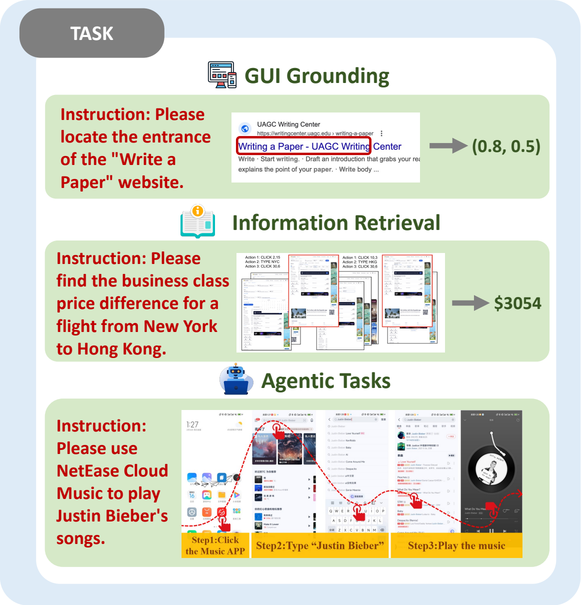
Phone. Phone platforms such as Android (Li et al., 2024c; Lee et al., 2024b; Bishop et al., 2024) or iOS (Yan et al., 2023) present unique challenges for OS Agents. While phone GUI elements are simpler due to smaller screens, they require more complex actions, such as precise gestures for navigating widgets or zooming, which imposes higher demands on the agents' planning and action grounding capabilities.

Browser. Browser platforms are essential interfaces to access online resources. Webpages (Koh et al., 2024a; Lù et al., 2024; Drouin et al., 2024; Yao et al., 2022; Shi et al., 2017) are open and built with HTML, CSS, and JavaScript, making them easy to inspect and modify in real-time.

4.2.2 Task

To comprehensively assess the capabilities of OS Agents, a spectrum of specialized tasks has been integrated into the established benchmarks and introduced as follows. Figure 4 illustrates some cases of different task types.

GUI Grounding. GUI grounding tasks (Li et al., 2020; Fan et al., 2025b) aim to evaluate agent's abilities to transform instructions to various actionable elements. Grounding is fundamental for interacting with operation systems that OS Agents must possess.


TASK
GUI Grounding
Writing a Paper - UAGC Writing Center
(0.8, 0.5)
Information Retrieval
$3054
Agentic Tasks
Step1:Click the Music APP
Step2:Type "Justin Bieber"
Step3:Play the music

Instruction: Please locate the entrance of the "Write a Paper" website.

Instruction: Please find the business class price difference for a flight from New York to Hong Kong.

Instruction: Please use NetEase Cloud Music to play Justin Bieber's songs.

Figure 4: Three types of tasks: Information Retrieval, and Agentic Tasks in OS Agent benchmarks, with some images sourced from (Wang et al., 2024d).

Information Retrieval. Information Retrieval tasks (Pan et al., 2024; Zhang et al., 2024l; Drouin et al., 2024) examine agent's ability to process complex and dynamic information by understanding instructions and GUI interfaces, extracting the desired information or data.

Agentic Tasks. Agentic tasks (Lù et al., 2024; Zhang et al., 2024l) are a key focus in current research. In these tasks, OS Agents are provided with an instruction or goal and tasked with identifying the required steps, planning actions, and executing them until the target state is reached, without relying on any explicit navigation guidance.

5 Product of OS Agents

The rapid advancement and increasing interest in OS Agents research have significantly accelerated the development of commercial products in this domain. The interplay between research and product development is crucial, as cutting-edge academic breakthroughs often serve as the foundation for innovative commercial applications, while real-world product feedback further refines and drives research directions. This symbiotic relationship not only bridges the gap between theoretical exploration and practical implementation but also ensures that OS Agents evolve to meet both technological and user-centric demands.

OS Agent products are evolving towards platform diversification (e.g., computers (Anthropic, 2024a), phones (Apple, 2024), browsers (DeepMind, 2024)) and functional stratification into task execution- and search-oriented types. From 2023 to 2024, they progressed from technological validation and demonstration to deeper OS integration, enhanced capabilities, and actual productivity. Due to space limitations, further details and a list of recent commercial products are in Table 4 in Appendix D.

6 Challenge & Future

6.1 Safety & Privacy

Many studies (Deng et al., 2024c; Gan et al., 2024a; Yao et al., 2024) investigate the security and privacy risks associated with (M)LLMs-based Agents. OS Agents are also confronted with these risks, especially considering its wide applications on personal devices with user data.

Researchers have highlighted significant security vulnerabilities in OS Agents. Attack methodologies include Web Indirect Prompt Injection (WIPI) using embedded web instructions (Wu et al., 2024b), adversarial images misleading agent perception (Wu et al., 2025a), and environmental injection attacks that embed malicious instructions in web pages to induce unintended actions or data theft (Ma et al., 2024b; Liao et al., 2024). Other identified threats encompass backdoor attacks, adversarial pop-ups, and the susceptibility of refusal-trained LLMs to jailbreaking in browser contexts (Yang et al., 2024b; Zhang et al., 2024j; Kumar et al., 2024).

While general security frameworks for LLM agents exist (Ruan et al., 2024; Hua et al., 2024), defenses specifically tailored to OS Agents are still nascent (Pedro et al., 2023). This underscores the need for robust defense mechanisms against threats like injection and backdoors.

To assess OS Agent robustness, several benchmarks have emerged. ST-WebAgentBench (Levy et al., 2024) evaluates web agent safety in enterprise settings. MobileSafetyBench (Lee et al., 2024a) assesses mobile agent security, particularly for safety-critical tasks. AgentDojo (Debenedetti et al., 2024) offers a dynamic environment for testing prompt injection attacks, and AgentHarm (Andriushchenko et al., 2024) measures the potential harm from agents executing malicious tasks.

Due to limited space, we place the detailed discussion in the Appendix E.1.1 analyzes various attack strategies targeting OS Agents, §E.1.2 explores existing defense mechanisms and limitations, and §E.1.3 reviews existing security benchmarks designed to assess the robustness and reliability of OS Agents.

6.2 Personalization & Self-Evolution

Much like Jarvis as Iron Man's personal assistant in the movies, developing personalized OS Agents has been a long-standing goal in AI research. A personal assistant is expected to self-evolve, which means to continuously adapt and provide enhanced experiences based on individual user preferences. OpenAI's memory feature[6] has made strides in this direction, but many OS Agents today still perform insufficient in providing personalized experience to users and self-evolving over user interactions.

Early LLM-based Agents in games demonstrated the effectiveness of text-based memory for self-evolution (Wang et al., 2023b; Zhu et al., 2023), which is later validated for OS Agents (Zhang et al., 2023a; Wu et al., 2024e). Wang et al. (2024h) introduces a general framework and synthesizes realistic benchmarks for lifelong personalization of LLM-based Agents. Some products, such as Mem0 (Chhikara et al., 2025), offer a memory layer as standalone solutions for LLM-based agents to enhance personalization and self-evolution.

However, significant challenges persist, particularly in expanding memory to multi-modal forms and ensuring its efficient management and retrieval. Overcoming these hurdles is key to developing truly context-aware and continually evolving OS Agents. We place the detailed discussion in the Appendix E.2.

7 Conclusion

In this survey, we outline the fundamentals underlying OS Agents, including their key components and capabilities. We have also reviewed various approaches to their construction and evaluation. Looking ahead, we identify challenges and future of OS Agents. We hope this survey contribute to the ongoing development of OS Agents and support their relevance and utility in both academic and industrial settings.

[6] https://openai.com/index/memory-and-new-controls-for-chatgpt/

Limitations

In this survey, we acknowledge that there are areas closely related to OS Agents that, due to space limitations, have not been discussed in depth. One such area is the technology of effective models deployed on edge devices like mobile phones, which is crucial for the practical deployment of OS Agents. Additionally, our focus has been on single agent setting, and several research works on multi-agent frameworks will be updated soon.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62402429, U24A20326, and 62441236), the Key Research and Development Program of Zhejiang Province (No. 2024C03270), the Ningbo Yongjiang Talent Introduction Programme (No. 2023A-397-G), and the Young Elite Scientists Sponsorship Program by CAST (No. 2024QNRC001). This work was also partially supported by the ZJU Kunpeng & Ascend Center of Excellence, MYbank, and Ant Group. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies or the government.

References

Tamer Abuelsaad, Deepak Akkil, Prasenjit Dey, Ashish Jagmohan, Aditya Vempaty, and Ravi Kokku. 2024. Agent-e: From autonomous web navigation to foundational design principles in agentic systems. *arXiv preprint arXiv:2407.13032*.

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. 2024. Agent s: An open agentic framework that uses computers like a human. *arXiv preprint arXiv:2410.08164*.

Amazon. 2024. Alexa - amazon. Accessed: 2024-12-04.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Eric Winsor, Jerome Wynne, Yarin Gal, and Xander Davies. 2024. Agentharm: A benchmark for measuring harmfulness of llm agents. *Preprint*, arXiv:2410.09024.

Anthropic. 2024a. 3.5 models and computer use - anthropic. Accessed: 2024-12-04.

Anthropic. 2024b. Claude model - anthropic. Accessed: 2024-12-04.

Apple. 2024. Apple intelligence. Accessed: 2024-12-04.

Ruhana Azam, Tamer Abuelsaad, Aditya Vempaty, and Ashish Jagmohan. 2024. Multimodal auto validation for self-refinement in web agents. *arXiv preprint arXiv:2410.00689*.

Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. 2024. Screenai: A vision-language model for ui and infographics understanding. *arXiv preprint arXiv:2402.04615*.

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. 2024. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *arXiv preprint arXiv:2406.11896*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *Preprint*, arXiv:2308.12966.

Husam Barham and Mohammed Fasha. 2024. Towards llmci-multimodal ai for llm-vision ui operation.

William E Bishop, Alice Li, Christopher Rawles, and Oriana Riva. 2024. Latent state estimation helps ui agents to reason. *arXiv preprint arXiv:2405.11120*.

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, et al. 2024. Windows agent arena: Evaluating multi-modal os agents at scale. *arXiv preprint arXiv:2409.08264*.

Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A Plummer. 2022. A dataset for interactive vision-language navigation with unknown command feasibility. In *European Conference on Computer Vision*, pages 312–328. Springer.

Andrea Burns, Kate Saenko, and Bryan A Plummer. 2024. Tell me what's next: Textual foresight for generic ui representations. *arXiv preprint arXiv:2406.07822*.

Hongru Cai, Yongqi Li, Wenjie Wang, Fengbin Zhu, Xiaoyu Shen, Wenjie Li, and Tat-Seng Chua. 2024. Large language models empowered personalized web agents. *arXiv preprint arXiv:2410.17236*.

Ruisheng Cao, Fangyu Lei, Haoyuan Wu, Jixuan Chen, Yeqiao Fu, Hongcheng Gao, Xinzhuang Xiong, Hanchong Zhang, Yuchen Mao, Wenjing Hu, et al. 2024. Spider2-v: How far are multimodal agents from automating data science and engineering workflows? *arXiv preprint arXiv:2407.10956*.

Yuxiang Chai, Siyuan Huang, Yazhe Niu, Han Xiao, Liang Liu, Dingyu Zhang, Peng Gao, Shuai Ren, and Hongsheng Li. 2024. Amex: Android multi-annotation expo dataset for mobile gui agents. *arXiv preprint arXiv:2407.17490*.

Yuxiang Chai, Hanhao Li, Jiayu Zhang, Liang Liu, Guozhi Wang, Shuai Ren, Siyuan Huang, and Hongsheng Li. 2025. A3: Android agent arena for mobile gui agents. *Preprint*, arXiv:2501.01149.

Jingxuan Chen, Derek Yuen, Bin Xie, Yuhao Yang, Gongwei Chen, Zhihao Wu, Li Yixing, Xurui Zhou, Weiwen Liu, Shuai Wang, et al. 2024a. Spa-bench: A comprehensive benchmark for smartphone agent evaluation. In *NeurIPS 2024 Workshop on Open-World Agents*.

Qi Chen, Dileepa Pitawela, Chongyang Zhao, Gengze Zhou, Hsiang-Ting Chen, and Qi Wu. 2024b. Webvln: Vision-and-language navigation on websites. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1165–1173.

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, et al. 2024c. Guicourse: From general vision language models to versatile gui agents. *arXiv preprint arXiv:2406.11317*.

Xuetian Chen, Hangcheng Li, Jiaqing Liang, Sihang Jiang, and Deqing Yang. 2024d. Edge: Enhanced grounded gui understanding with enriched multi-granularity synthetic data. *arXiv preprint arXiv:2410.19461*.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. 2024a. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*.

Yuheng Cheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang, Zekai Wang, Feng Yin, Junhua Zhao, et al. 2024b. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv preprint arXiv:2401.03428*.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

Junhee Cho, Jihoon Kim, Daseul Bae, Jinho Choo, Youngjune Gwon, and Yeong-Dae Kwon. 2024. Caap: Context-aware action planning prompting to solve computer tasks with front-end ui only. *arXiv preprint arXiv:2406.06947*.

Cognosys. 2024. Ottogrid. Accessed: 2025-02-01.

Tianyu Cui, Yanling Wang, Chuanpu Fu, Yong Xiao, Sijia Li, Xinhao Deng, Yunpeng Liu, Qinglin Zhang, Ziyi Qiu, Peiyang Li, Zhixing Tan, Junwu Xiong, Xinyu Kong, Zujie Wen, Ke Xu, and Qi Li. 2024. Risk taxonomy, mitigation, and assessment benchmarks of large language model systems. *Preprint*, arXiv:2401.05778.

Yong Dai, Duyu Tang, Liangxin Liu, Minghuan Tan, Cong Zhou, Jingquan Wang, Zhangyin Feng, Fan Zhang, Xueyu Hu, and Shuming Shi. 2022. One model, multiple modalities: A sparsely activated approach for text, sound, image, video and code. *Preprint*, arXiv:2205.06126.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. 2024. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Preprint*, arXiv:2406.13352.

Google DeepMind. 2024. Project mariner. Accessed: 2024-12-04.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024a. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Yang Deng, Xuan Zhang, Wenxuan Zhang, Yifei Yuan, See-Kiong Ng, and Tat-Seng Chua. 2024b. On the multi-turn instruction following for conversational web agents. *arXiv preprint arXiv:2402.15057*.

Zehang Deng, Yongjian Guo, Changzhou Han, Wanlun Ma, Junwu Xiong, Sheng Wen, and Yang Xiang. 2024c. Ai agents under threat: A survey of key security challenges and future pathways. *Preprint*, arXiv:2406.02630.

Tinghe Ding. 2024. Mobileagent: enhancing mobile control via human-machine interaction and sop integration. *arXiv preprint arXiv:2401.04124*.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, et al. 2024. Workarena: How capable are web agents at solving common knowledge work tasks? *arXiv preprint arXiv:2403.07718*.

Yue Fan, Handong Zhao, Ruiyi Zhang, Yu Shen, Xin Eric Wang, and Gang Wu. 2025a. Guibee: Align gui action grounding to novel environments via autonomous exploration. *arXiv preprint arXiv:2501.13896*.

Yue Fan, Handong Zhao, Ruiyi Zhang, Yu Shen, Xin Eric Wang, and Gang Wu. 2025b. Guibee: Align gui action grounding to novel environments via autonomous exploration. *Preprint*, arXiv:2501.13896.

Haishuo Fang, Xiaodan Zhu, and Iryna Gurevych. 2024. Inferact: Inferring safe actions for llm-based agents through preemptive evaluation and human feedback. *Preprint*, arXiv:2407.11843.

Moghis Fereidouni et al. 2024. Search beyond queries: Training smaller language models for web interactions via reinforcement learning. *arXiv preprint arXiv:2404.10887*.

Emilio Ferrara. 2023. Should chatgpt be biased? challenges and risks of bias in large language models. *First Monday*.

Kelin Fu, Yang Tian, and Kaigui Bian. 2024. Periguru: A peripheral robotic mobile app operation assistant based on gui image understanding and prompting with llm. *arXiv preprint arXiv:2409.09354*.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. 2023. Multimodal web navigation with instruction-finetuned foundation models. *arXiv preprint arXiv:2305.11854*.

Hiroki Furuta, Yutaka Matsuo, Aleksandra Faust, and Izzeddin Gur. 2024. Exposing limitations of language model agents in sequential-task compositions on the web. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Yuyou Gan, Yong Yang, Zhe Ma, Ping He, Rui Zeng, Yiming Wang, Qingming Li, Chunyi Zhou, Songze Li, Ting Wang, Yunjun Gao, Yingcai Wu, and Shouling Ji. 2024a. Navigating the risks: A survey of security, privacy, and ethics threats in llm-based agents. *Preprint*, arXiv:2411.09523.

Yuyou Gan, Yong Yang, Zhe Ma, Ping He, Rui Zeng, Yiming Wang, Qingming Li, Chunyi Zhou, Songze Li, Ting Wang, et al. 2024b. Navigating the risks: A survey of security, privacy, and ethics threats in llm-based agents. *arXiv preprint arXiv:2411.09523*.

Difei Gao, Lei Ji, Zechen Bai, Mingyu Ouyang, Peiran Li, Dongxing Mao, Qinchen Wu, Weichen Zhang, Peiyi Wang, Xiangwu Guo, et al. 2023. Assistgui: Task-oriented desktop graphical user interface automation. *arXiv preprint arXiv:2312.13108*.

Minghe Gao, Wendong Bu, Bingchen Miao, Yang Wu, Yunfei Li, Juncheng Li, Siliang Tang, Qi Wu, Yueting Zhuang, and Meng Wang. 2024. Generalist virtual agents: A survey on autonomous agents across digital platforms. *Preprint*, arXiv:2411.10943.

Zhiqi Ge, Juncheng Li, Xinglei Pang, Minghe Gao, Kaihang Pan, Wang Lin, Hao Fei, Wenqiao Zhang, Siliang Tang, and Yueting Zhuang. 2024. Iris: Breaking gui complexity with adaptive focus and self-refining. *arXiv preprint arXiv:2412.10342*.

Google. 2024. Google assistant. Accessed: 2024-12-04.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2024. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243*.

Yu Gu, Boyuan Zheng, Boyu Gou, Kai Zhang, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. 2024. Is your llm secretly a world model of the internet? model-based planning for web agents. *arXiv preprint arXiv:2411.06559*.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2023. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024a. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Hongming Zhang, Tianqing Fang, Zhenzhong Lan, and Dong Yu. 2024b. Openwebvoyager: Building multimodal web agents via iterative real-world exploration, feedback and optimization. *arXiv preprint arXiv:2410.19609*.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290.

Jakub Hoscilowicz, Bartosz Maj, Bartosz Kozakiewicz, Oleksii Tymoshchuk, and Artur Janicki. 2024. Clickagent: Enhancing ui location capabilities of autonomous agents. *arXiv preprint arXiv:2410.11872*.

Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. 2023. Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*.

Sihao Hu, Tiansheng Huang, Fatih Ilhan, Selim Tekin, Gaowen Liu, Ramana Kompella, and Ling Liu. 2024a. A survey on large language model-based game agents. *arXiv preprint arXiv:2404.02039*.

Siyuan Hu, Mingyu Ouyang, Difei Gao, and Mike Zheng Shou. 2024b. The dawn of gui agent: A preliminary case study with claude 3.5 computer use. *Preprint*, arXiv:2411.10323.

Wenyue Hua, Xianjun Yang, Mingyu Jin, Zelong Li, Wei Cheng, Ruixiang Tang, and Yongfeng Zhang. 2024. Trustagent: Towards safe and trustworthy llm-based agents. *Preprint*, arXiv:2402.01586.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. *Preprint*, arXiv:2212.10403.

Tenghao Huang, Kinjal Basu, Ibrahim Abdelaziz, Pavan Kapanipathi, Jonathan May, and Muhao Chen.

2025. R2d2: Remembering, reflecting and dynamic decision making for web agents. *Preprint*, arXiv:2501.12485.

Tian Huang, Chun Yu, Weinan Shi, Zijian Peng, David Yang, Weiqi Sun, and Yuanchun Shi. 2024a. Promptrpa: Generating robotic process automation on smartphones from textual prompts. *arXiv preprint arXiv:2404.02475*.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024b. Understanding the planning of llm agents: A survey. *arXiv preprint arXiv:2402.02716*.

Faria Huq, Zora Zhiruo Wang, Frank F. Xu, Tianyue Ou, Shuyan Zhou, Jeffrey P. Bigham, and Graham Neubig. 2025. Cowpilot: A framework for autonomous and human-agent collaborative web navigation. *Preprint*, arXiv:2501.16609.

iMean.AI. 2024. imean. Accessed: 2025-02-01.

Apple Inc. 2024. Siri - apple. Accessed: 2024-12-04.

Iat Long Iong, Xiao Liu, Yuxuan Chen, Hanyu Lai, Shuntian Yao, Pengbo Shen, Hao Yu, Yuxiao Dong, and Jie Tang. 2024. Openwebagent: An open toolkit to enable web agents on large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 72–81.

Yue Jiang, Eldon Schoop, Amanda Swearngin, and Jeffrey Nichols. 2023. Iluvui: Instruction-tuned language-vision modeling of uis from machine conversations. *arXiv preprint arXiv:2310.04869*.

Yilun Jin, Zheng Li, Chenwei Zhang, Tianyu Cao, Yifan Gao, Pratik Jayarao, Mao Li, Xin Liu, Ritesh Sarkhel, Xianfeng Tang, et al. 2024. Shopping mmlu: A massive multi-task online shopping benchmark for large language models. *arXiv preprint arXiv:2410.20745*.

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem Alshikh, and Ruslan Salakhutdinov. 2024. Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. *arXiv preprint arXiv:2402.17553*.

Jihyung Kil, Chan Hee Song, Boyuan Zheng, Xiang Deng, Yu Su, and Wei-Lun Chao. 2024. Dual-view visual contextualization for web navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14445–14454.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2024a. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36.

Jaekyeom Kim, Dong-Ki Kim, Lajanugen Logeswaran, Sungryull Sohn, and Honglak Lee. 2024b. Autointent: Automated intent discovery and self-exploration for large language model web agents. *arXiv preprint arXiv:2410.22552*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024a. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*.

Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. 2024b. Tree search for language model agents. *arXiv preprint arXiv:2407.01476*.

Priyanshu Kumar, Elaine Lau, Saranya Vijayakumar, Tu Trinh, Scale Red Team, Elaine Chang, Vaughn Robinson, Sean Hendryx, Shuyan Zhou, Matt Fredrikson, Summer Yue, and Zifan Wang. 2024. Refusal-trained llms are easily jailbroken as browser agents. *Preprint*, arXiv:2410.13886.

Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, et al. 2024. Autowebglm: A large language model-based web navigating agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5295–5306.

langchain-ai. 2025. Langmem: Prebuilt utilities for memory management and retrieval. `https://github.com/langchain-ai/langmem`. Version 0.0.27.

Juyong Lee, Dongyoon Hahm, June Suk Choi, W. Bradley Knox, and Kimin Lee. 2024a. Mobilesafetybench: Evaluating safety of autonomous agents in mobile device control. *Preprint*, arXiv:2410.17520.

Juyong Lee, Taywon Min, Minyong An, Dongyoon Hahm, Haeone Lee, Changyeon Kim, and Kimin Lee. 2024b. Benchmarking mobile device control agents across diverse configurations. *arXiv preprint arXiv:2404.16660*.

Sunjae Lee, Junyoung Choi, Jungjae Lee, Munim Hasan Wasi, Hojun Choi, Steven Y Ko, Sangeun Oh, and Insik Shin. 2023. Explore, select, derive, and recall: Augmenting llm with human-like memory for mobile task automation. *arXiv preprint arXiv:2312.03003*.

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. 2024. St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents. *Preprint*, arXiv:2410.06703.

Dongxu Li, Yudong Liu, Haoning Wu, Yue Wang, Zhiqi Shen, Bowen Qu, Xinyao Niu, Guoyin Wang, Bei Chen, and Junnan Li. 2024a. Aria: An open multimodal native mixture-of-experts model. *Preprint*, arXiv:2410.05993.

Eric Li and Jim Waldo. 2024. Websuite: Systematically evaluating why web agents fail. *arXiv preprint arXiv:2406.01623*.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and ZHAO-XIANG ZHANG. 2024b. Sheetcopilot: Bringing software productivity to the next level through large language models. *Advances in Neural Information Processing Systems*, 36.

Tao Li, Gang Li, Zhiwei Deng, Bryan Wang, and Yang Li. 2023. A zero-shot language agent for computer control with structured reflection. *arXiv preprint arXiv:2310.08740*.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2024c. On the effects of data scale on computer control agents. *arXiv preprint arXiv:2406.03679*.

Wei Li, William E Bishop, Alice Li, Christopher Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on ui control agents. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024d. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9.

Yanda Li, Chi Zhang, Wanqi Yang, Bin Fu, Pei Cheng, Xin Chen, Ling Chen, and Yunchao Wei. 2024e. Appagent v2: Advanced agent for flexible mobile interactions. *arXiv preprint arXiv:2408.11824*.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020. Mapping natural language instructions to mobile ui action sequences. *arXiv preprint arXiv:2005.03776*.

Yingji Li, Mengnan Du, Rui Song, Xin Wang, and Ying Wang. 2024f. A survey on fairness in large language models. *Preprint*, arXiv:2308.10149.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, Rui Kong, Yile Wang, Hanfei Geng, Jian Luan, Xuefeng Jin, Zilong Ye, Guanjing Xiong, Fan Zhang, Xiang Li, Mengwei Xu, Zhijun Li, Peng Li, Yang Liu, Ya-Qin Zhang, and Yunxin Liu. 2024g. Personal llm agents: Insights and survey about the capability, efficiency and security. *Preprint*, arXiv:2401.05459.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. 2024h. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*.

Zhangsheng Li, Keen You, Haotian Zhang, Di Feng, Harsh Agrawal, Xiujun Li, Mohana Prasad Sathya Moorthy, Jeff Nichols, Yinfei Yang, and Zhe Gan. 2024i. Ferret-ui 2: Mastering universal user interface understanding across platforms. *arXiv preprint arXiv:2410.18967*.

Zeyi Liao, Lingbo Mo, Chejian Xu, Mintong Kang, Jiawei Zhang, Chaowei Xiao, Yuan Tian, Bo Li, and Huan Sun. 2024. Eia: Environmental injection attack on generalist web agents for privacy leakage. *Preprint*, arXiv:2409.11295.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. 2024. Showui: One vision-language-action model for generalist gui agent. In *NeurIPS 2024 Workshop on Open-World Agents*.

Guangyi Liu, Pengxiang Zhao, Liang Liu, Yaxuan Guo, Han Xiao, Weifeng Lin, Yuxiang Chai, Yue Han, Shuai Ren, Hao Wang, Xiaoyu Liang, Wenhao Wang, Tianze Wu, Linghao Li, Hao Wang, Guanjing Xiong, Yong Liu, and Hongsheng Li. 2025a. Llm-powered gui agents in phone automation: Surveying progress and prospects. *Preprint*, arXiv:2504.19838.

Jiarun Liu, Jia Hao, Chunhong Zhang, and Zheng Hu. 2024a. Wepo: Web element preference optimization for llm-based web navigation. *arXiv preprint arXiv:2412.10742*.

Junpeng Liu, Tianyue Ou, Yifan Song, Yuxiao Qu, Wai Lam, Chenyan Xiong, Wenhu Chen, Graham Neubig, and Xiang Yue. 2024b. Harnessing webpage uis for text-rich visual understanding. *arXiv preprint arXiv:2410.13824*.

Xiao Liu, Bo Qin, Dongzhu Liang, Guang Dong, Hanyu Lai, Hanchen Zhang, Hanlin Zhao, Iat Long Iong, Jiadai Sun, Jiaqi Wang, et al. 2024c. Autoglm: Autonomous foundation agents for guis. *arXiv preprint arXiv:2411.00820*.

Yuhang Liu, Pengxiang Li, Zishu Wei, Congkai Xie, Xueyu Hu, Xinchen Xu, Xiaotian Han, Hongxia Yang, and Fei Wu. 2025b. Infiguiagent: A multimodal generalist gui agent with native reasoning and reflection. *arXiv preprint arXiv:2501.04575*.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*.

Quanfeng Lu, Wenqi Shao, Zitao Liu, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. *arXiv preprint arXiv:2406.08451*.

Xing Han Lù, Zdeněk Kasner, and Siva Reddy. 2024. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. 2024. Omniparser for pure vision based gui agent. *arXiv preprint arXiv:2408.00203*.

Gen Luo, Xue Yang, Wenhan Dou, Zhaokai Wang, Jifeng Dai, Yu Qiao, and Xizhou Zhu. 2024. Mono-internvl: Pushing the boundaries of monolithic multimodal large language models with endogenous visual pre-training. *arXiv preprint arXiv:2410.08202*.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024a. Agentboard: An analytical evaluation board of multi-turn llm agents. *arXiv preprint arXiv:2401.13178*.

Kaixin Ma, Hongming Zhang, Hongwei Wang, Xiaoman Pan, Wenhao Yu, and Dong Yu. 2023. Laser: Llm agent with state-space exploration for web navigation. *arXiv preprint arXiv:2309.08172*.

Xinbei Ma, Yiting Wang, Yao Yao, Tongxin Yuan, Aston Zhang, Zhuosheng Zhang, and Hai Zhao. 2024b. Caution for the environment: Multimodal agents are susceptible to environmental distractions. *Preprint*, arXiv:2408.02544.

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2024c. Coco-agent: A comprehensive cognitive mllm agent for smartphone gui automation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9097–9110.

Kai Mei, Zelong Li, Shuyuan Xu, Ruosong Ye, Yingqiang Ge, and Yongfeng Zhang. 2024. Aios: Llm agent operating system. *arXiv e-prints, pp. arXiv–2403*.

Ziyang Meng, Yu Dai, Zezheng Gong, Shaoxiong Guo, Minglong Tang, and Tongquan Wei. 2024. Vga: Vision gui assistant–minimizing hallucinations through image-centric fine-tuning. *arXiv preprint arXiv:2406.14056*.

Shikhar Murty, Dzmitry Bahdanau, and Christopher D Manning. 2024. Nnetscape navigator: Complex demonstrations for web agents without a demonstrator. *arXiv preprint arXiv:2410.02907*.

N4NO. 2024. Visiopilot. Accessed: 2025-02-01.

Seth Neel and Peter Chang. 2024. Privacy issues in large language models: A survey. *Preprint*, arXiv:2312.06717.

Runliang Niu, Jindong Li, Shiqi Wang, Yali Fu, Xiyu Hu, Xueyuan Leng, He Kong, Yi Chang, and Qi Wang. 2024. Screenagent: A vision language model-driven computer control agent. *arXiv preprint arXiv:2402.07945*.

Songqin Nong, Jiali Zhu, Rui Wu, Jiongchao Jin, Shuo Shan, Xiutian Huang, and Wenhao Xu. 2024. Mobileflow: A multimodal llm for mobile gui agent. *arXiv preprint arXiv:2407.04346*.

OpenAI. Home - openai. Accessed: 2024-12-12.

OpenAI. 2025. Introducing deep research. `https://openai.com/index/introducing-deep-research/`. Accessed: 2025-05-31.

OpenAI. 2025. Operator. Accessed: 2025-02-01.

OthersideAI. 2023. Self-operating computer. Accessed: 2025-02-01.

Tianyue Ou, Frank F Xu, Aman Madaan, Jiarui Liu, Robert Lo, Abishek Sridhar, Sudipta Sengupta, Dan Roth, Graham Neubig, and Shuyan Zhou. 2024. Synatra: Turning indirect knowledge into direct demonstrations for digital agents at scale. *arXiv preprint arXiv:2409.15637*.

Jiayi Pan, Yichi Zhang2 Nicholas Tomlin1 Yifei Zhou, Sergey Levine, and Alane Suhr. Autonomous evaluation and refinement of web agents.

Yichen Pan, Dehan Kong, Sida Zhou, Cheng Cui, Yifei Leng, Bing Jiang, Hangyu Liu, Yanyi Shang, Shuyan Zhou, Tongshuang Wu, et al. 2024. Webcanvas: Benchmarking web agents in online environments. *arXiv preprint arXiv:2406.12373*.

Danny Park. 2024. Human player outwits freysa ai agent in $47,000 crypto challenge. Accessed: 2024-11-30.

Panupong Pasupat, Tian-Shun Jiang, Evan Zheran Liu, Kelvin Guu, and Percy Liang. 2018. Mapping natural language commands to web elements. *arXiv preprint arXiv:1808.09132*.

Ajay Patel, Markus Hofmarcher, Claudiu Leoveanu-Condrei, Marius-Constantin Dinu, Chris Callison-Burch, and Sepp Hochreiter. 2024. Large language models can self-improve at web agent tasks. *arXiv preprint arXiv:2405.20309*.

Pawel Pawlowski, Krystian Zawistowski, Wojciech Lapacz, Marcin Skorupa, Adam Wiacek, Sebastien Postansque, and Jakub Hoscilowicz. 2024. Tinyclick: Single-turn agent for empowering gui automation. *arXiv preprint arXiv:2410.11871*.

Rodrigo Pedro, Daniel Castro, Paulo Carreira, and Nuno Santos. 2023. From prompt injections to sql injection attacks: How protected is your llm-integrated web application? *Preprint*, arXiv:2308.01990.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Reasoning with language model prompting: A survey. *arXiv preprint arXiv:2212.09597*.

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. 2025. Ui-tars: Pioneering automated gui interaction with native agents. *Preprint*, arXiv:2501.12326.

Dezhi Ran, Mengzhou Wu, Hao Yu, Yuetong Li, Jun Ren, Yuan Cao, Xia Zeng, Haochuan Lu, Zexin Xu, Mengqian Xu, Ting Su, Liangchao Yao, Ting Xiong, Wei Yang, Yuetang Deng, Assaf Marron, David Harel, and Tao Xie. 2025. Beyond pass or fail: A multi-dimensional benchmark for mobile ui navigation. *Preprint*, arXiv:2501.02863.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. 2024a. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2024b. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36.

Revanth Gangi Reddy, Sagnik Mukherjee, Jeonghwan Kim, Zhenhailong Wang, Dilek Hakkani-Tur, and Heng Ji. 2024. Infogent: An agent-based framework for web information aggregation. *arXiv preprint arXiv:2410.19054*.

Microsoft Research. 2024. Cortana research - microsoft research. Accessed: 2024-12-04.

Reworkd. 2023. Agentgpt. Accessed: 2025-02-01.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. 2024. Identifying the risks of lm agents with an lm-emulated sandbox. *Preprint*, arXiv:2309.15817.

Pascal J. Sager, Benjamin Meyer, Peng Yan, Rebekka von Wartburg-Kottler, Layan Etaiwi, Aref Enayati, Gabriel Nobel, Ahmed Abdulkadir, Benjamin F. Grewe, and Thilo Stadelmann. 2025. Ai agents for computer use: A review of instruction-based computer control, gui automation, and operator assistants. *Preprint*, arXiv:2501.16150.

Iqbal H Sarker. 2024. Llm potentiality and awareness: a position paper from the perspective of trustworthy and responsible ai modeling. *Discover Artificial Intelligence*, 4(1):40.

Sentius.AI. 2023. Sentius. Accessed: 2025-02-01.

Mobina Shahbandeh, Parsa Alian, Noor Nashid, and Ali Mesbah. 2024. Naviqate: Functionality-guided web application navigation. *Preprint*, arXiv:2409.10741.

Md Shamsujjoha, Qinghua Lu, Dehai Zhao, and Liming Zhu. 2024. Designing multi-layered runtime guardrails for foundation model based agents: Swiss cheese model for ai safety by design. *Preprint*, arXiv:2408.02205.

Peter Shaw, Mandar Joshi, James Cohan, Jonathan Berant, Panupong Pasupat, Hexiang Hu, Urvashi Khandelwal, Kenton Lee, and Kristina N Toutanova. 2023. From pixels to ui actions: Learning to follow instructions via graphical user interfaces. *Advances in Neural Information Processing Systems*, 36:34354–34370.

Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. 2023. Survey of vulnerabilities in large language models revealed by adversarial attacks. *Preprint*, arXiv:2310.10844.

Huawen Shen, Chang Liu, Gengluo Li, Xinlong Wang, Yu Zhou, Can Ma, and Xiangyang Ji. 2024a. Falcon-ui: Understanding gui before following user instructions. *arXiv preprint arXiv:2412.09362*.

Junhong Shen, Atishay Jain, Zedian Xiao, Ishan Amlekar, Mouad Hadji, Aaron Podolny, and Ameet Talwalkar. 2024b. Scribeagent: Towards specialized web agents using production-scale workflow data. *arXiv preprint arXiv:2411.15004*.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*, pages 3135–3144. PMLR.

Yucheng Shi, Wenhao Yu, Wenlin Yao, Wenhu Chen, and Ninghao Liu. 2025. Towards trustworthy gui agents: A survey. *Preprint*, arXiv:2503.23434.

Zeru Shi, Kai Mei, Mingyu Jin, Yongye Su, Chaoji Zuo, Wenyue Hua, Wujiang Xu, Yujie Ren, Zirui Liu, Mengnan Du, et al. 2024. From commands to prompts: Llm-based semantic file system for aios. *arXiv preprint arXiv:2410.11843*.

Yueqi Song, Frank Xu, Shuyan Zhou, and Graham Neubig. 2024a. Beyond browsing: Api-based web agents. *arXiv preprint arXiv:2410.16464*.

Yunpeng Song, Yiheng Bian, Yongtao Tang, Guiyu Ma, and Zhongmin Cai. 2024b. 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*, pages 1–17.

Zirui Song, Yaohang Li, Meng Fang, Zhenhao Chen, Zecheng Shi, Yuan Huang, and Ling Chen. 2024c. Mmac-copilot: Multi-modal agent collaboration operating system copilot. *arXiv preprint arXiv:2404.18074*.

Abishek Sridhar, Robert Lo, Frank F Xu, Hao Zhu, and Shuyan Zhou. 2023. Hierarchical prompting assists large language model on web navigation. *arXiv preprint arXiv:2305.14257*.

Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. 2022. Meta-gui: Towards multi-modal conversational agents on mobile gui. *arXiv preprint arXiv:2205.11029*.

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Gang Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, et al. Cradle: Empowering foundation agents towards general computer control. In *NeurIPS 2024 Workshop on Open-World Agents*.

Brian Tang and Kang G Shin. 2024. Steward: Natural language web automation. *arXiv preprint arXiv:2409.15441*.

Heyi Tao, Sethuraman TV, Michal Shlapentokh-Rothman, and Derek Hoiem. 2023. Webwise: Web interface control and sequential exploration with large language models. *arXiv preprint arXiv:2310.16042*.

TaxyAI. 2023. Taxy ai. Accessed: 2025-02-01.

Lucas-Andrei Thil, Mirela Popa, and Gerasimos Spanakis. 2024. Navigating webai: Training agents to complete web tasks with large language models and reinforcement learning. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, pages 866–874.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. 2024. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. *Preprint*, arXiv:2407.18901.

Amrita S Tulshan and Sudhir Namdeorao Dhage. 2019. Survey on virtual assistant: Google assistant, siri, cortana, alexa. In *Advances in Signal Processing and Intelligent Recognition Systems: 4th International Symposium SIRS 2018, Bangalore, India, September 19–22, 2018, Revised Selected Papers 4*, pages 190–201. Springer.

Sagar Gubbi Venkatesh, Partha Talukdar, and Srini Narayanan. 2022. Ugif: Ui grounded instruction following. *arXiv preprint arXiv:2211.07615*.

Gaurav Verma, Rachneet Kaur, Nishan Srishankar, Zhen Zeng, Tucker Balch, and Manuela Veloso. 2024. Adaptagent: Adapting multimodal web agents with few-shot learning from human demonstrations. *Preprint*, arXiv:2411.13451.

Minh Duc Vu, Han Wang, Jieshan Chen, Zhuang Li, Shengdong Zhao, Zhenchang Xing, and Chunyang Chen. 2024. Gptvoicetasker: Advancing multi-step mobile task efficiency through dynamic interface exploration and learning. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–17.

Alan Wake, Albert Wang, Bei Chen, CX Lv, Chao Li, Chengen Huang, Chenglin Cai, Chujie Zheng, Daniel Cooper, Ethan Dai, et al. 2024. Yi-lightning technical report. *arXiv preprint arXiv:2412.01253*.

Bryan Wang, Gang Li, and Yang Li. 2023a. Enabling conversational interaction with mobile ui using large language models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–17.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023b. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024a. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. *arXiv preprint arXiv:2401.16158*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024b. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Luyuan Wang, Yongyu Deng, Yiwei Zha, Guodong Mao, Qinmin Wang, Tianchen Min, Wei Chen, and Shoufa Chen. 2024c. Mobileagentbench: An efficient and user-friendly benchmark for mobile llm agents. *arXiv preprint arXiv:2406.08184*.

Maria Wang, Srinivas Sunkara, Gilles Baechler, Jason Lin, Yun Zhu, Fedir Zubach, Lei Shu, and Jindong Chen. 2024d. Webquest: A benchmark for multimodal qa on web page sequences. *arXiv preprint arXiv:2409.13711*.

Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. 2024e. Boosting llm agents with recursive contemplation for effective deception handling. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9909–9953.

Shuai Wang, Weiwen Liu, Jingxuan Chen, Weinan Gan, Xingshan Zeng, Shuai Yu, Xinlong Hao, Kun Shao, Yasheng Wang, and Ruiming Tang. 2024f. Gui agents with foundation models: A comprehensive survey. *Preprint*, arXiv:2411.04890.

Shuai Wang, Weiwen Liu, Jingxuan Chen, Yuqi Zhou, Weinan Gan, Xingshan Zeng, Yuhan Che, Shuai Yu, Xinlong Hao, Kun Shao, Bin Wang, Chuhan Wu, Yasheng Wang, Ruiming Tang, and Jianye Hao. 2025. Gui agents with foundation models: A comprehensive survey. *Preprint*, arXiv:2411.04890.

Taiyi Wang, Zhihao Wu, Jianheng Liu, Jianye Hao, Jun Wang, and Kun Shao. 2024g. Distrl: An asynchronous distributed reinforcement learning framework for on-device control agents. *arXiv preprint arXiv:2410.14803*.

Tiannan Wang, Meiling Tao, Ruoyu Fang, Huilin Wang, Shuai Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2024h. Ai persona: Towards life-long personalization of llms. *Preprint*, arXiv:2412.13103.

Xiaoqiang Wang and Bang Liu. 2024. Oscar: Operating system control via state-aware reasoning and re-planning. *arXiv preprint arXiv:2410.18963*.

Yiqin Wang, Haoji Zhang, Jingqi Tian, and Yansong Tang. 2024i. Ponder & press: Advancing visual gui agent towards general computer control. *Preprint*, arXiv:2412.01268.

Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. 2024j. Officebench: Benchmarking language agents across multiple applications for office automation. *arXiv preprint arXiv:2407.19056*.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2024k. Agent workflow memory. *arXiv preprint arXiv:2409.07429*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. 2024a. Autodroid: Llm-powered task automation in android. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 543–557.

Hao Wen, Shizuo Tian, Borislav Pavlov, Wenjie Du, Yixuan Li, Ge Chang, Shanhui Zhao, Jiacheng Liu, Yunxin Liu, Ya-Qin Zhang, and Yuanchun Li. 2024b. Autodroid-v2: Boosting slm-based gui agents via code generation. *Preprint*, arXiv:2412.18116.

Hao Wen, Hongming Wang, Jiaxuan Liu, and Yuanchun Li. 2023. Droidbot-gpt: Gpt-powered ui automation for android. *arXiv preprint arXiv:2304.07061*.

Biao Wu, Yanda Li, Meng Fang, Zirui Song, Zhiwei Zhang, Yunchao Wei, and Ling Chen. 2024a. Foundations and recent trends in multimodal mobile agents: A survey. *Preprint*, arXiv:2411.02006.

Chen Henry Wu, Rishi Shah, Jing Yu Koh, Ruslan Salakhutdinov, Daniel Fried, and Aditi Raghunathan. 2025a. Dissecting adversarial robustness of multimodal lm agents. *Preprint*, arXiv:2406.12814.

Fangzhou Wu, Shutong Wu, Yulong Cao, and Chaowei Xiao. 2024b. Wipi: A new web threat for llm-driven web agents. *Preprint*, arXiv:2402.16965.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025b. Webwalker: Benchmarking llms in web traversal. *Preprint*, arXiv:2501.07572.

Qinchen Wu, Difei Gao, Kevin Qinghong Lin, Zhuoyu Wu, Xiangwu Guo, Peiran Li, Weichen Zhang, Hengxu Wang, and Mike Zheng Shou. 2024c. Gui action narrator: Where and when did that action take place? *arXiv preprint arXiv:2406.13719*.

Qinzhuo Wu, Weikai Xu, Wei Liu, Tao Tan, Jianfeng Liu, Ang Li, Jian Luan, Bin Wang, and Shuo Shang. 2024d. Mobilevlm: A vision-language model for better intra-and inter-ui understanding. *arXiv preprint arXiv:2409.14818*.

Zhiyong Wu, Chengcheng Han, Zichen Ding, Zhenmin Weng, Zhoumianze Liu, Shunyu Yao, Tao Yu, and Lingpeng Kong. 2024e. Os-copilot: Towards generalist computer agents with self-improvement. *arXiv preprint arXiv:2402.07456*.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. 2024f. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. 2024. Guardagent: Safeguard llm agents by a guard agent via knowledge-enabled reasoning. *Preprint*, arXiv:2406.09187.

Junlin Xie, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. 2024a. Large multimodal agents: A survey. *arXiv preprint arXiv:2402.15116*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. 2024b. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*.

Mingzhe Xing, Rongkai Zhang, Hui Xue, Qi Chen, Fan Yang, and Zhen Xiao. 2024. 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*, pages 6061–6072.

Chejian Xu, Mintong Kang, Jiawei Zhang, Zeyi Liao, Lingbo Mo, Mengqi Yuan, Huan Sun, and Bo Li. 2024a. Advweb: Controllable black-box attacks on vlm-powered web agents. *Preprint*, arXiv:2410.17401.

Kevin Xu, Yeganeh Kordi, Tanay Nayak, Ado Asija, Yizhong Wang, Kate Sanders, Adam Byerly, Jingyu Zhang, Benjamin Van Durme, and Daniel Khashabi. 2024b. Tur [k] ingbench: A challenge benchmark for web agents. *arXiv preprint arXiv:2403.11905*.

Yifan Xu, Xiao Liu, Xueqiao Sun, Siyi Cheng, Hao Yu, Hanyu Lai, Shudan Zhang, Dan Zhang, Jie Tang, and Yuxiao Dong. 2024c. Androidlab: Training and systematic benchmarking of android autonomous agents. *arXiv preprint arXiv:2410.24024*.

Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. 2024d. Agenttrek: Agent trajectory synthesis via guiding replay with web tutorials. *arXiv preprint arXiv:2412.09605*.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2024e. Aguvis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454*.

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

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *Preprint*, arXiv:2310.11441.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. 2024a. Agentoccam: A simple yet strong baseline for llm-based web agents. *arXiv preprint arXiv:2410.13825*.

Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. 2024b. Watch out for your agents! investigating backdoor threats to llm-based agents. *Preprint*, arXiv:2402.11208.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. 2024c. Aria-ui: Visual grounding for gui instructions. *arXiv preprint arXiv:2412.16256*.

Yulong Yang, Xinshan Yang, Shuaidong Li, Chenhao Lin, Zhengyu Zhao, Chao Shen, and Tianwei Zhang. 2024d. Security matrix for multimodal agents on mobile devices: A systematic and proof of concept study. *Preprint*, arXiv:2407.09295.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2):100211.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, page nwae403.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. 2025. Ferret-ui: Grounded mobile ui understanding with multimodal llms. In *European Conference on Computer Vision*, pages 240–255. Springer.

Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Liqun Li, Si Qin, Yu Kang, Minghua Ma, Qingwei Lin, Saravan Rajmohan, et al. 2024a. Large language model-brained gui agents: A survey. *arXiv preprint arXiv:2411.18279*.

Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Liqun Li, Si Qin, Yu Kang, Minghua Ma, Guyue Liu, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2025. Large language model-brained gui agents: A survey. *Preprint*, arXiv:2411.18279.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2023a. Appagent: Multimodal agents as smartphone users. *arXiv preprint arXiv:2312.13771*.

Danyang Zhang, Lu Chen, and Kai Yu. 2023b. Mobile-env: A universal platform for training and evaluation of mobile interaction. *arXiv preprint arXiv:2305.08144*.

Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. 2024b. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*.

Jiaqi Zhang, Chen Gao, Liyuan Zhang, Yong Li, and Hongzhi Yin. 2024c. Smartagent: Chain-of-user-thought for embodied personalized agent in cyber world. *Preprint*, arXiv:2412.07472.

Jiayi Zhang, Chuang Zhao, Yihan Zhao, Zhaoyang Yu, Ming He, and Jianping Fan. 2024d. Mobileexperts: A dynamic tool-enabled agent team in mobile devices. *arXiv preprint arXiv:2407.03913*.

Jiwen Zhang, Jihao Wu, Yihua Teng, Minghui Liao, Nuo Xu, Xiao Xiao, Zhongyu Wei, and Duyu Tang. 2024e. Android in the zoo: Chain-of-action-thought for gui agents. *arXiv preprint arXiv:2403.02713*.

Jiwen Zhang, Yaqi Yu, Minghui Liao, Wentao Li, Jihao Wu, and Zhongyu Wei. 2024f. Ui-hawk: Unleashing the screen stream understanding for gui agents. *Preprints*.

Li Zhang, Shihe Wang, Xianqing Jia, Zhihan Zheng, Yunhe Yan, Longxi Gao, Yuanchun Li, and Mengwei Xu. 2024g. Llamatouch: A faithful and scalable testbed for mobile ui task automation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–13.

Shaoqing Zhang, Zhuosheng Zhang, Kehai Chen, Xinbei Ma, Muyun Yang, Tiejun Zhao, and Min Zhang. 2024h. Dynamic planning for llm-based graphical user interface automation. *arXiv preprint arXiv:2410.00467*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023c. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Adrian de Wynter, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. 2024i. Llm as a mastermind:

A survey of strategic reasoning with large language models. *Preprint*, arXiv:2404.01230.

Yanzhe Zhang, Tao Yu, and Diyi Yang. 2024j. Attacking vision-language computer agents via pop-ups. *Preprint*, arXiv:2411.02391.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024k. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*.

Zhizheng Zhang, Wenxuan Xie, Xiaoyi Zhang, and Yan Lu. 2023d. Reinforced ui instruction grounding: Towards a generic ui task automation api. *arXiv preprint arXiv:2310.04716*.

Zhuosheng Zhang and Aston Zhang. 2023. You only look at screens: Multimodal chain-of-action agents. *arXiv preprint arXiv:2309.11436*.

Ziniu Zhang, Shulin Tian, Liangyu Chen, and Ziwei Liu. 2024l. Mmina: Benchmarking multihop multimodal internet agents. *arXiv preprint arXiv:2404.09992*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024a. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Boyuan Zheng, Boyu Gou, Scott Salisbury, Zheng Du, Huan Sun, and Yu Su. 2024b. Webolympus: An open platform for web agents on live websites. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 187–197.

Kaizhi Zheng, Xuehai He, and Xin Eric Wang. 2024c. Minigpt-5: Interleaved vision-and-language generation via generative vokens. *Preprint*, arXiv:2310.02239.

Longtao Zheng, Zhiyuan Huang, Zhenghai Xue, Xinrun Wang, Bo An, and Shuicheng Yan. 2024d. Agentstudio: A toolkit for building general virtual agents. *arXiv preprint arXiv:2403.17918*.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. 2023. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *The Twelfth International Conference on Learning Representations*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. 2023a. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Long Li, Jialong Wu, Tiannan Wang, Shi Qiu, Jintian Zhang, Jing Chen, Ruipu Wu, Shuai Wang, Shiding Zhu, Jiyu Chen, Wentao Zhang, Xiangru Tang, Ningyu Zhang, Huajun Chen, Peng Cui, and Mrinmaya Sachan. 2023b. Agents: An open-source framework for autonomous language agents. *Preprint*, arXiv:2309.07870.

Wangchunshu Zhou, Yixin Ou, Shengwei Ding, Long Li, Jialong Wu, Tiannan Wang, Jiamin Chen, Shuai Wang, Xiaohua Xu, Ningyu Zhang, Huajun Chen, and Yuchen Eleanor Jiang. 2024a. Symbolic learning enables self-evolving agents. *Preprint*, arXiv:2406.18532.

Yifei Zhou, Qianlan Yang, Kaixiang Lin, Min Bai, Xiong Zhou, Yu-Xiong Wang, Sergey Levine, and Erran Li. 2024b. Proposer-agent-evaluator(pae): Autonomous skill discovery for foundation model internet agents. *Preprint*, arXiv:2412.13194.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. 2023. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*.

Zichen Zhu, Hao Tang, Yansi Li, Kunyao Lan, Yixuan Jiang, Hao Zhou, Yixiao Wang, Situo Zhang, Liangtai Sun, Lu Chen, et al. 2024. Moba: A two-level agent system for efficient mobile task automation. *arXiv preprint arXiv:2410.13757*.

A Related Work

(Multimodal) Large Language Models (Wake et al., 2024; Li et al., 2024a; Zheng et al., 2024c; Bai et al., 2023; Dai et al., 2022; Luo et al., 2024) have emerged as transformative tools in artificial intelligence, driving significant advancements across various domains. Zhao et al. (2023) summarize a foundational overview of LLMs. Yin et al. (2024); Zhang et al. (2024b) comprehensively reviews the progress of Multimodal LLMs. In addtion, Long et al. (2024) explores the use of synthetic data for training. Zhang et al. (2023c) presents the current state of research on the field of instruction tuning for LLMs.

With the flourishing development of (M)LLM-based Agents, numerous comprehensive surveys have emerged, offering detailed insights into various aspects of these systems. Wang et al. (2024b); Cheng et al. (2024b); Gan et al. (2024b) provides an overview of general LLM-based Agents. For the agent frameworks, Zhou et al. (2023b); Zhang et al. (2024k); Li et al. (2024d) explore methods to enhance agents' capabilities of planning, memory and multi-agents interaction. Qiao et al. (2022) presents comprehensive comparisons for LLM's reasoning abilities. Hou et al. (2023); Hu et al. (2024a); Li et al. (2024h) summarizes studies in different application fields including software engineering, game and personal assistance. Some concurrent works (Li et al., 2024g; Wu et al., 2024a; Wang et al., 2024f; Gao et al., 2024; Zhang et al., 2024a) touch on concepts that share some features with OS Agents, such as personalized agents, GUI Agents and generalist virtual agents. This work aims to provide an integrated view on the construction and evaluation of OS Agents, that leverage environments and interfaces provided by operating systems, while identifying open challenges and future directions in this domain for forthcoming studies.

B Detailed Discussions on the Construction of OS Agents

B.1 Foundation Model

As illustrated in Figure 5, training strategies that are applied in construction of foundation models for OS Agents mainly include pre-training, supervised finetuning and reinforcement learning. Table 1 summarizes the architecture and training strategies used in the recent foundation models for OS Agents.

B.2 Agent Framework

As illustrated in Figure 6, these components work together to enable OS Agents to understand, plan, remember, and interact with operating systems. Table 2 summarizes the technical characteristics of recent OS Agent frameworks, including their specific implementations across these four core components.

C Detailed Discussions on the Evaluation of OS Agents

We have provided the recent benchmarks for OS Agents in Table 3. Apart from the categorization of platforms, the environmental spaces for OS Agents to percept and take actions vary across different evaluation benchmarks. We have organized the existing benchmark environments, primarily dividing them into **static** and **interactive** categories, with the interactive environments further split into **simulated** and **real-world** settings.

Static. Static Environments, which are prevalent in early studies, are often created by caching website copies or static data, thereby establishing an offline context for evaluation. The process of setting up a static environment is quite simple, as it merely involves caching the content from real websites. Evaluations generally rely on the cached static content for tasks such as visual grounding, and only one-step action are supported. MiniWoB (Shi et al., 2017) is built on simple HTML/CSS/JavaScript pages and employs predefined simulation tasks. Mind2Web (Deng et al., 2024a) captures comprehensive snapshots of each website along with complete interaction traces, enabling seamless offline replay. ANDROIDLAB (Xu et al., 2024c) generation method combines self-exploration and manual annotation, and uses preloaded usage records in the AVD image to ensure normal usability without an internet connection. Owing to the lack of dynamic interaction and environmental feedback, such static evaluations tend to be less authentic and versatile, making them inadequate for a comprehensive assessment.

Interactive. Interactive Environments provide a more authentic scenario, characterized by their dynamism and interactivity. In contrast to static environments, OS Agents can execute a sequence of actions, receive feedback from the environment, and make corresponding adjustments. Interactive evaluation settings facilitate the evaluation of an agent's skills in more sophisticated settings. These interactive environments can be subdivided into simulated and real-world types. (1) For the *simulated environment*, FormWoB (Shi et al., 2017) created a virtual website to avoid the reproducibility issues caused by the dynamic nature of real-world environments, while Rawles et al. (2024b) developed virtual apps to assess the capabilities of OS Agents. However, these simulated environments are often overly simplistic by excluding unexpected conditions, thus failing to capture the complexity of real-world scenarios. (2) For the *real-world environment*, which is truly authentic and encompasses real websites and apps, one must consider the continuously updating nature of the environment, uncontrollable user behaviors, and diverse device setups. This scenario underscores the requirement for agents to exhibit strong generalization across real-world conditions. OSWorld (Xie et al., 2024b), for example, constructed virtual machines running Windows, Linux, and MacOS to systematically evaluate the performance of OS Agents across different operating systems. Similarly, AndroidWorld (Rawles et al., 2024a), con-

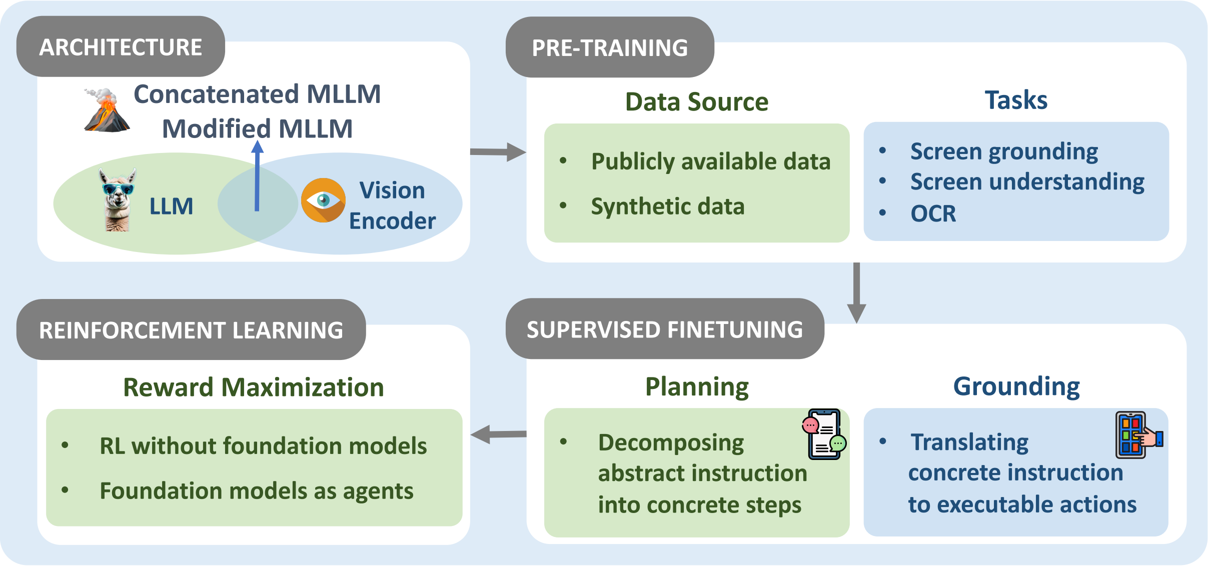


Figure 5: Summary of the content about foundation models for OS Agents in §3.1.

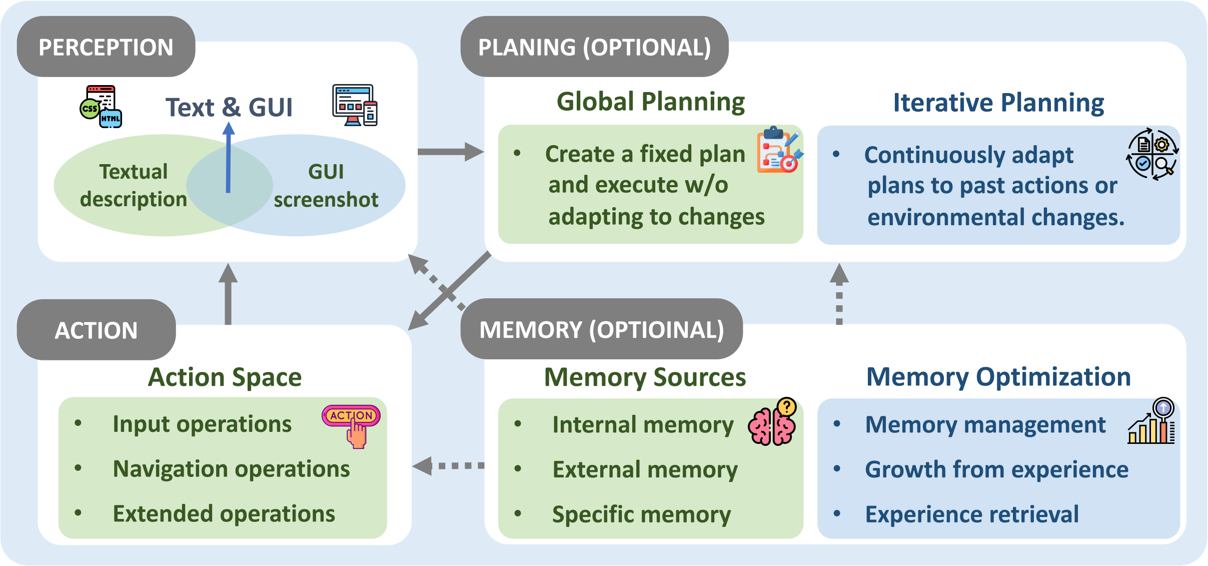


Figure 6: Summary of the content about agent frameworks for OS Agents in §3.2.

Table 1: Recent foundation models for OS Agents. Arch.: Architecture, Exist.: Existing, Mod.: Modified, Concat.: Concatenated, PT: Pre-Train, SFT: Supervised Fine-Tune, RL: Reinforcement Learning.

Model	**Arch.**	**PT**	**SFT**	**RL**	**Date**
InfiGUIAgent (Liu et al., 2025b)	Exist. MLLMs	-	✓	-	01/2025
Aria-UI (Yang et al., 2024c)	Mod. MLLMs	-	✓	-	12/2024
Iris (Ge et al., 2024)	Mod. MLLMs	✓	✓	-	12/2024
AgentTrek (Xu et al., 2024d)	Exist. MLLMs	-	✓	-	12/2024
Falcon-UI (Shen et al., 2024a)	Concat. MLLMs	-	✓	-	12/2024
AGUVIS (Xu et al., 2024e)	Exist. MLLMs	✓	✓	-	12/2024
ScribeAgent (Shen et al., 2024b)	Exist. LLMs	-	✓	-	11/2024
OS-Atlas (Wu et al., 2024f)	Exist. MLLMs	✓	✓	-	10/2024
AutoGLM (Liu et al., 2024c)	Exist. LLMs	✓	✓	✓	10/2024
EDGE (Chen et al., 2024d)	Exist. MLLMs	-	✓	-	10/2024
Ferret-UI 2 (Li et al., 2024i)	Exist. MLLMs	-	✓	-	10/2024
ShowUI (Lin et al., 2024)	Exist. MLLMs	-	✓	-	10/2024
UIX (Liu et al., 2024b)	Exist. MLLMs	-	✓	-	10/2024
TinyClick (Pawlowski et al., 2024)	Exist. MLLMs	✓	-	-	10/2024
UGround (Gou et al., 2024)	Exist. MLLMs	-	✓	-	10/2024
NNetNav (Murty et al., 2024)	Exist. LLMs	-	✓	-	10/2024
Synatra (Ou et al., 2024)	Exist. LLMs	-	✓	-	09/2024
MobileVLM (Wu et al., 2024d)	Exist. MLLMs	✓	✓	-	09/2024
UI-Hawk (Zhang et al., 2024f)	Mod. MLLMs	✓	✓	-	08/2024
GUI Action Narrator (Wu et al., 2024c)	Exist. MLLMs	-	✓	-	07/2024
MobileFlow (Nong et al., 2024)	Mod. MLLMs	✓	✓	-	07/2024
VGA (Meng et al., 2024)	Exist. MLLMs	-	✓	-	06/2024
OdysseyAgent (Lu et al., 2024)	Exist. MLLMs	-	✓	-	06/2024
Textual Foresight (Burns et al., 2024)	Concat. MLLMs	✓	✓	-	06/2024
WebAI (Thil et al., 2024)	Concat. MLLMs	-	✓	✓	05/2024
GLAINTEL (Fereidouni et al., 2024)	Exist. MLLMs	-	-	✓	04/2024
Ferret-UI (You et al., 2025)	Exist. MLLMs	-	✓	-	04/2024
AutoWebGLM (Lai et al., 2024)	Exist. LLMs	-	✓	✓	04/2024
Patel et al. (2024)	Exist. LLMs	-	✓	-	03/2024
ScreenAI (Baechler et al., 2024)	Exist. MLLMs	✓	✓	-	02/2024
Dual-VCR (Kil et al., 2024)	Concat. MLLMs	-	✓	-	02/2024
SeeClick (Cheng et al., 2024a)	Exist. MLLMs	✓	✓	-	01/2024
CogAgent (Hong et al., 2024)	Mod. MLLMs	✓	✓	-	12/2023
ILuvUI (Jiang et al., 2023)	Mod. MLLMs	-	✓	-	10/2023
RUIG (Zhang et al., 2023d)	Concat. MLLMs	-	-	✓	10/2023
WebAgent (Iong et al., 2024)	Concat. LLMs	✓	✓	-	07/2023
WebGUM (Furuta et al., 2023)	Concat. MLLMs	-	✓	-	05/2023

Table 2: Recent agent frameworks for OS Agents. Text: Textual Description, Screen: GUI Screenshots, Vis: Visual Description, Sem: Semantic Description, Dual: Dual Description, Glob: Global, Iter: Iterative, Growth: Growth Experience, Retrival: Experience Retrieval, Manage: Management, Input: Input Operations, Nav: Navigation Operations, Ext: Extended Operations.

Agent	Perception	Planning	Memory	Action	Date
CowPilot (Huq et al., 2025)	-	Iter	GrowthE	InputO, NavO	01/2025
UI-TARS (Qin et al., 2025)	GScreen, VisD	Iter	ERetri	InputO, NavO	01/2025
R2D2 (Huang et al., 2025)	-	Iter	ERetri	InputO, NavO	01/2025
AutoDroid-V2 (Wen et al., 2024b)	Text	-	ManageA	ExtO	12/2024
PAE (Zhou et al., 2024b)	GScreen, VisD	Iter	GrowthE	InputO, NavO	12/2024
SmartAgent (Zhang et al., 2024c)	GScreen	Iter	-	InputO, NavO	12/2024
Ponder & Press (Wang et al., 2024i)	GScreen	-	-	InputO, NavO	12/2024
AdaptAgent (Verma et al., 2024)	-	Iter	ERetri	InputO	11/2024
Claude Computer Use (Hu et al., 2024b)	GScreen	Iter	-	InputO, NavO, ExtO	11/2024
WebDreamer (Gu et al., 2024)	-	Iter	ERetri	InputO, NavO	11/2024
WebOlympus (Zheng et al., 2024b)	GScreen, SemD	-	ERetri	InputO, NavO	11/2024
OpenWebVoyager (He et al., 2024b)	GScreen, SemD	-	-	InputO, NavO	10/2024
OSCAR (Wang and Liu, 2024)	GScreen, DualD	Iter	GrowthE	ExtO	10/2024
Auto-Intent (Kim et al., 2024b)	Text	-	GrowthE	InputO, NavO	10/2024
VisionTasker (Song et al., 2024b)	Text	Iter	ERetri, GrowthE, ManageA	InputO, NavO	10/2024
D-PoT (Zhang et al., 2024h)	-	Iter	-	InputO, NavO	10/2024
Agent-E with Self-Verifier (Azam et al., 2024)	-	-	-	InputO, NavO	10/2024
PUManageA (Cai et al., 2024)	Text	-	-	InputO, NavO, ExtO	10/2024
AgentOccam (Yang et al., 2024a)	Text	Iter	ManageA	InputO, NavO	10/2024
Agent S (Agashe et al., 2024)	GScreen, SemD	GlobL	ERetri, GrowthE, ManageA	InputO, NavO	10/2024
ClickAgent (Hoscilowicz et al., 2024)	GScreen	Iter	GrowthE	InputO, NavO	10/2024
LSFS (Shi et al., 2024)	GScreen, SemD	-	-	ExtO	09/2024
NaviQate (Shahbandeh et al., 2024)	GScreen, SemD	-	-	InputO	09/2024
PeriGuru (Fu et al., 2024)	GScreen, DualD	Iter	ERetri, GrowthE	InputO, NavO	09/2024
Steward (Tang and Shin, 2024)	Text	-	ERetri, ManageA	InputO, NavO	09/2024
Navi (Bonatti et al., 2024)	GScreen, VisD	Iter	ERetri, GrowthE	InputO	09/2024
AWM (Wang et al., 2024k)	Text	-	-	InputO	09/2024
OpenWebAgent (Iong et al., 2024)	GScreen, DualD	-	-	InputO	08/2024
UI-Hawk (Zhang et al., 2024f)	-	-	GrowthE	InputO, NavO	08/2024
AutoWebGlobLM (Lai et al., 2024)	Text	-	-	InputO, NavO	08/2024
OmniParser (Lu et al., 2024)	GScreen, DualD	-	GrowthE	InputO, NavO	08/2024
LLMCI (Barham and Fasha, 2024)	GScreen, SemD	-	-	ExtO	07/2024
Agent-E (Abuelsaad et al., 2024)	Text	Iter	GrowthE, ManageA	InputO, NavO	07/2024
Search-Agents (Koh et al., 2024b)	-	Iter	-	InputO, NavO	07/2024
CAAP Agent (Cho et al., 2024)	Text	Iter	-	InputO, NavO	06/2024
M3A (Rawles et al., 2024a)	GScreen, VisD	Iter	ManageA	InputO	05/2024
Domain-General Evaluators (Pan et al.)	GScreen, SemD	-	ERetri	InputO, NavO	04/2024
PromptRPA (Huang et al., 2024a)	-	-	ManageA	InputO, NavO	04/2024
Cradle (Tan et al.)	GScreen	Iter	ERetri, GrowthE, ManageA	ExtO	03/2024
CoAT (Zhang et al., 2024e)	GScreen	Iter	-	InputO, NavO	03/2024
Self-ManageAP (Deng et al., 2024b)	-	Iter	ERetri	InputO	02/2024
OS-Copilot (Wu et al., 2024e)	Text	GlobL	ERetri, GrowthE	InputO, ExtO	02/2024
CoCo-Agent (Ma et al., 2024c)	GScreen, SemD	-	GrowthE	InputO, NavO	02/2024
ScreenAgent (Niu et al., 2024)	GScreen	Iter	ERetri, GrowthE	InputO, NavO	02/2024
SeeClick (Cheng et al., 2024a)	GScreen	-	-	InputO	01/2024
Mobile-Agent (Wang et al., 2024a)	GScreen, SemD	Iter	GrowthE	InputO, NavO	01/2024
WebVoyager (He et al., 2024a)	GScreen, VisD	Iter	ManageA	InputO, NavO	01/2024
AIA (Ding, 2024)	GScreen, VisD	GlobL	-	InputO, NavO	01/2024
SeeAct (Zheng et al., 2024a)	GScreen, VisD	-	GrowthE	InputO	01/2024
AppAgent (Zhang et al., 2023a)	GScreen, DualD	Iter	GrowthE	InputO, NavO	12/2023
ACE (Gao et al., 2023)	Text	GlobL	GrowthE	InputO, NavO	12/2023
MobileGPT (Lee et al., 2023)	Text	GlobL	ManageA	InputO, NavO	12/2023
LLMPA (Li et al., 2023)	Text	GlobL	GrowthE	InputO, NavO	12/2023
MM-Navigator (Yan et al., 2023)	GScreen, VisD	-	ManageA	InputO, NavO	11/2023
WebWise (Tao et al., 2023)	Text	-	ManageA	InputO, NavO	10/2023
Li et al. (2023)	Text	Iter	GrowthE	InputO, NavO	10/2023
Laser (Ma et al., 2023)	Text	Iter	GrowthE	InputO, NavO	09/2023
AutoDroid (Wen et al., 2024a)	Text	-	-	InputO, NavO	08/2023
MINDACT (Deng et al., 2024a)	Text	-	-	InputO	06/2023
Synapse (Zheng et al., 2023)	-	-	ManageA	InputO	06/2023
ASH Prompting (Sridhar et al., 2023)	Text	-	-	InputO, NavO	05/2023
GUI-TOD (Sun et al., 2022)	GScreen, SemD	-	-	InputO, NavO	05/2022
SheetCopilot (Li et al., 2024b)	Text	Iter	GrowthE	ExtO	05/2023
RCI (Kim et al., 2024a)	-	Iter	GrowthE	InputO, NavO	03/2023
Wang et al. (2023a)	Text	-	-	InputO	09/2022

ducted tests on real apps using Android emulators, highlighting the importance of evaluating agents under diverse and realistic conditions.

D Detailed Discussions on Products of OS Agents

Over the past few years, OS Agent-related products have undergone notable evolution, characterized by a clear trend toward platform diversification and functional stratification. This progression reflects the growing demand for more sophisticated and versatile agent-based solutions across various computing environments. From a platform perspective, the current mainstream forms can be categorized into three types: **browser-based** (e.g., DeepMind's Project Mariner (DeepMind, 2024), Taxy AI (TaxyAI, 2023)), **computer-based** (e.g., Anthropic's Computer Use (Anthropic, 2024a), Self-Operating Computer (OthersideAI, 2023)), and **phone-based** (e.g., Apple Intelligence (Apple, 2024), Zhipu's AutoGLM (Liu et al., 2024c) cross-app control). Browser-based products, such as web browser plugins, with their low invasiveness, have become an early exploration direction (e.g., AgentGPT (Reworkd, 2023) in 2023), while the newly released Apple Intelligence and AutoGLM in 2024 highlight the trend of deep integration in the mobile domain, achieving scenario closure by accessing contacts and enabling multi-app collaboration.

In terms of functional positioning, products are gradually diverging into two paths: **task execution-oriented** and **search-oriented**. The former focuses on cross-application operational capabilities, such as AutoGLM controlling applications like Taobao and WeChat, and Computer Use managing PC workflows. The latter, exemplified by OpenAI DeepResearch (OpenAI, 2025), concentrates on automatic information integration, addressing the blind spots of traditional search engines in handling tabular data. Notably, early projects (pre-2023) mostly focused on single-function prototype validation (e.g., Self-Operating Computer's GPT-4V command-line experiments), while products in 2024 and 2025 emphasize multimodal interaction (e.g., Project Mariner's voice control + decision visualization) and system permission upgrades (e.g., Siri's deep access to iOS notifications/schedules after its redesign in Apple Intelligence).

Timeline-wise, 2023 can be seen as a period of technological validation, with startups exploring basic interaction frameworks through browser plugins (e.g., AgentGPT) or CLI tools (e.g., Self-Operating Computer). By 2024, leading manufacturers began embedding agent capabilities into the operating system's underlying layers (e.g., Apple Intelligence), enhancing personalized services through RAG (e.g., Apple's contact understanding) and optimizing complex task decomposition with tree search (e.g., Project Mariner). This shift from the tool layer to the system layer, and from passive response to active service, marks the transition of OS Agents from technological demonstrations to actual productivity transformations.

E Detailed Discussions on Challenge & Future

E.1 Safety & Privacy

A recent report (Park, 2024) highlighted a notable case where a human player successfully outwitted the Freysa AI agent in a $47,000 crypto challenge, underscoring vulnerabilities even in advanced AI systems and emphasizing the need to address these security risks. This incident aligns with broader concerns as (M)LLMs are increasingly integrated into diverse domains, such as healthcare, education, and autonomous systems, where security has become a critical issue. This growing adoption has led to numerous studies (Deng et al., 2024c; Gan et al., 2024a; Yao et al., 2024; Shayegani et al., 2023; Cui et al., 2024; Wang et al., 2024e; Neel and Chang, 2024) investigating the security risks associated with LLMs and their applications. In particular, some research has delved into the challenges faced by OS Agents regarding security risks. The following subsections discuss existing research on the security aspects of OS Agents. §E.1.1 analyzes various attack strategies targeting OS Agents, §E.1.2 explores existing defense mechanisms and limitations, and §E.1.3 reviews existing security benchmarks designed to assess the robustness and reliability of OS Agents.

E.1.1 Attack

Several researchers have investigated attacks targeting OS Agents. Wu et al. (2024b) identified a novel threat called Web Indirect Prompt Injection (WIPI), in which adversaries indirectly control LLM-based Web Agents by embedding natural language instructions into web pages. Recent findings (Wu et al., 2025a) further uncovered security risks for MLLMs, illustrating how adversaries can generate adversarial images that cause the cap-

Table 3: Recent benchmarks for OS Agents. We divided the Benchmarks into three sections based on the Platform (as mentioned in §4.2.1) and sorted them by release date. Grd: GUI Grounding, Info: Information Processing, Code: Code Generation.

Benchmark	**Platform**	**Benchmark Setting**	**Environment**	**Task**	**Date**
WindowsAgentArena (Bonatti et al., 2024)	Computer	Interactive	Real World	Agent	09/2024
OfficeBench (Wang et al., 2024j)	Computer	Interactive	Real World	Agent	07/2024
Spider2-V (Cao et al., 2024)	Computer	Interactive	Real World	Agent, Code	07/2024
VIBench (Song et al., 2024c)	Computer	Static	-	Agent	04/2024
OSimulatedorld (Xie et al., 2024b)	Computer	Interactive	Real World	Agent	04/2024
OmniACT (Kapoor et al., 2024)	Computer	Static	-	Code	02/2024
ASSIStaticGUI (Gao et al., 2023)	Computer	Interactive	Real World	Agent	12/2023
WebWalkerQA (Wu et al., 2025b)	Phone	Interactive	Simulated	Grd, Info	01/2025
Sphinx (Ran et al., 2025)	Phone	Interactive	Real World	Grd,Agent	01/2025
A3 (Chai et al., 2025)	Phone	Interactive	Real World	Info,Agent	01/2025
SmartSpot (Zhang et al., 2024c)	Phone	Static	-	Grd,Agent	12/2024
AndroidLab (Xu et al., 2024c)	Phone	Interactive	Real World	Grd,Info,Agent	10/2024
SPA-Bench (Chen et al., 2024a)	Phone	Interactive	Real World	Info,Agent	10/2024
MobBench (Zhu et al., 2024)	Phone	Interactive	Simulated	Agent	09/2024
AppWorld Benchmark (Trivedi et al., 2024)	Phone	Interactive	Simulated	Grd, Code	07/2024
Expert Eval (Zhang et al., 2024d)	Phone	Interactive	Real World	Info,Agent	07/2024
AMEX (Chai et al., 2024)	Phone	Static	-	Agent	07/2024
GUI Odyssey (Lu et al., 2024)	Phone	Interactive	Real World	Agent	06/2024
AndroidControl (Li et al.)	Phone	Static	-	Agent	06/2024
AndroidWorld (Rawles et al., 2024a)	Phone	Interactive	Real World	Agent	05/2024
Android-50 (Bishop et al., 2024)	Phone	Interactive	Real World	Agent	05/2024
B-MoCA (Lee et al., 2024b)	Phone	Interactive	Real World	Agent	04/2024
LlamaTouch (Zhang et al., 2024g)	Phone	Interactive	Real World	Agent	04/2024
You et al. (2025)	Phone	Static	-	Grd, Info	04/2024
AndroidArena (Xing et al., 2024)	Phone	Interactive	Real World	Agent	02/2024
Mobile-Eval (Wang et al., 2024a)	Phone	Interactive	Real World	Agent	01/2024
iOS Screen Navigation(Yan et al., 2023)	Phone	Static	-	Agent	11/2023
DroidTask(Wen et al., 2024a)	Phone	Interactive	Real World	Agent	08/2023
AInteractiveW (Rawles et al., 2024b)	Phone	Static	-	Agent	07/2023
Wen et al. (2023)	Phone	Interactive	Real World	Agent	04/2023
UGInfo-DataSet (Venkatesh et al., 2022)	Phone	Static	-	Agent	11/2022
META-GUI (Sun et al., 2022)	Phone	Static	-	Agent	05/2022
MoTInfo (Burns et al., 2022)	Phone	Static	-	Agent	02/2022
PIXELHELP (Li et al., 2020)	Phone	Interactive	Real World	Grd	05/2020
NovelScreenSpot (Fan et al., 2025b)	Browser	Static	-	Grd	01/2025
PersonalWAB (Cai et al., 2024)	Browser	Interactive	Simulated	Info, Agent	10/2024
WebQuest (Wang et al., 2024d)	Browser	Static	-	Info	09/2024
Mind2Web-Live (Pan et al., 2024)	Browser	Interactive	Real World	Info, Agent	06/2024
WebSuite (Li and Waldo, 2024)	Browser	Interactive	Simulated	Info, Agent	06/2024
MMInA (Zhang et al., 2024l)	Browser	Interactive	Real World	Info, Agent	04/2024
AutoWebBench (Lai et al., 2024)	Browser	Static	-	Info, Agent	04/2024
GroundUI (Zheng et al., 2024d)	Browser	Static	-	Grd	03/2024
TurkingBench (Xu et al., 2024b)	Browser	Interactive	Real World	Agent	03/2024
WorkArena (Drouin et al., 2024)	Browser	Interactive	Real World	Info, Agent	03/2024
MT-Mind2Web (Deng et al., 2024b)	Browser	Static	-	Agent	02/2024
WebLINX (Lù et al., 2024)	Browser	Static	-	Info, Agent	02/2024
He et al. (2024a)	Browser	Interactive	Real World	Grd, Info, Agent	01/2024
Visualwebarena (Koh et al., 2024a)	Browser	Interactive	Real World	Info, Agent	01/2024
Agentboard (Ma et al., 2024a)	Browser	Interactive	Real World	Agent	01/2024
WebVLN-v1 (Chen et al., 2024b)	Browser	Interactive	Real World	Info, Agent	12/2023
CompWoB (Furuta et al., 2024)	Browser	Static	-	Agent	11/2023
WebArena (Zhou et al., 2023a)	Browser	Interactive	Real World	Agent	07/2023
Mind2Web (Deng et al., 2024a)	Browser	Static	-	Info, Agent	06/2023
WikiHow (Zhang et al., 2023b)	Browser	Interactive	Simulated	Info	05/2023
WebShop (Yao et al., 2022)	Browser	Static	-	Agent	07/2022
PhraseNode (Pasupat et al., 2018)	Browser	Static	-	Grd	08/2018
MiniWoB (Shi et al., 2017)	Browser	Static	-	Agent	08/2017
FormWoB (Shi et al., 2017)	Browser	Interactive	Simulated	Agent	08/2017

Table 4: Recent commercial products for OS Agents.

Name	**Affiliation**	**Platform**	**Target**	**Date**
Operator (OpenAI, 2025)	OpenAI	Browser	Execution	01/2025
Project Mariner (DeepMind, 2024)	Google Deepmind	Browser	Execution	12/2024
Apple Intelligence (Apple, 2024)	Apple	Computer, Phone	Execution	10/2024
AutoGLM (Liu et al., 2024c)	Zhipu.AI	Phone, Browser	Execution	10/2024
Computer Use (Anthropic, 2024a)	Anthropic	Computer	Execution	10/2024
VisioPilot (N4NO, 2024)	N4NO	Browser	Execution	10/2024
Ottogrid (Cognosys, 2024)	Cognosys	Browser	Execution	06/2024
iMean (iMean.AI, 2024)	iMean.AI	Browser	Search	01/2024
Self-Operating Computer (OthersideAI, 2023)	OthersideAI	Computer	Execution	11/2023
Sentius (Sentius.AI, 2023)	Sentius.AI	Browser	Execution	07/2023
AgentGPT (Reworkd, 2023)	Reworkd	Browser	Search	04/2023
Taxy AI (TaxyAI, 2023)	Taxy AI	Browser	Execution	04/2023

tioner to produce adversarial captions, ultimately leading the agents to deviate from the user's intended goals. Similar vulnerabilities have been identified in other studies. Ma et al. (2024b) introduced an attack method called environmental injection, highlighting that advanced MLLMs are vulnerable to environmental distractions, which can cause agents to perform unfaithful behaviors. Expanding on the concept, Liao et al. (2024) executed an environmental injection attack by embedding invisible malicious instructions within web pages, prompting the agents to assist adversaries in stealing users' personal information. Xu et al. (2024a) further advanced this approach by leveraging malicious instructions generated by an adversarial prompter model, trained on both successful and failed attack data, to mislead MLLM-based Web Agents into executing targeted adversarial actions. Yang et al. (2024b) investigated backdoor threats in LLM-based agents and implemented this threat in web shopping and tool utilization tasks. Their work reveals significant security vulnerabilities in LLM-based agents when facing various covert forms of backdoor attacks.

Other studies have explored security issues in specific environments. Zhang et al. (2024j) explored adversarial pop-up window attacks on MLLM-based Web Agents, demonstrating how this method interferes with the decision-making process of the agents. Kumar et al. (2024) investigated the security of refusal-trained LLMs when deployed as browser agents. Their study found that these models' ability to reject harmful instructions in conversational settings does not effectively transfer to browser-based environments. Moreover, existing attack methods can successfully bypass their security measures, enabling jailbreaking. Yang et al. (2024d) proposed a security threat matrix for agents running on mobile devices, systematically examining the security issues of MLLM-based Mobile Agents and identifying four realistic attack paths and eight attack methods.

E.1.2 Defense

Although several security frameworks have been developed for LLM-based Agents (Ruan et al., 2024; Hua et al., 2024; Fang et al., 2024; Xiang et al., 2024; Shamsujjoha et al., 2024), studies on defenses specific to OS Agents (Pedro et al., 2023) remain limited. Bridging this gap requires the development of robust defense mechanisms tailored to the vulnerabilities of OS Agents, such as injection attacks, backdoor exploits, and other potential threats. Future research could prioritize these areas, focusing on developing comprehensive and scalable security solutions for OS Agents.

E.1.3 Benchmark

Several security benchmarks (Levy et al., 2024; Lee et al., 2024a; Debenedetti et al., 2024; Andriushchenko et al., 2024) have been introduced to evaluate the robustness of OS Agents in various scenarios. The online benchmark ST-WebAgentBench (Levy et al., 2024) has been developed to systematically assess the safety and trustworthiness of web agents within enterprise environments. It focuses on six key dimensions of reliability, offering a comprehensive framework for evaluating agent behavior in high-risk contexts. Similarly, a benchmarking platform named MobileSafetyBench (Lee et al., 2024a) has been developed to assess the security of LLM-based Mobile Agents, focusing on evaluating their performance in handling safety-critical tasks within Android environments, including interactions with messaging and banking applications. (Debenedetti et al., 2024) Introduced AgentDojo, a dynamic environment for evaluating prompt injection attacks and defenses against LLM-based agents. (Andriushchenko et al., 2024) propose the AgentHarm benchmark to measure the harmfulness of LLM agents executing malicious tasks, including 110 distinct tasks across 11 harm categories, such as fraud, cybercrime, and harassment.

E.2 Personalization & Self-Evolution

Much like Jarvis as Iron Man's personal assistant in the movies, developing personalized OS Agents has been a long-standing goal in AI research. A personal assistant is expected to continuously adapt and provide enhanced experiences based on individual user preferences. OpenAI's memory feature[7] has made strides in this direction, but many (M)LLMs today still perform insufficient in providing personalized experience to users and self-evolving over user interactions.

Early works (Wang et al., 2023b; Zhu et al., 2023) allowed LLM-based Agents to interact with environments of games, summarizing experiences into text, thus accumulating memory and facilitating self-evolution (Zhou et al., 2024a). For example, Wang et al. (2023b) demonstrated the potential for agents to adapt and evolve through experience. Later, researchers applied these principles to the OS Agent domain (Zhang et al., 2023a; Li et al., 2024e; Wu et al., 2024e). These efforts validated the feasibility of memory mechanisms in OS Agents. Wang et al. (2024h) introduces a general framework for lifelong personalization of LLM-based Agents, also methods to synthesize realistic benchmarks and robust evaluation metrics. Some products, such as LangMem (langchain-ai, 2025) and Mem0 (Chhikara et al., 2025), offer a memory layer as standalone solutions for LLM-based agents, enabling personalization and self-evolution.

However, expanding the modalities of memory from text to other forms, such as images, voice, presents significant challenges. Managing and retrieving this memory effectively also remains an open issue. We believe that in the future, overcoming these challenges will enable OS Agents to provide more personalized, dynamic, and context-aware assistance, with more sophisticated self-evolution mechanisms that continually adapt to the user's needs and prefernces.

F Supplementary Materials

F.1 Ethical Concerns in Developing OS Agents

The development of OS agents necessitates careful consideration of several ethical concerns beyond security and privacy. A primary issue is the potential for bias and unfairness; agents may inherit biases from their training data, leading to discriminatory actions or perpetuating societal inequalities(Sarker, 2024; Li et al., 2024f). Ensuring representative data and continuous improvement are vital to mitigate such biases (Ferrara, 2023). Furthermore, the societal impact of OS agents, including effects on the labor market and the consequences of erroneous agent decisions leading to financial loss or data corruption, must be addressed (Wang et al., 2025; Zhang et al., 2025). The automated decision-making by OS agents also raises questions of accountability and transparency, particularly when errors occur (Zhang et al., 2025). There are concerns about the potential for agents to misinform users or unduly influence their beliefs (Sarker, 2024; Shi et al., 2025). Developing agents that are culturally and socially aware, catering to diverse user needs and contexts, is also a significant ethical challenge (Shi et al., 2025). These multifaceted ethical dimensions highlight the need for robust guidelines, transparency in agent operations, and ongoing research into the responsible development and deployment of OS agents (Sarker, 2024; Zhang et al., 2024a; Shi et al., 2025).

F.2 Distinguishing Our Survey from Contemporary OS Agent Research

The rapidly evolving field of Large Language Model (LLM) based OS Agents has recently seen a surge in survey papers. Our work distinguishes itself from several concurrent surveys, primarily submitted within approximately one month of our own, through two main contributions: 1) a

[7] https://openai.com/index/memory-and-new-controls-for-chatgpt/

Table 5: Related Works Published Within the Same Period.

Title	Date Submitted to arXiv
Large Multimodal Agents: A Survey (Xie et al., 2024a)	23/02/2024
Foundations and Recent Trends in Multimodal Mobile Agents: A Survey (Wu et al., 2024a)	04/11/2024
GUI Agents with Foundation Models: A Comprehensive Survey (Wang et al., 2024f)	07/11/2024
Large Language Model-Brained GUI Agents: A Survey (Zhang et al., 2024a)	27/11/2024
OS Agents: A Survey on MLLM-based Agents for Computer, Phone and Browser Use (Ours)	14/12/2024
GUI Agents: A Survey (Zhang et al., 2024a)	18/12/2024
LLM-Powered GUI Agents in Phone Automation: Surveying Progress and Prospects (Liu et al., 2025a)	05/01/2025
AI Agents for Computer Use: A Review of Instruction-based Computer Control, GUI Automation, and Operator Assistants (Sager et al., 2025)	27/01/2025

broader conceptualization of OS Agents, which encompasses agents based on (Multimodal) LLMs interacting with various interfaces (GUIs and APIs) across diverse digital platforms (computers, phones, and browsers); and 2) a clear taxonomy and in-depth analysis on learning approaches (tuning and prompt-based methods) and evaluation based on a large number of recent papers in a short length. We provide a detailed comparison with these related surveys in Table 5 in chronological order of their appearance or our awareness.

- **Large Multimodal Agents: A Survey**. This survey focuses broadly on Large Multimodal Agents, with GUI automation as one specific application area (see its Section 6.1). In contrast, our work centers on OS Agents as a specific domain, encompassing specifically their current state and future development. Unlike this survey, we are not limited to Large Multimodal Agents but also include LLM-based agents, offering a wider scope within the OS Agent context.

- **Foundations and Recent Trends in Multimodal Mobile Agents: A Survey.** This survey concentrates exclusively on Mobile Agents, whereas our paper spans multiple digital platforms, including computers, phones, and browsers, bringing a wider survey for readers.

- **GUI Agents with Foundation Models: A Comprehensive Survey**. Our survey provides broader coverage of the literature while delivering enhanced analytical depth. Beyond sheer volume, we provide a clearer taxonomy, more detailed analysis, and more insights. For instance, in our Section 4, we give a deep analysis of Evaluation Benchmarks and metrics.

- **Large Language Model-Brained GUI Agents: A Survey**. In our paper, we conduct the survey from the perspective of "OS Agents", a unifying concept for agents operating across computers, phones, and browsers with both GUIs and APIs—encompassing many terms including GUI Agents, API Agents, Computer-Using Agents, Mobile Agents, Web Agents, and more. We argue that OS Agents leverage a range of OS-provided interfaces (not just GUIs but also APIs (Wu et al., 2024e; Song et al., 2024a)) to automate tasks across different digital platforms. Our survey is from this broader perspective, different from their focus on GUI Agents. Moreover, Our survey focuses on learning approaches (tuning or prompt-based method) and evaluation of OS Agents, offering a clear taxonomy and insights within a short length format prioritizing depth over exhaustive coverage, making it accessible to newcomers seeking an at-a-glance understanding, compared with their much longer survey.

- **GUI Agents: A Survey**. Our survey offers clear taxonomy and a broader scope based on much more related paper. For instance, we discuss the memory module, which is crucial for building agents taking long-horizon tasks and self-evolution seperately.

- **LLM-Powered GUI Agents in Phone Automation: Surveying Progress and Prospects**. Similar to (2), this survey focuses on GUI Agents in phone automation, while ours encompasses multiple digital platforms—computers, phones, and browsers—providing a more broader view of OS Agents.

- **AI Agents for Computer Use: A Review of Instruction-based Computer Control, GUI**

Automation, and Operator Assistants. Their review focuses on Computer Control Agents (CCAs) executing complex actions on personal computers or mobile devices via GUIs. Our survey, however, covers a wider array of digital platforms (computers, phones, and browsers) and incorporates both GUI and API-based interactions, offering a more comprehensive perspective.

F.3 Performance of OS Agents on Real-World Benchmarks

Table 6: Performance of OS Agents on Real-World Benchmarks. We present the performance of OS Agents, including both Research Work and Commercial Products, on recently recognized benchmarks. The results are sourced from publicly available Leader Boards.

Platform/ Benchmark	Model		Metric(SR)
Computer/ OSWorld	Commercial Product	OpenAI Operator	38.10%
		Agent S2	34.50%
		Claude Computer Use	22.00%
	Research Work	OSCAR (GPT-4o)	24.50%
		UI-TARS-72B	18.80%
		OS-Atlas-7B (GPT-4o as planner)	14.60%
		Cradle (GPT-4o)	10.50%
		AGUVIS-72B	10.26%
		SeeClick (GPT-4o)	9.21%
		Qwen2.5-VL-72B-Instruct	8.83%
Phone/ AndroidWorld	Commercial Product	Agent S2	50.00%
		Claude Computer Use	27.90%
	Research Work	OSCAR (GPT-4o)	61.60%
		APP Agent (GPT-4o)	59.90%
		UI-TARS-72B-SFT	46.60%
		Mobile Agent (GPT-4o)	40.80%
		Qwen2.5-VL-72B-Instruct	35.00%
		UGround (GPT-4o)	32.80%
Browser/ WebArena	Commercial Product	IBM CUGA	61.70%
		OpenAI Operator	58.10%
	Research Work	ScribeAgent	53.00%
		AgentOccam	45.70%
		AgentTrek-1.0-32B	22.40%
		AutoWebGLM	18.20%

Evaluating the comparative effectiveness of different OS Agents on real-world tasks is essential for understanding their current capabilities and guiding future development. To this end, we have undertaken a detailed examination and comparison of several prominent real-world task benchmarks. While the academic community is still in the process of establishing universally recognized, comprehensive benchmarks specifically for OS Agents that span all functionalities and platforms, the existing benchmarks provide valuable initial insights. These selected benchmarks, though often platform-specific (computer, phone, or browser), are derived from real-world task scenarios and have gained notable traction for evaluating both commercial products and research prototypes.

Table 6 summarizes preliminary success rates of various OS Agents on these established benchmarks. It is important to note that direct comparisons across different benchmarks or even different agents within the same benchmark can be challenging due to variations in evaluation protocols, task complexities, and the specific versions of models or products tested. Nevertheless, this compilation offers a snapshot of the current landscape.