# A Survey of WebAgents: Towards Next-Generation AI Agents for Web Automation with Large Foundation Models

Liangbo Ning*
The Hong Kong
Polytechnic University
Hong Kong SAR
BigLemon1123@gmail.com

Ziran Liang*
The Hong Kong
Polytechnic University
Hong Kong SAR
ziran.liang@connect.polyu.hk

Zhuohang Jiang
The Hong Kong
Polytechnic University
Hong Kong SAR
zhuohang.jiang@outlook.com

Haohao Qu
The Hong Kong
Polytechnic University
Hong Kong SAR
haohao.qu@connect.polyu.hk

Yujuan Ding
The Hong Kong
Polytechnic University
Hong Kong SAR
dingyujuan385@gmail.com

Wenqi Fan†
The Hong Kong
Polytechnic University
Hong Kong SAR
wenqifan03@gmail.com

Xiao-yong Wei
The Hong Kong
Polytechnic University
Hong Kong SAR
cs007.wei@polyu.edu.hk

Shanru Lin
City University of Hong
Kong
Hong Kong SAR
lllam32316@gmail.com

Hui Liu
Michigan State University
Michigan, USA
liuhui7@msu.edu

Philip S. Yu
University of Illinois at
Chicago
Chicago, USA
psyu@uic.edu

Qing Li
The Hong Kong
Polytechnic University
Hong Kong SAR
qing-prof.li@polyu.edu.hk

## Abstract

With the advancement of web techniques, they have significantly revolutionized various aspects of people's lives. Despite the importance of the web, many tasks performed on it are repetitive and time-consuming, negatively impacting the overall quality of life. To efficiently handle these tedious daily tasks, one of the most promising approaches is to advance autonomous agents to incorporate human-like intelligence based on Artificial Intelligence (AI) techniques, referred to as **AI Agents**. AI Agents offer significant advantages in handling such tasks since they can operate continuously without fatigue or performance degradation. Therefore, leveraging AI Agents – termed **WebAgents** in the context of web – to automatically assist people in handling tedious daily tasks can dramatically enhance productivity and efficiency. Recently, Large Foundation Models (**LFMs**) containing billions of parameters have exhibited human-like language understanding and reasoning capabilities, showing proficiency in performing various complex tasks. This naturally raises the question: '*Can LFMs be utilized to develop powerful AI Agents that automatically handle web tasks, providing significant convenience to users?*' To fully explore the potential of LFMs, extensive research has emerged on WebAgents designed to complete daily web tasks according to user instructions, significantly enhancing the convenience of daily human life. In this survey[1], we comprehensively review existing research studies on WebAgents across three key aspects: architectures, training, and trustworthiness. Additionally, several promising directions for future research are explored to provide deeper insights.

## CCS Concepts

• **Computing methodologies → Intelligent agents**; • **Information systems → Web applications**.

## Keywords

WebAgents, Large Foundation Models, AI Agents, AI Assistants, Prompting, Pre-training, Fine-tuning.

*Both authors contributed equally to this research.

†Corresponding author: Wenqi Fan, Department of Computing, and Department of Management and Marketing, The Hong Kong Polytechnic University.

[1]The long version of this survey can be found at: *https://arxiv.org/abs/2503.23350*.

## 1 Introduction

As the web has rapidly evolved, it has profoundly transformed various aspects of people's lives, including information access [21, 47, 50], shopping experiences [9, 43], and communications [10, 82]. For instance, the web serves as the largest knowledge repository to date, offering instant access to news [41, 124], academic papers (e.g., ArXiv [17]), and encyclopedias (e.g., Wikipedia [96, 105]), enabling individuals to freely acquire desired information. This advancement has eliminated geographical barriers, providing people in remote areas with access to critical resources in education, healthcare, and law. Despite the importance of the web, many daily tasks we perform on it are repetitive and extremely time-consuming.
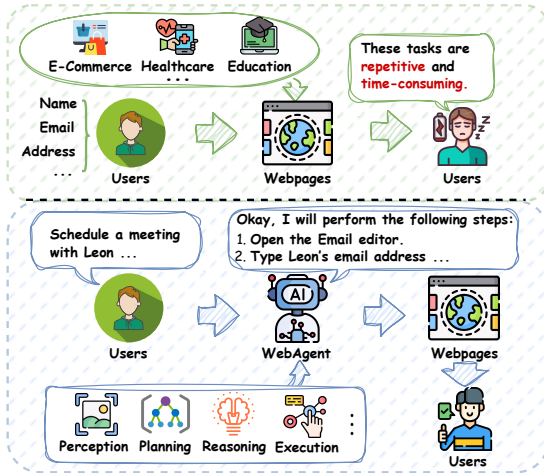
**Figure 1: Illustration of basic web tasks and the pipeline of WebAgents. Given the user instruction, WebAgents autonomously complete tasks by perceiving the environment, reasoning action sequences, and executing interactions.**

For example, as shown in Figure 1, when registering accounts on various platforms or filling out different application forms, we are often required to repeatedly enter the same personal information, such as our name, contact details, and address. Similarly, when purchasing a product, we need to compare numerous options, review their ratings and prices, and ultimately decide on the final purchase. To effectively execute tedious daily tasks, one of the most promising techniques is to develop automatic agents embedded with human intelligence by taking advantage of Artificial Intelligence (AI) techniques, known as **AI Agents**. In addition, AI Agents can execute tasks continuously without fatigue or performance degradation [39], ensuring reliability in repetitive workflows. Therefore, leveraging AI Agents – *termed **WebAgents** in the context of web* – to assist people in handling tedious daily tasks automatically can extremely enhance productivity and efficiency, thereby further improving their quality of life.

Recently, large foundation models (**LFMs**) with billions of parameters, trained on massive data, have exhibited emergent human-like capabilities such as comprehension and reasoning, revolutionizing various domains including healthcare [21, 29, 79], e-commerce [23, 74], and AI4Science [22, 33]. For example, LFMs are integrated with protein data to capture the foundational protein knowledge, enabling better understanding and generation of protein structures, which can significantly advance the development of drug discovery and disease mechanism research [22]. The human-like reasoning capabilities of LFMs are also leveraged in recommender systems (RecSys) to provide better item recommendations, significantly enhancing user online experience [107, 128]. By leveraging their extensive open-world knowledge, advanced instruction-following, and language comprehension and reasoning abilities, LFMs exhibit proficiency in simulating human-like behaviors to execute a variety of complex tasks. This naturally raises the promising topic: '*Can LFMs be utilized to develop powerful AI Agents that automatically handle web tasks, providing significant convenience to users?*'

To fully explore the potential of LFMs, recent efforts have been made to advance LFM-empowered **WebAgents** to complete various web tasks according to user instructions [39]. For instance, the recent debut of a novel AI Agent framework named *AutoGPT* has attracted significant interest from both academic and industrial communities, which exhibits impressive capabilities in autonomously handling complex tasks across both work and daily environments [87]. Unlike chatbots, AutoGPT can plan and execute complex tasks independently, performing automated searches and multi-step actions without requiring ongoing user instructions and supervision. In this context, as illustrated in Figure 1, users only need to provide a natural language instruction, such as '*Schedule a meeting with Leon at Starbucks on November 23, 2024, at 4:00 pm via email.*' WebAgents can autonomously open the 'Email' application, retrieve Leon's email address, compose the email, and send it, thereby automating the entire scheduling process and greatly enhancing the convenience of daily life. Given the remarkable progress in developing LFM-empowered WebAgents and the growing number of related studies, there is a pressing need for a systematic review of recent advances in this field.

To bridge this gap, this survey provides a comprehensive overview of WebAgents by summarizing representative methods from the perspectives of architecture, training, and trustworthiness. Specifically, in Section 2, we review existing studies based on the three processes of WebAgents: perception, planning & reasoning, and execution. Next, we summarize two crucial aspects (i.e., data and training strategies) in the training of WebAgents in Section 3. After that, we review studies that focus on investigating the trustworthy WebAgents, including their safety & robustness, privacy, and generalizability, in Section 4. Finally, in Section 5, we discuss promising future research directions in WebAgents.

## 2 WebAgent Architectures

There are three crucial and consecutive processes for WebAgents to fulfill user commands: **1) Perception** requires WebAgents to accurately observe the current environment, **2) Planning & Reasoning** require WebAgents to analyze the current environment, interpret user-given tasks, and predict reasonable next actions, and **3) Execution** requires that WebAgents perform the generated actions and interact with the environment effectively. In the following section, we will comprehensively review the important techniques employed by WebAgents during these processes.

### 2.1 Perception

Typical LFMs merely need to accept user instructions and generate corresponding responses through reasoning. However, WebAgents, operating within complex web environments, are further expected to accurately perceive the external environment and perform behavioral reasoning based on the dynamic environment combined with the user's task. As shown in Figure 2, according to the data modality provided by the environment to WebAgents, we can categorize existing studies into three classes: 1) **Text-based**, 2) **Screenshot-based**, and 3) **Multi-modal** WebAgents.

***2.1.1 Text-based WebAgents.*** With the advancement of large language models (LLMs), extensive studies have been proposed to leverage its human-like understanding and reasoning abilities to
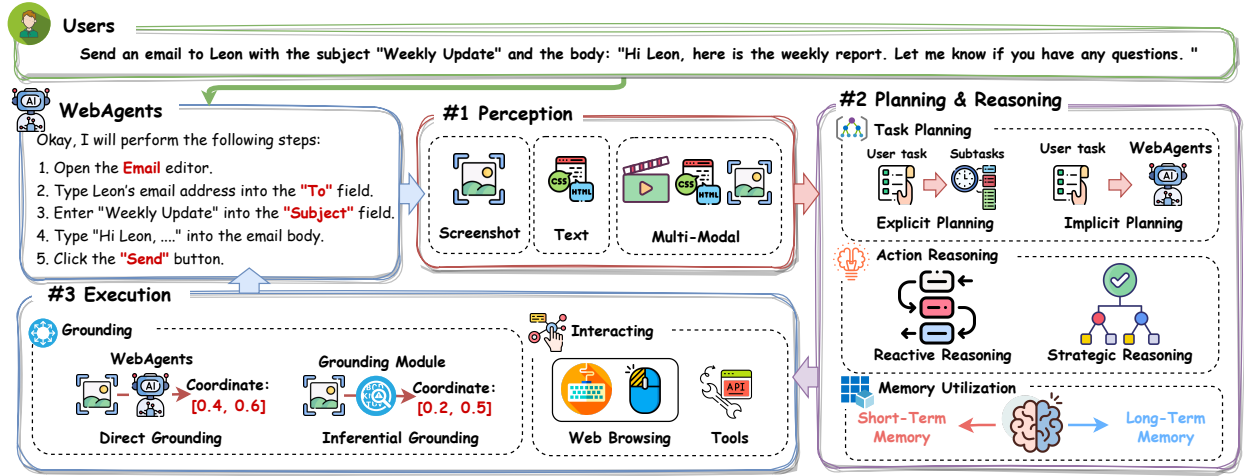
**Figure 2: Illustration of the overall framework of WebAgents, which contains three crucial processes: *Perception, Planning & Reasoning,* and *Execution.* Given the user's command, WebAgents first observe the environmental information during the perception process. Based on the observation, the action is generated in the planning & reasoning process. Finally, WebAgents execute the generated action to complete the user's task.**

assist users in addressing complex tasks. Since LLMs can only handle natural language, these WebAgents usually leverage the textual metadata of webpages (e.g., **HTML** and **accessibility trees**) to perceive the environment [45, 67, 69, 129]. For example, MindAct [18] introduces a two-stage framework that combines a fine-tuned small language model (LM) with an LLM to efficiently process large HTML documents, significantly reducing the input size while preserving essential information. This approach enables accurate prediction of both the target element and the corresponding action, effectively balancing efficiency and performance in web-based tasks. Gur et al. [35] introduce an LLM-driven agent that learns from self-experience to complete tasks on real-world webpages. It summarises long HTML documents into task-relevant snippets to extract the environmental information and decomposes user instructions into sub-tasks for effective planning.

***2.1.2 Screenshot-based WebAgents.*** Despite the remarkable success of text-based WebAgents, leveraging the textual metadata of the environment usually fails to align closely with human cognitive processes since the Graphical User Interfaces (GUI) are inherently visual [84, 116]. Additionally, textual representations usually vary across different environments and are verbose, leading to poor generalization abilities and increased computational overhead [127]. Recently, breakthroughs in large vision-language models (VLMs) have significantly enhanced the capabilities of AI systems in processing complex visual interfaces. To leverage the visual understanding capabilities of VLMs, numerous studies have integrated them into WebAgents, utilizing screenshots to perceive the environment [27, 31, 44, 127]. For example, SeeClick [15] only relies on screenshots as observations to predict the next action and enhances the agent's ability to locate relevant visual elements within screenshots by introducing a grounding pre-training process. OmniParser [68] introduces an effective method to parse user interface screenshots into structured elements and enhances GPT-4V's [119] ability to accurately ground actions to specific regions on the screen.

***2.1.3 Multi-modal WebAgents.*** In addition to solely utilizing textual metadata or screenshots to comprehend the environment, numerous studies also leverage multi-modal data, combining their complementary strengths to provide WebAgents with a more comprehensive environmental perception [42, 52, 99]. For instance, MMAC-Copilot [91] integrates GPT-4V for interpreting visual information from screenshots while leveraging Gemini Vision [59] to process and analyze video content, significantly enhancing the model's capabilities in handling multi-modal data. WebVoyager [37] is a multi-modal WebAgent that autonomously completes web tasks end-to-end by processing both screenshots and textual content from interactive web elements. It leverages Set-of-Mark Prompting [117] to overlay bounding boxes of the interactive elements on the webpages, significantly enhancing the agent's decision-making ability and enabling accurate action prediction and execution.

## 2.2 Planning & Reasoning

Subsequent to the perception of environmental information, WebAgents are generally tasked with determining the appropriate action to execute the user's command. This involves analyzing the current state of the environment and utilizing the reasoning capabilities of LFMs. As shown in Figure 2, there are three subtasks involved in this process: 1) **Task Planning**, which focuses on reorganizing the user's instruction and setting sub-objectives to help WebAgents effectively handle complex user queries; 2) **Action Reasoning**, which guides WebAgents to generate appropriate actions to fulfill the user's commands; and 3) **Memory Utilization**, which equips WebAgents with internal information (e.g., previous actions) or external information (e.g., open-world knowledge from web search) to predict more appropriate actions.

***2.2.1 Task Planning.*** In the context of WebAgents, the objective of task planning is to determine a sequence of steps that the agent should take to complete the user-defined task efficiently and effectively [112]. Based on whether WebAgents explicitly involve

a task decomposition process, existing studies can be categorized into two types: 1) **Explicit Planning** and 2) **Implicit Planning**. Explicit planning methods usually decompose user instructions into multiple subtasks and generate actions to complete them step by step [55, 92, 121]. For example, ScreenAgent [75] introduces a structured workflow where the agent decomposes user tasks into subtasks, describes the screenshot, and generates the next actions in a function-call format. It further incorporates a reflection phase, which guides the agent to decide whether to proceed, retry, or reformulate the plan based on the current progress, making the entire workflow aligned with human thought processes. OS-Copilot [110] introduces a versatile planner that decomposes complex user requests into manageable subtasks while retrieving relevant external information about the agent's capabilities (e.g., available tools and operating system information) to assist planning.

Implicit planning WebAgents directly feed the user instruction and the environmental observation into agents without an explicit task decomposition process [16, 49, 69]. For instance, WebWISE [94] directly provides the task message to the agent and prompts it to generate actions progressively based on filtered Document Object Model (DOM) elements as observations. OpenWebAgent [42] first uses a Web Processing Module to extract and simplify HTML elements, perform Optical Character Recognition (OCR) [88] on screenshots, and annotate interactive elements. Then, it introduces an Action Generation Module to predict the next action by directly feeding the current task and webpage context into agents.

***2.2.2 Action Reasoning.*** Action reasoning involves leveraging the agent's reasoning capabilities and current environmental observations to infer the next action. Based on the strategies, existing reasoning methods can be generally divided into two categories: 1) **Reactive Reasoning** and 2) **Strategic Reasoning**. Reactive reasoning means that WebAgents simply receive input prompts, including observations and instructions, and directly generate the next actions without additional operations [38, 93, 123, 126]. For instance, Agent-E [2] introduces a planner agent to generate the next action without using complicated mechanisms. After that, a browser navigation agent equipped with multiple predefined foundational skills is employed to observe and interact with the environment to accomplish the user task. ASH [67] first proposes a SUMMARIZER prompt to condense the full webpage into an action-aware observation. During the action reasoning process, ASH directly utilizes an ACTOR prompt to predict the next action based on the summarized observation that contains relevant elements.

Strategic reasoning usually introduces additional operations to enhance the agent's action reasoning capability. Two of the most common approaches include incorporating an additional **exploration process** and integrating extra **in-context information**, both of which can effectively improve the accuracy of the agent's action generation [3, 62]. For example, WebDreamer [30] proposes a novel LLM-driven exploration strategy to simulate and predict the outcomes of candidate actions before execution using natural language descriptions, allowing the agent to evaluate and select the optimal action at each step. This action simulation mechanism can significantly enhance decision accuracy and efficiency while reducing unnecessary interactions with the webpage. Auto-intent [46] introduces an unsupervised method to extract compact intents from multiple target domain demonstrations and trains an intent

predictor to forecast the next intent based on past observations and actions. By providing top-k predicted intents as additional in-context hints, Auto-intent enhances the agent's performance on large-scale real-website navigation benchmarks.

***2.2.3 Memory Utilization.*** Beyond task planning and action reasoning, the effective utilization of memory is another key factor contributing to the powerful capabilities of WebAgents. Depending on its source, memory can generally be divided into 1) **Short-term Memory** and 2) **Long-term Memory**. Short-term memory generally denotes the previous actions performed to accomplish the user's current task. Considering short-term memory when generating the next action can effectively avoid redundant operations and improve task completion efficiency [38, 52, 68, 81, 123]. For example, LLMPA [31] introduces a Previous Action Description Generator to generate action descriptions based on the high-level task description and the corresponding historical behavior sequences. Previous actions and their generated descriptions are injected into input prompts as the short-term memory for the next action prediction.

Long-term memory refers to external information that persists over time, such as action trajectories from previously executed tasks and knowledge acquired by online search. By retrieving this external knowledge as a reference, task success rates of WebAgents can be significantly improved [94, 101, 110, 121]. For instance, Agent S [3] leverages both Online Web Search for external knowledge and Narrative Memory for internal task-specific experience that includes summaries of both successful and failed trajectories to generate a sequence of subtasks that can accomplish the user instruction. After that, some similar subtask experience is also retrieved and used for the action generator to predict the next action. Synapse [129] first processes the raw web state into concise task-relevant observations to reduce the computational overhead and introduces a trajectory-as-exemplar (TaE) prompting strategy, which retrieves similar trajectories (i.e., sequences of abstracted states and actions) to guide the generation of the next action.

## 2.3 Execution

The final step for WebAgents to complete the user's command is to interact with the webpages and execute the generated actions. As shown in Figure 2, there are two tasks during this process: 1) **Grounding**, which aims to locate the elements within the webpage that agents will interact with, and 2) **Interacting**, which focuses on performing specific actions on the selected elements.

***2.3.1 Grounding.*** Since webpages often contain numerous interactive elements, selecting the correct element to execute the generated action is crucial for completing the user's task. Based on the grounding strategies of WebAgents, we categorize existing research into two categories: 1) **Direct Grounding** and 2) **Inferential Grounding**. Direct grounding refers to WebAgents directly generating the coordinates of the candidate element within a screenshot or selecting an element from the entire HTML for interactions [42, 52, 118, 123]. For example, ShowUI [62] directly guides the agent to generate the correct action (e.g., [CLICK]) and its corresponding parameters (e.g., coordinates for [CLICK]) to locate the elements within the webpage that the action will perform. Auto-intent [46] leverages the HTML elements as the in-context

example and guides the agents to correctly select the target element that the generated action will operate on.

Inferential grounding involves leveraging extra auxiliary modules to locate target elements [3, 38, 81, 83]. For instance, Ponder & Press [100] introduces a general-purpose multi-modal LLM as an interpreter, responsible for translating high-level user instructions into detailed action descriptions, and a GUI-specific multi-modal LM as a locator, which accurately identifies the target GUI elements within visual screenshots based on the generated action descriptions. OSCAR [99] introduces a dual-grounding observation approach that contains visual grounding and semantic grounding. Visual grounding uses Set-of-Mark Prompting with precise element bounding boxes extracted from the accessibility tree, and explicit semantic grounding annotates key elements with descriptive labels. This dual-grounding strategy significantly improves GUI understanding, enabling OSCAR to accurately locate and interact with GUI elements in complex and dynamic web environments. MindAct [18] introduces a two-stage framework that first uses a fine-tuned small LM to rank and filter webpage elements, followed by an LLM that processes the selected elements to predict both the target element and the required action.

***2.3.2 Interacting.*** Finally, WebAgents need to interact with the target element using the generated actions. By repeating the above steps multiple times, WebAgents can complete the user's complex task. Based on the approach WebAgents interact with the webpages, existing studies can be generally categorized into two classes: 1) **Web Browsing-based** and 2) **Tool-based** methods. Web browsing-based methods utilize typical actions that humans employ when navigating websites, such as clicking, scrolling, and typing [38, 62, 84, 100, 116]. For example, NaviQAte [83] defines three widely-used actions [Click], [Type], and [Select] to perform the web application navigation task. AgentOccam [118] simplifies the action space for WebAgents by removing redundant, low-utility, and embodiment-dependent actions, while introducing high-level commands like [Note], [Stop], and so on, significantly enhancing the interacting efficiency of WebAgents.

Tool-based methods involve using additional tools, such as application programming interfaces (APIs), to interact with the webpages [30, 71, 90, 110]. For instance, API-calling agent [90] introduces a novel framework that expands the traditional WebAgent action space by incorporating direct API interactions, enabling agents to bypass GUI-based interactions entirely, improving efficiency and adaptability for real-world online tasks. Infogent [30] proposes to utilize external tools like Google Search API to navigate the web and a scraper to extract webpage contents for web information aggregation tasks.

## 3 Training of WebAgents

There are two fundamental aspects in the training of WebAgents: 1) **Data** serves as the cornerstone of WebAgent training, providing diverse and representative samples that help models learn web-specific patterns. 2) **Training Strategies** indicate how WebAgents acquire and refine their capabilities through different approaches. The overall training framework of WebAgents is presented in Figure 3, including constructing training data and training strategies.

### 3.1 Data

Data serves as the foundation that powers contemporary WebAgents, with training data construction encompassing two essential steps: **1) Data Pre-processing**, which refines and structures the data to enhance its usability and quality, and **2) Data Augmentation**, which augments both the quantity and diversity of the dataset.

***3.1.1 Data Pre-processing.*** Web environments usually encompass multiple modalities, such as text and images, as well as diverse data formats across platforms. The multi-modal nature of web data offers rich information that facilitates a deeper understanding of the current environment. However, the discrepancies across different modalities and inconsistencies in data granularity also present significant challenges. To effectively utilize the multi-modal information, several studies propose various approaches for **modality alignment** [8, 78]. For instance, Liu et al. [64] capture critical web elements and layout structures through screenshots and augmented accessibility trees, enhancing rich interactions between text and visuals while filtering out irrelevant data within websites. Gou et al. [28] construct *<screenshot, referring expression, coordinates>* triplets for multi-modality alignment, where the referring expressions are generated based on HTML, incorporating visual, positional, and functional information. Moreover, web environments across different platforms usually manifest data discrepancies, such as naming conflicts, requiring robust **format alignment**. For example, the action *tap* on mobile devices corresponds to *click* on PCs, potentially confusing model consistency in understanding and executing tasks across platforms. To address this issue, OS-Atlas [111] aligns the action space in cross-platform datasets, mitigating potential inconsistencies introduced by heterogeneous data integration.

***3.1.2 Data Augmentation.*** Large-scale data serves as the foundation for the emergent intelligence of LFMs. For LFM-empowered WebAgents, it is also crucial to collect large volumes of diverse training data, thereby enhancing their capabilities to accurately perceive website environments, reason next action, and execute complex web-based operations. Depending on the methods of data acquisition, data augmentation approaches can be categorized into two distinct types: 1) **Data Collection** and 2) **Data Synthesis**.

Data collection involves gathering data from public datasets or directly from real-world scenarios. Rather than indiscriminately aggregating all available data [7], some studies like ShowUI [62] demonstrate that cautiously sampling high-quality, representative data from public datasets can significantly enhance WebAgents' performance. While data annotated by human experts exhibits high quality, its labor-intensive process limits the total amount of the resulting datasets, leading to undertrained models with limited generalization to unseen online contexts. To address this challenge, several autonomous generation methods have been proposed to construct cost-effective datasets without reliance on human assistance or VLMs [84, 85]. For instance, UINav [57] mitigates the pressure of scarce training data by randomizing the attributes of secondary UI elements, such as text embedding or element offsets, to augment demonstrations without additional data collection overhead.

Data synthesis involves automatically generating plausible web-relevant datasets using LLMs or VLMs to enrich training data when real-world samples are insufficient or costly to obtain. Beyond some basic data synthesis approaches that merely crawl elements from
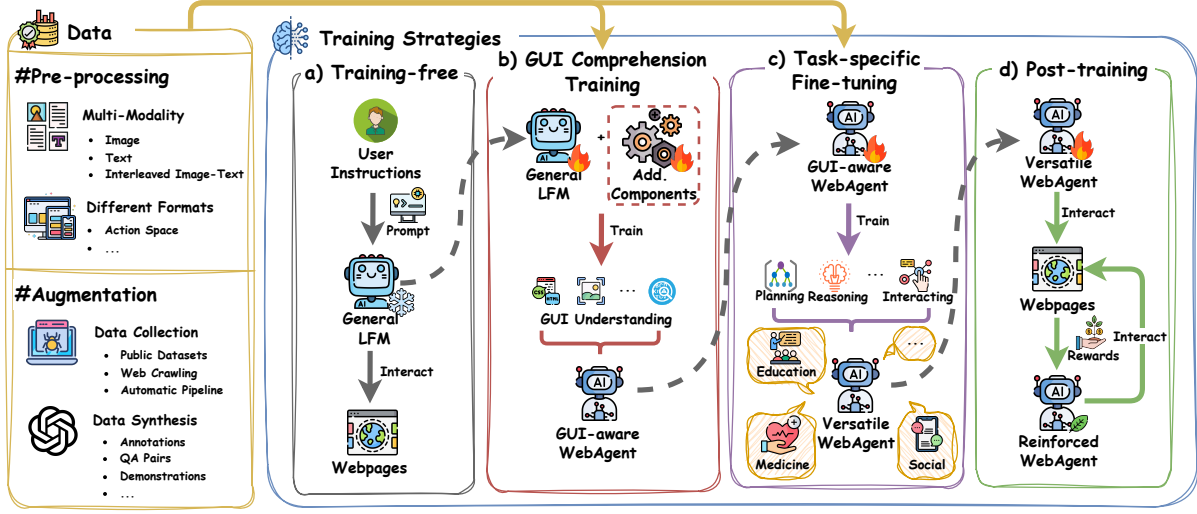
**Figure 3: Illustration of the training for WebAgents. There are two processes to construct the training data: 1) *Data Pre-processing* aims to reduce discrepancy across various data modalities and formats, and 2) *Data Augmentation* enhances the quantity and diversity of the training data. For training strategies, there are mainly four categories: 1) *Training-free* approach directly utilizes prompts to guide LFMs in completing web tasks, 2) *GUI Comprehension Training* aims to enhance GUI understanding capabilities of general-purpose LFMs, 3) *Task-specific Fine-tuning* improves task-oriented capabilities of WebAgents, and 4) *Post-training* interacts with webpages to receive rewards for further optimizing the policy of WebAgents.**

webpages and generate annotations for these elements, a group of studies focuses on creating rich question-answering (QA) pairs to further advance the GUI understanding of WebAgents, including general QA pairs [5, 13] and interaction-focused QA pairs [14, 60]. Complementing these QA-based approaches, some studies utilize VLMs to produce trajectory-related data that captures sequential interactions for explicit planning and reasoning [76, 116]. For example, AgentTrek [115] leverages collected web tutorials as step-by-step instructions to simulate execution trajectories. Moreover, NNet-Nav [73] generates substantial, realistic, and interactive trajectory data through a hierarchical exploration policy and pruning techniques to navigate websites within an exponentially large space.

## 3.2 Training Strategies

Once large datasets are collected, effectively using them for WebAgent training is essential. Depending on the training strategies employed, which differ in learning paradigms, data usage, and optimization objectives, existing studies can generally be divided into four categories: 1) **Training-free**, 2) **GUI Comprehension Training**, 3) **Task-specific Fine-tuning**, and 4) **Post-training**.

***3.2.1 Training-free.*** The rapid evolution of LFMs, endowed with human-like intelligence and robust comprehension of visual and textual information, substantially boosts the development of intelligent WebAgents. Building upon these capabilities, training-free methods directly adapt LFMs as specialized WebAgents, harnessing well-crafted prompts to guide models in executing web tasks without any architectural modifications or parameter updates [12, 80, 102, 103]. For example, CoAT [122] proposes a chain-of-action-thought prompting paradigm, which integrates actions and thoughts to enable more efficient navigation capabilities. The

paradigm incorporates screen descriptions, previous actions, and their current outcomes to offer explicit explanations behind the next action determination, alongside generating textual descriptions of the subsequent step and its potential results.

***3.2.2 GUI Comprehension Training.*** Although general-purpose LFMs are extensively trained on enormous datasets, these models often fall short in GUI-aware capabilities, particularly screen understanding and OCR, hindering them to comprehend and interact effectively with webpages. Specifically, general LFMs might focus on decorative icons or background text instead of the key interface elements, leading to misinterpretations of available functionalities [64]. To bridge this gap, numerous GUI comprehension training approaches are developed to enhance the critical foundational GUI understanding capabilities of WebAgents through supervised learning on large-scale web datasets [7, 20, 84, 109, 111]. For instance, Aguvis [116] introduces a two-stage training paradigm that begins with unifying GUI environments as images and concentrates on equipping the model to comprehend and interact with objects within a single GUI screenshot during the pre-training stage, laying a robust GUI comprehension foundation for subsequent fine-tuning. Moreover, given the context sensitivity of the UI elements, LVG [78] introduces layout-guided contrastive learning, which captures the semantics of individual UI elements based on their visual organization, further enhancing the model's GUI comprehension.

Beyond the aforementioned approaches that focus on introducing new training objectives or stages, several approaches also incorporate additional components during training to strengthen the GUI understanding capabilities of general-purpose LFMs. For example, Spotlight [54] implements Region Summarizer to extract

the most essential regions of screenshots based on VLM encoding, thereby facilitating GUI understanding. ScreenAI [5] employs Pix2Struct [53], a versatile patching strategy that maps visual elements to their corresponding HTML. This approach enables the model to adapt its image understanding capabilities to the specific context of UI-related screenshots.

**3.2.3 Task-specific Fine-tuning.** Although WebAgents can effectively understand environmental information through GUI comprehension training, accurately reasoning and generating the next steps for web interactions based on user tasks remains a significant challenge due to the complexity of web environments and the diversity of user objectives in web automation. Therefore, task-specific fine-tuning aims to equip WebAgents with web task-oriented skills, such as planning, reasoning, and interacting capabilities [26, 31, 73, 84, 116]. For instance, Gur et al. [34] propose an LLM-driven agent fine-tuned with scripted planning datasets, which decomposes natural language instructions into manageable sub-instructions, summarizes lengthy HTML documents into task-relevant snippets, and executes actions through self-generated Python programming. Besides, WebGUM [25] proposes a data-driven offline supervised fine-tuning process that adapts WebAgents for web navigation tasks based on instruction-following format problems and chain-of-thought examples across a variety of domains. The web navigation action output is in free-form text, enhancing the flexibility and adaptability to tackle real-world web navigation.

**3.2.4 Post-training.** Post-training allows WebAgents to continuously adapt and improve after supervised training when facing exponentially large and dynamic web environments. Given the open-domain nature of interactive web exploration, training methods relying solely on static datasets face clear limitations, making reinforcement learning a key post-training technique. As web interfaces constantly evolve and user requirements change, reinforcement learning enables WebAgents to adapt in real-time by exploring dynamic environments and learning from interaction feedback [24, 40, 52, 95]. For instance, AutoGLM [65] follows a progressive reinforcement learning framework that enables a continuous self-evolving learning paradigm through autonomous interactions with web environments. The knowledge it acquires from real websites is inherently dynamic, enabling the system to adapt and refine its decision-making capability in real-time.

## 4 Trustworthy WebAgents

Alongside the remarkable progress of WebAgents, a growing body of research and real-world incidents has highlighted the inherent risks and challenges associated with WebAgents. Recent studies have begun to reveal the potential threats posed by agents closely integrated with web systems. These threats range from unreliable and opaque decision-making in safety-critical contexts to the perpetuation of bias and unfairness towards marginalized communities [19, 58]. Moreover, concerns have been raised about the inadvertent exposure of user privacy and sensitive business information, further emphasizing the critical need to ensure the security of WebAgent technologies. Additionally, limitations in generalizability pose serious risks for WebAgents when confronted with diverse and unforeseen situations, potentially leading to critical failures in handling out-of-distribution data or operating across different domains. Consequently, developing advanced techniques for trustworthy WebAgents has become an area of increasing interest for researchers, primarily focusing on creating and deploying AI systems that prioritize **Safety & Robustness**, **Privacy**, and **Generalizability**.

**1) Safety & Robustness.** WebAgents need to be resilient to both noisy changes and adversarial attacks, which is essential due to the complexity and volatility of the real web environments in which they operate [32, 120]. Motivated by these concerns, AdvWeb [113] explores the vulnerabilities of VLM-based WebAgents to black-box attacks, such as the injection of adversarial prompts into webpages. These vulnerabilities could result in significant issues, including inappropriate stock purchases or erroneous bank transactions. Another instance of such threats is WIPI, a novel web threat that indirectly manipulates WebAgents to carry out malicious instructions embedded in publicly accessible webpages [106]. To provide a thorough investigation, several rigorous benchmarks have been presented. For example, ARE [104] includes 200 targeted adversarial tasks and assessment scripts within a realistic threat model using VisualWebArena [48], a real-world environment for WebAgents. Additionally, Kumar et al. [51] develop a comprehensive test suite called BrowserART, specifically designed for red-teaming browser agents. This suite comprises 100 diverse harmful behaviors related to browsers. Their findings indicate that refusal-trained models can be easily compromised when functioning as browser agents.

Aiming to address the safety concerns associated with WebAgents, advanced methods have been proposed. For example, Step [89] dynamically composes policies for web actions into a Markov Decision Process, ensuring that WebAgents effectively manage the handoff of control between different policies. Another common solution involves enabling WebAgents to converse with each other to successfully accomplish complex tasks [108]. This collaborative approach allows agents to verify the accuracy and appropriateness of actions before execution.

**2) Privacy.** WebAgents have demonstrated remarkable potential in autonomously completing a wide range of tasks on real websites, significantly boosting human productivity. However, web tasks, such as booking flights, often involve users' personal data, financial details, and proprietary business information, which may be exposed to potential privacy risks if WebAgents accidentally interact with compromised websites [125]. Protecting privacy is crucial to safeguarding the data used by WebAgents from breaches and unauthorized access. Building on this understanding, Wang et al. [97] have explored the privacy risks associated with LLM agents, particularly concerning their memory. Their research introduces a black-box attack known as the Memory EXTRaction Attack (MEXTRA), which examines the vulnerability of LLM agents in preventing the extraction of private information from memory across various scenarios, including web applications. Furthermore, Liao et al. [61] propose an Environmental Injection Attack (EIA) to explore two adversarial objectives within web environments: stealing specific personal information from users or capturing entire user requests. It involves injecting malicious content that is tailored to adapt effectively to the environments in which the agents operate.

**3) Generalizability.** The effectiveness of many WebAgents depends on the assumption that both their training and testing data originate from the same distribution. However, this assumption

often fails due to the out-of-distribution (OOD) issue, which occurs when the training and testing data come from different distributions [6, 56, 77]. This mismatch poses significant challenges for deploying WebAgents in critical situations, highlighting the need to ensure generalizability in trustworthy WebAgents. To address this challenge, Mind2Web [18] has been developed as the first dataset aimed at creating and evaluating generalist WebAgents. Mind2Web includes over 2,000 open-ended tasks gathered from 137 websites across 31 domains, along with crowd-sourced action sequences for these tasks. It offers three essential components for building generalist WebAgents: (i) a variety of domains, websites, and tasks, (ii) the use of real-world websites rather than simulated or simplified ones, and (iii) a wide range of user interaction patterns. Recently, a World-Model-Augmented (WMA) WebAgent has been introduced [11], representing a groundbreaking advancement in incorporating world models into LFM-empowered WebAgents. WMA lays the groundwork for policy adaptation by utilizing feedback from simulated environments during web navigation, and its LLM backbone is trained with a novel method known as transition-focused observation abstraction. Mazzaglia et al. [70] introduce an agent learning framework called GenRL, which connects and aligns the representation of foundation VLMs with the latent space of generative world models for reinforcement learning without any language annotations. This framework enables the specification of tasks through vision and/or language prompts, grounds these tasks in the dynamics of the embodied domain, and facilitates the learning of corresponding behaviors through imagination.

In addition to the three dimensions previously discussed, trustworthy WebAgents also encompass other important dimensions, such as Fairness and Explainability [63]. While these aspects are crucial, they are still in the early stages of development, with limited literature available, particularly concerning WebAgent techniques. Therefore, we will explore these dimensions as future research directions requiring dedicated efforts in the next section.

## 5 Future Directions

As research on WebAgents remains in its nascent stages, this section aims to discuss several promising research directions in the future. *1) Fairness and Explainability of WebAgents.* Existing research [4] primarily focuses on enhancing the capabilities of WebAgents, while their trustworthiness remains largely under-explored, particularly in terms of Fairness and Explainability [63, 98]. **Fairness** requires WebAgents to operate without bias in perception, reasoning, and execution. For instance, when users of different genders request WebAgents to search for suitable job positions, the agents should handle these requests fairly rather than relying on prior assumptions, such as suggesting that men are more suited to be lawyers while women are more suited to be nurses. **Explainability** requires that WebAgents be capable of justifying their actions, helping users understand their internal mechanisms, and ensuring their reliability in high-stakes environments, such as stock investment [1] and molecular design [66]. *2) Datasets and Benchmarks of WebAgents.* The systematic evaluation of WebAgents has emerged as a critical research focus, with several comprehensive benchmarks developed to rigorously assess their performance [86, 114, 132]. Despite the remarkable success,

most benchmarks focus on limited facets of functionality or particular use cases, often overlooking crucial aspects such as adapting to varied web layouts, resilience in the face of unforeseen errors, or proficiency in managing intricate tasks. Moreover, many existing assessments fail to adequately address real-world complexities, including fluctuating internet speeds, inconsistent website structures, and the need for sustained contextual reasoning across prolonged interactions. Therefore, there is an urgent need for more comprehensive and well-rounded benchmarks that can fully evaluate the wide-ranging capabilities of WebAgents.

*3) Personalized WebAgents.* While existing LFM-empowered WebAgents have demonstrated impressive capabilities in various web tasks, they still face significant limitations in personalization due to their massive parameter size (often in the billions) and expensive training costs. Therefore, developing WebAgents that simultaneously demonstrate robust capabilities while offering meaningful personalization represents a challenging yet promising research direction. To tackle this challenge, researchers have begun exploring innovative approaches, such as integrating Retrieval-Augmented Generation (RAG) systems [21] with memory mechanisms that encompass both long-term [36, 130] and short-term [72, 131] components. Specifically, the long-term memory supports WebAgents by retrieving and storing relevant information over extended periods, enabling WebAgents to maintain a consistent and personalized understanding of the user. In contrast, short-term memory allows WebAgents to quickly adapt to immediate contextual cues in real-time prompts, ensuring responsive and dynamic adjustments to the current conversation or task. By integrating these elements, personalized WebAgents can achieve higher adaptability, offering tailored solutions that better meet individual user needs.

## 6 Conclusion

Web tasks have become an integral part of people's daily lives, as individuals rely on the web for various activities. Despite the importance of the web, many daily web tasks remain repetitive and time-consuming. Recently, extensive efforts have been made to develop autonomous WebAgents capable of completing these complex web tasks based on user instructions. With the rapid advancements in WebAgents, there is an urgent need to review existing studies systematically. To address this gap, in this paper, we comprehensively summarize recent developments in WebAgents from three perspectives: architectures, training, and trustworthiness, providing researchers with a clear overview of this field. Furthermore, as research on WebAgents is still in its early stages, we discuss potential research directions to offer insights and inspiration to the research community.

## Acknowledgments

## References

[1] Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. 2024. Cooperation, competition, and maliciousness: LLM-stakeholders

interactive negotiation. *Advances in Neural Information Processing Systems* 37 (2024), 83548–83599.

[2] Tamer Abuelsaad, Deepak Akkil, Prasenjit Dey, Ashish Jagmohan, Aditya Vempaty, and Ravi Kokku. [n. d.]. Agent-E: From Autonomous Web Navigation to Foundational Design Principles in Agentic Systems. In *NeurIPS 2024 Workshop on Open-World Agents*.

[3] Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. [n. d.]. Agent S: An Open Agentic Framework that Uses Computers Like a Human. In *The Thirteenth International Conference on Learning Representations*.

[4] Amanda Aird, Paresha Farastu, Joshua Sun, Elena Stefancová, Cassidy All, Amy Voida, Nicholas Mattei, and Robin Burke. 2024. Dynamic fairness-aware recommendation through multi-agent social choice. *ACM Transactions on Recommender Systems* 3, 2 (2024), 1–35.

[5] 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* (2024).

[6] Hao Bai, Yifei Zhou, Jiayi Pan, Mert Cemri, Alane Suhr, Sergey Levine, and Aviral Kumar. 2025. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *Advances in Neural Information Processing Systems* 37 (2025), 12461–12495.

[7] Pratyay Banerjee, Shweti Mahajan, Kushal Arora, Chitta Baral, and Oriana Riva. 2023. Lexi: Self-supervised learning of the ui language. *arXiv preprint arXiv:2301.10165* (2023).

[8] 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* (2024).

[9] Yi Cai and Brenda J Cude. 2016. Online shopping. *Handbook of consumer finance research* (2016), 339–355.

[10] Marco Carbone, Kohei Honda, and Nobuko Yoshida. 2012. Structured communication-centered programming for web services. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 34, 2 (2012), 1–78.

[11] Hyungjoo Chae, Namyoung Kim, Kai Tzu-iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. 2024. Web agents with world models: Learning and leveraging environment dynamics in web navigation. In *The Thirteenth International Conference on Learning Representations*.

[12] Pei Chen, Boran Han, and Shuai Zhang. 2024. CoMM: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving. *arXiv preprint arXiv:2404.17729* (2024).

[13] Qi Chen, Dileepa Pitawela, Chongyang Zhao, Gengze Zhou, Hsiang-Ting Chen, and Qi Wu. 2024. WebVLN: Vision-and-language navigation on websites. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 1165–1173.

[14] Xuetian Chen, Hangcheng Li, Jiaqing Liang, Sihang Jiang, and Deqing Yang. 2024. EDGE: Enhanced grounded gui understanding with enriched multi-granularity synthetic data. *arXiv preprint arXiv:2410.19461* (2024).

[15] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. SeeClick: Harnessing GUI Grounding for Advanced Visual GUI Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 9313–9332.

[16] 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* (2024).

[17] Colin B Clement, Matthew Bierbaum, Kevin P O'Keeffe, and Alexander A Alemi. 2019. On the use of arxiv as a dataset. *arXiv preprint arXiv:1905.00075* (2019).

[18] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems* 36 (2023), 28091–28114.

[19] Zehang Deng, Yongjian Guo, Changzhou Han, Wanlun Ma, Junwu Xiong, Sheng Wen, and Yang Xiang. 2024. Ai agents under threat: A survey of key security challenges and future pathways. *Comput. Surveys* (2024).

[20] Nicolai Dorka, Janusz Marecki, and Ammar Anwar. 2024. Training a vision language model as smartphone assistant. *arXiv preprint arXiv:2404.08755* (2024).

[21] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6491–6501.

[22] Wenqi Fan, Yi Zhou, Shijie Wang, Yuyao Yan, Hui Liu, Qian Zhao, Le Song, and Qing Li. 2025. Computational Protein Science in the Era of Large Language Models (LLMs). *arXiv preprint arXiv:2501.10282* (2025).

[23] Chenhao Fang, Xiaohan Li, Zezhong Fan, Jianpeng Xu, Kaushiki Nag, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2024. Llm-ensemble: Optimal large language model ensemble method for e-commerce product attribute value extraction. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2910–2914.

[24] Moghis Fereidouni, Adib Mosharrof, and AB Siddique. 2024. Grounded Language Agent for Product Search via Intelligent Web Interactions. *arXiv preprint*

[25] *arXiv:2404.10887* (2024).

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* (2023).

[26] Hiroki Furuta, Yutaka Matsuo, Aleksandra Faust, and Izzeddin Gur. 2023. Exposing limitations of language model agents in sequential-task compositions on the web. *arXiv preprint arXiv:2311.18751* (2023).

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

[28] 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* (2024).

[29] Sagar Goyal, Eti Rastogi, Sree Prasanna Rajagopal, Dong Yuan, Fen Zhao, Jai Chintagunta, Gautam Naik, and Jeff Ward. 2024. Healai: A healthcare llm for effective medical documentation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 1167–1168.

[30] 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* (2024).

[31] Yanchu Guan, Dong Wang, Zhixuan Chu, Shiyu Wang, Feiyue Ni, Ruihua Song, Longfei Li, Jinjie Gu, and Chenyi Zhuang. 2023. Intelligent Virtual Assistants with LLM-based Process Automation. *CoRR* (2023).

[32] Chengquan Guo, Xun Liu, Chulin Xie, Andy Zhou, Yi Zeng, Zinan Lin, Dawn Song, and Bo Li. 2025. RedCode: Risky Code Execution and Generation Benchmark for Code Agents. *Advances in Neural Information Processing Systems* 37 (2025), 106190–106236.

[33] Kehan Guo, Bozhao Nan, Yujun Zhou, Taicheng Guo, Zhichun Guo, Mihir Surve, Zhenwen Liang, Nitesh Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Can llms solve molecule puzzles? a multimodal benchmark for molecular structure elucidation. *Advances in Neural Information Processing Systems* 37 (2024), 134721–134746.

[34] 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* (2023).

[35] Izzeddin Gur, Hiroki Furuta, Austin V Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2024. A Real-World WebAgent with Planning, Long Context Understanding, and Program Synthesis. In *ICLR*.

[36] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. *arXiv preprint arXiv:2502.14802* (2025).

[37] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 6864–6890.

[38] 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* (2024).

[39] Xueyu Hu, Tao Xiong, Biao Yi, Zishu Wei, Ruixuan Xiao, Yurun Chen, Jiasheng Ye, Meiling Tao, Xiangxin Zhou, Ziyu Zhao, et al. 2024. Os agents: A survey on mllm-based agents for general computing devices use.

[40] Peter C Humphreys, David Raposo, Tobias Pohlen, Gregory Thornton, Rachita Chhaparia, Alistair Muldal, Josh Abramson, Petko Georgiev, Adam Santoro, and Timothy Lillicrap. 2022. A data-driven approach for learning to control computers. In *International Conference on Machine Learning*. PMLR, 9466–9482.

[41] José Antonio Iglesias, Alexandra Tiemblo, Agapito Ledezma, and Araceli Sanchis. 2016. Web news mining in an evolving framework. *Information Fusion* 28 (2016), 90–98.

[42] 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)*. 72–81.

[43] Mohamed Khalifa and Moez Limayem. 2003. Drivers of Internet shopping. *Commun. ACM* 46, 12 (2003), 233–239.

[44] 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*. 14445–14454.

[45] Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. *Advances in Neural Information Processing Systems* 36 (2023), 39648–39677.

[46] Jaekyeom Kim, Dong-Ki Kim, Lajanugen Logeswaran, Sungryull Sohn, and Honglak Lee. 2024. Auto-Intent: Automated Intent Discovery and Self-Exploration for Large Language Model Web Agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 16531–16541.

[47] Mei Kobayashi and Koichi Takeda. 2000. Information retrieval on the web. *ACM computing surveys (CSUR)* 32, 2 (2000), 144–173.

[48] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024. VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 881–905.

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

[50] Manish Kumar, Rajesh Bhatia, and Dhavleesh Rattan. 2017. A survey of Web crawlers for information retrieval. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7, 6 (2017), e1218.

[51] Priyanshu Kumar, Elaine Lau, Saranya Vijayakumar, Tu Trinh, Scale Red Team, Elaine Chang, Vaughn Robinson, Sean Hendryx, Shuyan Zhou, Matt Fredrikson, et al. 2024. Refusal-trained llms are easily jailbroken as browser agents. *arXiv preprint arXiv:2410.13886* (2024).

[52] 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*. 5295–5306.

[53] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*. PMLR, 18893–18912.

[54] Gang Li and Yang Li. 2022. Spotlight: Mobile ui understanding using vision-language models with a focus. *arXiv preprint arXiv:2209.14927* (2022).

[55] Tao Li, Gang Li, Zhiwei Deng, Bryan Wang, and Yang Li. 2023. A Zero-Shot Language Agent for Computer Control with Structured Reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 11261–11274.

[56] Wei Li, William E Bishop, Alice Li, Christopher Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2025. On the effects of data scale on ui control agents. *Advances in Neural Information Processing Systems* 37 (2025), 92130–92154.

[57] Wei Li, Fu-Lin Hsu, Will Bishop, Folawiyo Campbell-Ajala, Max Lin, and Oriana Riva. 2023. UINav: A practical approach to train on-device automation agents. *arXiv preprint arXiv:2312.10170* (2023).

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

[59] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. 2024. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814* (2024).

[60] Zhangheng Li, Keen You, Haotian Zhang, Di Feng, Harsh Agrawal, Xiujun Li, Mohana Prasad Sathya Moorthy, Jeff Nichols, Yinfei Yang, and Zhe Gan. 2024. Ferret-UI 2: Mastering universal user interface understanding across platforms. *arXiv preprint arXiv:2410.18967* (2024).

[61] 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. *arXiv preprint arXiv:2409.11295* (2024).

[62] Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. 2024. ShowUI: One vision-language-action model for gui visual agent. *arXiv preprint arXiv:2411.17465* (2024).

[63] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Yunhao Liu, Anil Jain, and Jiliang Tang. 2022. Trustworthy ai: A computational perspective. *ACM Transactions on Intelligent Systems and Technology* 14, 1 (2022), 1–59.

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

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

[66] Yuyan Liu, Sirui Ding, Sheng Zhou, Wenqi Fan, and Qiaoyu Tan. 2024. Moleculargpt: Open large language model (llm) for few-shot molecular property prediction. *arXiv preprint arXiv:2406.12950* (2024).

[67] Robert Lo, Abishek Sridhar, Frank F Xu, Hao Zhu, and Shuyan Zhou. 2023. Hierarchical prompting assists large language model on web navigation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 10217–10244.

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

[69] Kaixin Ma, Hongming Zhang, Hongwei Wang, Xiaoman Pan, and Dong Yu. [n. d.]. LASER: LLM Agent with State-Space Exploration for Web Navigation. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

[70] Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron C Courville, and Sai Rajeswar Mudumba. 2025. GenRL: Multimodal-foundation world models for generalization in embodied agents. *Advances in Neural Information Processing Systems* 37 (2025), 27529–27555.

[71] Kai Mei, Xi Zhu, Wujiang Xu, Wenyue Hua, Mingyu Jin, Zelong Li, Shuyuan Xu, Ruosong Ye, Yingqiang Ge, and Yongfeng Zhang. 2024. Aios: Llm agent operating system. *arXiv preprint arXiv:2403.16971* (2024).

[72] Manisha Mukherjee, Sungchul Kim, Xiang Chen, Dan Luo, Tong Yu, and Tung Mai. 2025. From Documents to Dialogue: Building KG-RAG Enhanced AI Assistants. arXiv:2502.15237 [cs.IR] https://arxiv.org/abs/2502.15237

[73] Shikhar Murty, Hao Zhu, Dzmitry Bahdanau, and Christopher D. Manning. 2024. NNetNav: Unsupervised Learning of Browser Agents Through Environment Interaction in the Wild. *arXiv preprint arXiv:2410.02907* (2024).

[74] Guangtao Nie, Rong Zhi, Xiaofan Yan, Yufan Du, Xiangyang Zhang, Jianwei Chen, Mi Zhou, Hongshen Chen, Tianhao Li, Ziguang Cheng, et al. 2024. A hybrid multi-agent conversational recommender system with llm and search engine in e-commerce. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 745–747.

[75] 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. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. 6433–6441.

[76] Tianyue Ou, Frank F Xu, Aman Madaan, Jiarui Liu, Robert Lo, Abishek Sridhar, Sudipta Sengupta, Dan Roth, Graham Neubig, and Shuyan Zhou. 2025. Synatra: Turning indirect knowledge into direct demonstrations for digital agents at scale. *Advances in Neural Information Processing Systems* 37 (2025), 91618–91652.

[77] 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* (2024).

[78] Yijun Qian, Yujie Lu, Alexander G Hauptmann, and Oriana Riva. 2024. Visual grounding for user interfaces. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*. 97–107.

[79] Jianing Qiu, Kyle Lam, Guohao Li, Amish Acharya, Tien Yin Wong, Ara Darzi, Wu Yuan, and Eric J Topol. 2024. LLM-based agentic systems in medicine and healthcare. *Nature Machine Intelligence* 6, 12 (2024), 1418–1420.

[80] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2023. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems* 36 (2023), 59708–59728.

[81] 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* (2024).

[82] Antonio Ruiz-Martínez. 2012. A survey on solutions and main free tools for privacy enhancing Web communications. *Journal of network and computer applications* 35, 5 (2012), 1473–1492.

[83] Mobina Shahbandeh, Parsa Alian, Noor Nashid, and Ali Mesbah. 2024. Naviqate: Functionality-guided web application navigation. *arXiv preprint arXiv:2409.10741* (2024).

[84] Huawen Shen, Chang Liu, Gengluo Li, Xinlong Wang, Yu Zhou, Can Ma, and Xiangyang Ji. 2024. Falcon-UI: Understanding GUI Before Following User Instructions. *arXiv preprint arXiv:2412.09362* (2024).

[85] Junhong Shen, Atishay Jain, Zedian Xiao, Ishan Amlekar, Mouad Hadji, Aaron Podolny, and Ameet Talwalkar. 2024. ScribeAgent: Towards Specialized Web Agents Using Production-Scale Workflow Data. *arXiv preprint arXiv:2411.15004* (2024).

[86] Segev Shlomov, Aviad Sela, Ido Levy, Liane Galanti, Roy Abitbol, et al. 2024. From grounding to planning: Benchmarking bottlenecks in web agents. *arXiv preprint arXiv:2409.01927* (2024).

[87] Significant Gravitas. [n. d.]. *AutoGPT*. https://agpt.co/

[88] Amarjot Singh, Ketan Bacchuwar, and Akshay Bhasin. 2012. A survey of OCR applications. *International Journal of Machine Learning and Computing* 2, 3 (2012), 314.

[89] Paloma Sodhi, SRK Branavan, Yoav Artzi, and Ryan McDonald. 2023. Step: Stacked llm policies for web actions. *arXiv preprint arXiv:2310.03720* (2023).

[90] Yueqi Song, Frank Xu, Shuyan Zhou, and Graham Neubig. 2024. Beyond Browsing: API-Based Web Agents. *arXiv preprint arXiv:2410.16464* (2024).

[91] Zirui Song, Yaohang Li, Meng Fang, Zhenhao Chen, Zecheng Shi, Yuan Huang, and Ling Chen. 2024. MMAC-Copilot: Multi-modal Agent Collaboration Operating System Copilot. *CoRR* (2024).

[92] Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Gang Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, et al. [n. d.]. Cradle:

Empowering Foundation Agents towards General Computer Control. In *NeurIPS 2024 Workshop on Open-World Agents*.

[93] Brian Tang and Kang G Shin. 2024. Steward: Natural language web automation. *arXiv preprint arXiv:2409.15441* (2024).

[94] Heyi Tao, TV Sethuraman, Michal Shlapentokh-Rothman, and Derek Hoiem. 2023. WebWISE: Web Interface Control and Sequential Exploration with Large Language Models. *CoRR* (2023).

[95] 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*. 866–874.

[96] Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. 2006. Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web*. 585–594.

[97] Bo Wang, Weiyi He, Pengfei He, Shenglai Zeng, Zhen Xiang, Yue Xing, and Jiliang Tang. 2025. Unveiling Privacy Risks in LLM Agent Memory. *arXiv preprint arXiv:2502.13172* (2025).

[98] Han Wang, An Zhang, Nguyen Duy Tai, Jun Sun, Tat-Seng Chua, et al. 2024. ALI-Agent: Assessing LLMs' Alignment with Human Values via Agent-based Evaluation. *Advances in Neural Information Processing Systems* 37 (2024), 99040–99088.

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

[100] Yiqin Wang, Haoji Zhang, Jingqi Tian, and Yansong Tang. 2024. Ponder & press: Advancing visual gui agent towards general computer control. *arXiv preprint arXiv:2412.01268* (2024).

[101] Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2024. Agent workflow memory. *arXiv preprint arXiv:2409.07429* (2024).

[102] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[103] Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. 2023. Empowering llm to use smartphone for intelligent task automation. *arXiv preprint arXiv:2308.15272* 4 (2023).

[104] Chen Henry Wu, Rishi Rajesh Shah, Jing Yu Koh, Russ Salakhutdinov, Daniel Fried, and Aditi Raghunathan. [n.d.]. Dissecting Adversarial Robustness of Multimodal LM Agents. In *The Thirteenth International Conference on Learning Representations*.

[105] Fei Wu and Daniel S Weld. 2007. Autonomously semantifying wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 41–50.

[106] Fangzhou Wu, Shutong Wu, Yulong Cao, and Chaowei Xiao. 2024. Wipi: A new web threat for llm-driven web agents. *arXiv preprint arXiv:2402.16965* (2024).

[107] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2024. A survey on large language models for recommendation. *World Wide Web* 27, 5 (2024), 60.

[108] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155* (2023).

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

[110] Zhiyong Wu, Chengcheng Han, Zichen Ding, Zhenmin Weng, Zhoumianze Liu, Shunyu Yao, Tao Yu, and Lingpeng Kong. 2024. OS-Copilot: Towards Generalist Computer Agents with Self-Improvement. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

[111] Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. 2024. OS-Atlas: A foundation action model for generalist GUI agents. *arXiv preprint arXiv:2410.23218* (2024).

[112] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences* 68, 2 (2025), 121101.

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

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

[115] Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. 2024. AgentTrek: Agent Trajectory Synthesis via Guiding Replay with Web Tutorials. *arXiv preprint arXiv:2412.09605* (2024).

[116] Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2024. Aguvis: Unified Pure Vision Agents for Autonomous GUI Interaction. *arXiv preprint arXiv:2412.04454* (2024).

[117] 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. *arXiv preprint arXiv:2310.11441* (2023).

[118] Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. 2024. AgentOccam: A Simple Yet Strong Baseline for LLM-Based Web Agents. In *The Thirteenth International Conference on Learning Representations*.

[119] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421* 9, 1 (2023), 1.

[120] Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, et al. 2024. R-judge: Benchmarking safety risk awareness for llm agents. *arXiv preprint arXiv:2401.10019* (2024).

[121] Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei Lin, Saravan Rajmohan, et al. 2024. UFO: A UI-Focused Agent for Windows OS Interaction. *CoRR* (2024).

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

[123] Shaoqing Zhang, Zhuosheng Zhang, Kehai Chen, Xinbei Ma, Muyun Yang, Tiejun Zhao, and Min Zhang. 2024. Dynamic Planning for LLM-based Graphical User Interface Automation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 1304–1320.

[124] Xichen Zhang and Ali A Ghorbani. 2020. An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management* 57, 2 (2020), 102025.

[125] Xinyu Zhang, Huiyu Xu, Zhongjie Ba, Zhibo Wang, Yuan Hong, Jian Liu, Zhan Qin, and Kui Ren. 2024. Privacyasst: Safeguarding user privacy in tool-using large language model agents. *IEEE Transactions on Dependable and Secure Computing* (2024).

[126] Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. 2024. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. *arXiv preprint arXiv:2408.15978* (2024).

[127] Zhuosheng Zhang and Aston Zhang. 2024. You Only Look at Screens: Multimodal Chain-of-Action Agents. In *Findings of the Association for Computational Linguistics ACL 2024*. 3132–3149.

[128] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. 2024. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering* (2024).

[129] Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. [n.d.]. Synapse: Trajectory-as-Exemplar Prompting with Memory for Computer Control. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

[130] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19724–19731.

[131] Xuanyan Zhong, Haiyang Xin, Wenfeng Li, Zehui Zhan, and May-hung Cheng. 2024. The Design and application of RAG-based conversational agents for collaborative problem solving. In *Proceedings of the 2024 9th International Conference on Distance Education and Learning*. 62–68.

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