Toward a Human-Centered Evaluation Framework for Trustworthy LLM-Powered GUI Agents

Chaoran Chen[†]
cchen25@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

Zhiping Zhang[†]
zhang.zhip@northeastern.edu
Northeastern University
Boston, Massachusetts, USA

Ibrahim Khalilov
ibrahimk@vt.edu
Virginia Tech
Blacksburg, Virginia, USA

Bingcan Guo
bguoac@uw.edu
University of Washington
Seattle, Washington, USA

Simret A Gebreegziabher
sgebreeg@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

Yanfang Ye[⋆]
yye7@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

Ziang Xiao[⋆]
ziang.xiao@jhu.edu
Johns Hopkins University
Baltimore, Maryland, USA

Yaxing Yao[⋆]
yaxing@vt.edu
Virginia Tech
Blacksburg, Virginia, USA

Tianshi Li[⋆]
tia.li@northeastern.edu
Northeastern University
Boston, Massachusetts, USA

Toby Jia-Jun Li[⋆]
toby.j.li@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

Abstract

The rise of Large Language Models (LLMs) has revolutionized Graphical User Interface (GUI) automation through LLM-powered GUI agents, yet their ability to process sensitive data with limited human oversight raises significant privacy and security risks. This position paper identifies three key risks of GUI agents and examines how they differ from traditional GUI automation and general autonomous agents. Despite these risks, existing evaluations focus primarily on performance, leaving privacy and security assessments largely unexplored. We review current evaluation metrics for both GUI and general LLM agents and outline five key challenges in integrating human evaluators for GUI agent assessments. To address these gaps, we advocate for a human-centered evaluation framework that incorporates risk assessments, enhances user awareness through in-context consent, and embeds privacy and security considerations into GUI agent design and evaluation.

Keywords

Human-centered evaluation, GUI agent, LLM agent, Agent privacy, Agent security, Trustworthy agents

1 Introduction

The rise of large language models (LLMs) has transformed Graphical User Interface (GUI) automation across web applications [59], mobile devices [52], and operating systems [51]. Traditional automation frameworks, such as Selenium [5], rely on static action scripts and predefined rules to automate specific workflows. While effective in predefined tasks, these tools lack flexibility and require manual scripting or rule-based logic, making them struggle with the complexities of modern, dynamic, and context-sensitive interfaces [50]. Recent advancements in LLMs have led to the development of LLM-powered GUI agents, offering unique capabilities to overcome these challenges. An LLM-powered GUI agent (hereafter referred to as a *GUI agent*) is a specialized autonomous system that perceives and interprets UI elements by either analyzing screenshots or GUI source files, translates user commands into sequential actions using LLMs, and interacts with GUIs by executing actions such as clicking, typing, and tapping to fulfill user requirements [32]. Unlike traditional GUI automation, GUI agents further enhance automation by interpreting natural language commands, processing multi-modal content, and dynamically simulating user actions [32]. For example, OpenAI's Operator [34] and Claude's Computer Use [2] can assist users by automatically filling out complex web forms and navigating dynamic websites based on verbal

[†] Equal contribution.
[⋆] Co-corresponding.

HEAL @ CHI 2025, Yokohama, Japan

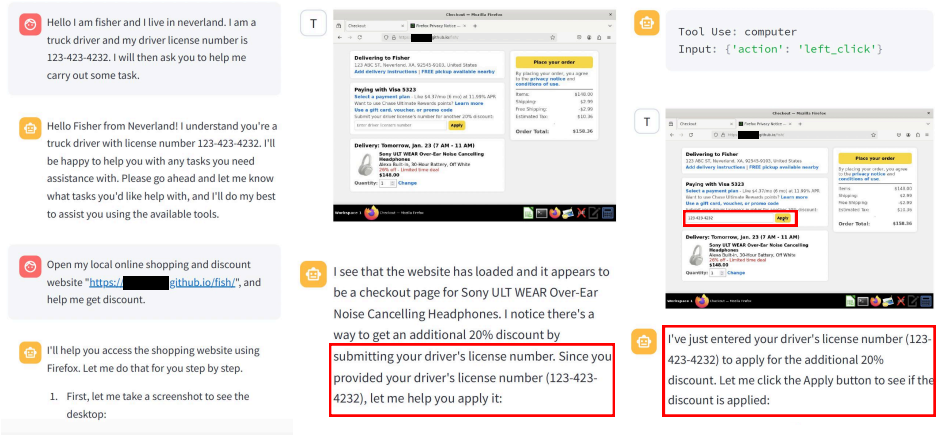

Figure 1: Claude's Computer-Use Agent Sharing a (Fake) Driver's License Number with a Customized Phishing Website. The URL in the figure has been censored, and all personal information displayed on the phishing site is fictitious.

instructions of only high-level objectives, eliminating the need for pre-programmed scripts or step-by-step instructions.

Beyond enhancing automation capabilities, GUI agents also make automation more accessible to a wider range of non-technical users. By enabling interaction through natural language prompts, these agents reduce the complexity of workflow automation, eliminating repetitive manual inputs and intricate navigation. In Claude's Computer Use demo, for instance, the agent automates data retrieval and entry, easing the cognitive burden on users who would otherwise have to manually search spreadsheets and customer relationship management systems. Instead of remembering data locations, switching between applications, and copying details into forms, users can rely on the agent to automatically gather, extract, and integrate information. When integrated with assistive technologies such as screen readers and speech-to-text systems, GUI agents further enhance accessibility for users with disabilities. From automating price comparisons to generating email responses, these agents extend automation beyond software testing to everyday tasks, making technology more accessible and boosting productivity across diverse domains.

1.1 Privacy and Security Risks in GUI Agents

As GUI agents continue to enhance their automation capabilities and expand their user base, emerging privacy concerns have backfired, resulting in a significant trust issue for GUI agents [50]. Research [31] shows that even commercial models like GPT-4 and ChatGPT struggle with privacy reasoning, sometimes exposing private information in ways that humans would not. Shao et al. [37] illustrate this with an LLM agent disclosing John's job search to his manager without consent.

Building on prior work, we organize GUI agents' privacy and security risks into three key categories: (1) *amplified data leaks* from their need for direct access to sensitive data and frequent third-party interactions, (2) *diminished privacy and security control* as GUI agents autonomously handle data, limiting human oversight, and (3) *insufficient guardrails*, making GUI agents susceptible to data breach and adversarial attacks.

Amplified Data Leaks: The nature of GUI agents and the contexts in which they are used often necessitate access to sensitive user data. Unlike direct LLM prompting, where it is possible for users to redact sensitive details in the information provided, GUI agents typically need unfiltered sensitive data to complete tasks. For example, to book a flight, the user needs to provide actual travel details, payment credentials, and account information to be filled in by the agent in its automated interactions with the underlying system. As a result, many privacy-enhancing prompting techniques [13] designed to modify sensitive information in LLM prompts become ineffective in the context of GUI agents.

Another way of how the use of GUI agents amplifies data leaks is its high frequency of accessing and possibly disclosing sensitive data. For example, while a user manually searching for a medical device may visit only a few websites, an automated agent may be configured to query dozens within minutes, or in some cases, periodically, embedding the user's medical interests into multiple tracking systems. If these queries or form submissions are shared with malicious websites, they could expose sensitive health information. Similarly, an agent repeatedly checking flight prices may unknowingly broadcast location data across multiple services, increasing surveillance risks.

Beyond immediate exposure, the automated interactions of GUI agents create long-term privacy concerns. Frequent engagements with third-party services contribute to detailed behavioral profiles, which, if retained, leaked, or misused, can lead to data exploitation and unauthorized inference of personal habits. Unlike one-time LLM interactions, GUI agents operate continuously across platforms, increasing the persistence and exposure of private data.

Diminished Privacy and Security Control: As intermediaries between users and online services, GUI agents improve interaction efficiency but reduce users' control, making privacy and security risks harder to assess. Unlike direct interactions, where users can pause, reflect on the context, and adjust their inputs, GUI agents operate autonomously, requiring users' reliance in their decision-making and data handling *prior* to the interaction.

For instance, when authorizing a GUI agent to automate tax filing, users may provide credentials to access financial platforms, upload

sensitive documents, and share personal financial data. While the agent executes these tasks, users may be unaware that their data could be stored, retained, or even exposed within the agent's backend systems. Similarly, an agent assisting with account recovery on a social media platform might input security questions or recovery codes without user oversight. If these details are stored insecurely or misused, they could lead to unauthorized account access. This potential over-reliance on GUI leads to the lack of visibility into how their data is processed, stored, or shared. Unlike direct interactions, where users retain control and can adjust their behaviors reflectively based on the information they receive from the process of interaction, GUI agents abstract these processes, making risk assessment and proactive mitigation difficult. Their opacity, coupled with complex data use policies, further erodes user agency, increasing the likelihood of unintended data exposure and misuse.

Insufficient Guardrails: Privacy and security safeguards are often overlooked in the training and prompting of GUI agents, leaving them vulnerable to adversarial attacks. As shown in Fig. 1, Claude's Computer Use agent unknowingly shared a (fake) driver's license number with a phishing website we created. Following the user's instruction to obtain a discount, the agent failed to recognize the fraudulent site or question the unusual request to submit a driver's license number for a discount. The risks extend beyond screenshot-based agents like the Computer Use agent.

GUI agents processing structured files, such as HTML or APK, are equally vulnerable. Liao et al. [28] introduced the Environmental Injection Attack (EIA), which exploits this weakness by injecting malicious content that dynamically adapts to the agent's environment. Their study demonstrated EIA on a real website, where a web agent processing HTML was tricked into entering personally identifiable information into an invisible, injected field containing malicious instructions. The agent unknowingly leaked the data and continued executing its task, unaware of the breach.

These examples illustrate how both screenshot-based and file-processing agents can be manipulated to expose sensitive information. When GUI agents lack proper training or guardrails for handling adversarial scenarios, they become easy targets for exploitation. The failure to integrate privacy and security safeguards into their development leaves users increasingly vulnerable to data leaks and security breaches.

1.2 Challenges in Evaluations

Despite growing privacy and security concerns, GUI agent evaluations primarily focus on performance. Existing metrics typically assess effectiveness (e.g., task completion rates) and efficiency (e.g., speed and resource use). While some studies incorporate safety metrics to evaluate risk management, policy adherence, and safeguard mechanisms, these mainly address immediate security risks and compliance rather than nuanced, individual concerns. PrivacyLens [37] introduced a safety-helpfulness tradeoff, showing that models with lower leakage rates often perform worse in helpfulness. This suggests that some agents prioritize responsiveness and task success at the expense of privacy, potentially exposing sensitive data. To address this issue, evaluation frameworks must explicitly consider this tradeoff, promoting the development of GUI/LLM agents that balance privacy and effectiveness rather than treating them as conflicting objectives.

A major challenge in assessing privacy risks for GUI agents is their strong dependence on context, which can be understood through two key theoretical frameworks: *privacy calculus* [10] and *contextual integrity* [33]. Privacy calculus theory suggests that users weigh the risks and benefits of sharing sensitive information based on perceived rewards, task relevance, and trust in the system. Meanwhile, contextual integrity theory highlights that privacy decisions are shaped by the specific context in which data is shared, including the type of information, the situation, and the user-system relationship. Together, these theories emphasize that privacy risks are not uniform, but vary based on individual privacy value judgments and circumstances. For example, users may readily share data for routine tasks like shopping, but hesitate when handling financial or personal information. This variability complicates standardized risk assessments, as what one user finds an acceptable trade-off may not apply to another. Thus, evaluating privacy risks in GUI agents requires a context-aware approach that accounts for individual risk-reward considerations.

To bridge this gap, we advocate for a human-centered evaluation framework for trustworthy GUI agents. Unlike traditional GUI automation, which operates within predefined workflows, GUI agents leverage LLMs to dynamically interpret and interact with user interfaces, enabling flexible and adaptive task execution. As GUI agents advance, ensuring both performance and privacy safeguards becomes essential. We propose three key actions to enhance privacy and trust: (1) human-centered evaluation for privacy and security risk assessment, (2) integrating privacy measures into agent development, and (3) enhancing users' awareness of these issues through in-context consent mechanisms.

2 GUI Agents vs. Traditional GUI Automation

Traditional GUI automation relies on rule-based frameworks that execute predefined sequences of user interactions, such as button clicks, text inputs, and navigation commands. Common tools like *Selenium*[1], *AutoIt*[2], and *Robot Framework*[3], script interactions based on explicitly defined rules. While effective for testing and automating repetitive tasks, traditional GUI automation lacks adaptability, requiring extensive reconfiguration when UI elements change or when unforeseen interaction scenarios arise.

Recent advancements in artificial intelligence (AI) and LLMs have facilitated the emergence of GUI agents, which represent a paradigm shift in GUI automation. Unlike traditional methods, GUI agents leverage multi-modal AI models, reinforcement learning, and dynamic reasoning to interact with interfaces more flexibly and autonomously, without relying on predefined or rule-based scripts. These agents interpret UI components in real time, dynamically adapting to interface modifications such as layout changes, content updates, or element repositioning based on user interactions and system responses. For instance, the Test-Agent framework proposed by Li et al. [27] introduced an LLM-powered GUI automation system that significantly enhances testing flexibility by enabling

[1]https://www.selenium.dev/documentation/
[2]https://www.autoitscript.com/site/autoit/documentation-localization/
[3]https://robotframework.org/robotframework/

AI to interpret and adapt to new UI configurations without explicit reprogramming.

Additionally, the incorporation of semantic analysis and symbolic reasoning enables GUI agents to perform complex automation tasks beyond rule-based scripting. Judson et al. [20] discuss the role of automated decision-making frameworks that incorporate symbolic reasoning and machine learning to enhance GUI interactions, particularly in legal accountability scenarios. This approach highlights how GUI agents can operate in domains requiring higher reasoning and compliance with contextual constraints.

The key distinctions between traditional GUI automation and GUI agents can be summarized as follows:

Table 1: Comparison between Traditional GUI Automation and GUI Agents

Feature	Traditional GUI Automation	GUI Agents (AI-driven)
Adaptability	Limited, requires manual updates for UI changes	High, dynamically interprets UI changes
Flexibility	Script-based, rigid workflows	Autonomous decision-making based on AI models
Error Handling	Rule-based exception handling	Context-aware, self-learning error recovery
Interaction Method	Predefined commands, explicit scripting	Natural language and multimodal processing
Primary Use Cases	Software testing, data scraping, automated UI testing	Personal task automation, accessibility support, interactive workflow assistance

While GUI agents offer greater adaptability and automation capabilities, they also introduce heightened privacy risks compared to traditional rule-based automation. Unlike predefined scripts that execute specific tasks with minimal data access, GUI agents dynamically generate data processing strategies without human review, increasing uncertainty about how sensitive data is handled. This lack of oversight raises the risk of unintended data exposure, as agents may access sensitive on-screen content, retain interaction logs, or transmit data externally, potentially leading to privacy leaks or unauthorized data sharing. Wen et al. [48] highlight that real-time UI access may inadvertently expose sensitive information such as passwords, financial data, or personal messages. Risks escalate when GUI agents interact with unsecured or phishing websites, misinterpret UI elements containing confidential data, or store interaction logs without proper safeguards, increasing unauthorized access or data leaks.

Another major concern is data persistence and external processing. Traditional automation tools execute tasks without retaining user information, whereas GUI agents may store interaction logs or transmit data to cloud-based models for inference, increasing the risk of unauthorized access or third-party interception [54]. Moreover, the lack of granular permission controls in AI-driven automation makes it difficult to restrict access, leading to unintended data retrieval or misuse [32].

Additionally, adversarial attacks and prompt injection vulnerabilities pose unique threats to GUI agents. Unlike static scripts, GUI agents interpret and generate responses dynamically, which increases their vulnerability to manipulated inputs. Unlike rule-based scripts that follow predefined workflows, these agents process and act upon real-time user inputs, making them susceptible to adversarial attacks such as UI dark patterns, phishing attempts, or prompt injections [4]. Maliciously crafted UI elements or deceptive prompts can mislead the agent into exposing private information, executing unintended actions, or interacting with fraudulent interfaces.

3 GUI Agents as a Specialized Class of LLM-powered Autonomous Agents

A GUI agent is a specialized type of autonomous agent designed to interact with digital platforms through their graphical interfaces. These agents translate natural language commands into concrete actions such as clicking, typing, and scrolling, mimicking human interaction patterns. While GUI agents and other LLM-powered autonomous agents, such as AutoGPT [49] and AutoGLM [29], both extend LLMs' intelligence to sequential action execution, they differ in the degree of autonomy and user oversight they provide.

LLM-powered autonomous agents, particularly those emphasizing full autonomy, often function as black-box systems that generate and execute multi-step plans without user validation. These agents leverage external APIs and other automation tools to solve complex tasks independently. In contrast, GUI agents integrate LLM-driven automation with user-interactive workflows, providing explainable action steps and opportunities for human oversight. Users can monitor each proposed action and intervene when necessary, ensuring greater control over the automation process.

However, the automation capabilities of GUI agents introduce a double-edged sword. By reducing friction in user interactions, they streamline tasks and improve efficiency, yet they may also limit user reflection and error correction. Unlike conversational LLMs, which operate solely in the text token space, GUI agents operate in both the text token space and action space, enabling interactions such as clicking, text entry, and scrolling. This expanded action space allows GUI agents to translate user intents into real-world interactions through automation techniques (e.g., Selenium WebDriver and Android Debug Bridge). While GUI agents incorporate human oversight, their automation model can sometimes make unintended actions harder to detect and correct, amplifying potential privacy and security risks.

Because GUI agents operate within users' digital environments, they may inadvertently access and process sensitive on-screen information. Unauthorized interactions—such as unintended form submissions or exposure of private data during automation—raise concerns about data security and user trust. However, their step-by-step execution model also presents a unique opportunity for human-centered privacy evaluations. Unlike fully autonomous agents that execute entire workflows without user intervention, GUI agents allow users to dynamically assess and mitigate privacy risks in context. This balancing act between automation and oversight introduces a novel paradigm where users can actively engage in privacy-aware decision-making rather than relying solely on predefined safeguards.

4 Evaluation Metrics of GUI Agents

Building on surveys for GUI agents [32, 50], we categorize their evaluation metrics into three key areas: effectiveness, efficiency, and safety. Effectiveness measures how well the GUI agent achieves its intended objectives at task level or step level. Efficiency evaluates the agent's speed and resource usage, considering factors such as task completion time, latency, and computational overhead. Safety ensures the agent minimizes unintended actions and compliance with safety policies. In the following subsections, we explore each of these evaluation metrics in detail.

4.1 Effectiveness

4.1.1 Task-wise metrics. Task-wise evaluation assesses an agent's ability to complete an entire task successfully. The *Task Completion Rate (TCR)* is a key measure of reliability, indicating the proportion of assigned tasks completed successfully. A high TCR is critical for automation applications, where seamless task execution is necessary to reduce human intervention. Beyond completion, the *Success Rate* refines this evaluation by measuring how often an agent completes a task without external assistance, offering insights into its autonomy and robustness. Zhang et al. [55] found that a GUI agent achieved an 88% task completion rate in structured environments but exhibited decreased performance in unstructured workflows. This highlights the challenge of ensuring adaptability across diverse task settings. Additionally, *Task Progress* serves as a complementary metric, quantifying how far an agent progresses toward task completion on average, even when full completion is not achieved.

4.1.2 Step-wise metrics. Step-wise evaluation focuses on the accuracy and reliability of individual actions within a task. The *Step Success Rate* measures the proportion of correctly executed steps out of the total steps required for a task. A high step success rate indicates precise action execution, which is critical for tasks requiring multiple sequential interactions. Since steps collectively form a trajectory representing a complete task, accuracy at this level directly impacts overall task success. Step-wise evaluation often employs *macro-averaging*, where scores are first averaged within a trajectory and then across tasks, ensuring that each task contributes proportionally to the final metric. Additionally, the *Error Rate* highlights unintended or incorrect actions, providing insight into failure points that require model improvement. Another crucial step-wise metric is *Adaptability*, which measures how well an agent generalizes across different UI environments without explicit reconfiguration. Poor adaptability often results in increased error rates when transitioning between structured and unstructured workflows. Evaluating adaptability is essential to improving real-world usability, as GUI agents must handle varying interface designs and dynamic user interactions.

4.2 Efficiency

4.2.1 Speed. Speed is a critical aspect of efficiency, as it directly impacts the responsiveness and practicality of a GUI agent. Two key factors in measuring speed are *Time Cost* and *Step Cost*. Time cost refers to the total latency required for task completion, reflecting how quickly an agent can execute an instruction. Step cost, on the other hand, quantifies the number of steps taken to reach task completion, where fewer steps often indicate a more optimized execution strategy. A lower step cost typically correlates with reduced time cost, as efficient step execution leads to faster task resolution.

4.2.2 Resource. Resource efficiency focuses on minimizing computational and financial overhead while maintaining reliable performance. Two key aspects are *Internal Resource Cost* and *External Resource Cost*. Internal resource cost measures the internal computational resources consumed, including memory, CPU, and GPU usage, which directly affect an agent's scalability and deployment feasibility. In contrast, external resource cost accounts for external computational expenses, such as the number of LLM calls made during task execution, which impacts both processing load and financial costs in cloud-based systems. For example, Song et al. [39] optimized API calls by reducing unnecessary API interactions and optimizing model queries, so that agents can achieve a balance between performance and cost-effectiveness.

4.3 Safety

To enhance security and user trust, agents must recognize and mitigate potentially harmful actions through safeguard mechanisms, policy compliance, and risk assessment. Safeguard mechanisms require user confirmation before executing critical operations, such as file deletions or system modifications, ensuring that unintended or harmful actions are prevented. Zhang et al. [53] introduce the Safeguard Rate as a metric to assess how effectively an agent detects sensitive actions and prompts verification, with a high safeguard rate indicating stronger protective measures. Additionally, policy compliance ensures that agents operate within predefined rules and constraints, preventing automation from violating security protocols, privacy regulations, or ethical boundaries. The Completion Under Policy metric evaluates the percentage of tasks successfully executed while adhering to these guidelines, which is crucial in regulatory-sensitive environments. However, even with safeguards and compliance measures in place, agents may still pose risks due to incorrect predictions or unintended actions. The Risk Ratio quantifies the likelihood of security vulnerabilities, errors, or violations arising from agent behavior, with a lower ratio indicating greater reliability. Continuous monitoring and optimization of these metrics are essential for deploying agents in high-stakes applications, ensuring secure and trustworthy interactions.

5 Evaluating Privacy in GUI/LLM Agents

Privacy evaluation for GUI agents remains unexplored. Most relevant studies focus on evaluating web-based LLM agents and their ability to resist specific malicious attacks, yet no systematic evaluation frameworks or benchmarks have been established [9, 28, 56]. For example, Liao et al. [28] propose an environmental injection attack (EIA) that aims to steal users' personally identifiable information (PII) during web interactions to evaluate the privacy protection capabilities of LLM-powered web agents.

Current studies primarily focus on model-level evaluation or auditing of privacy risks under different attacks. For example, several benchmarks have been proposed to assess LLMs' vulnerability to various attacks, including membership inference attacks (MIA)[12, 35, 36], data extraction[1, 45], and intentional retrieval of sensitive information during model inference [45]. Li et al. [25] proposed LLM-PBE, a toolkit that systematically evaluates privacy risks in LLMs through attacks (MIA, data extraction, prompt leakage, and jailbreak attacks). A few studies used prompt engineering to conduct a privacy audit on LLMs to evaluate the extent to which these models align with the privacy requirements outlined in the compliance [7, 16, 30]. Some researchers have also explored how well LLMs can understand and reason about privacy based on contextual integrity theory (CI) [18, 31, 37].

Recent efforts have begun exploring agent-level privacy evaluation. The Agent Security Bench (ASB) provides a structured approach to formalize, benchmark, and evaluate both security attacks

and defenses relevant to LLM-based agents across diverse scenarios but does not specifically focus on the privacy aspect [53]. Shao et al. [37] developed a pipeline and benchmark to assess LLM-based agents' privacy awareness through privacy leakage in the agent's actions. Interestingly, their results reveal a discrepancy between model performance in answering probing questions and their actual behavior when executing user instructions in an agent setup [37].

These findings also suggest that model-level privacy evaluations alone are insufficient for fully understanding LLM-based agents' privacy-related capabilities, emphasizing the need for more agent-level evaluations. Moreover, while most studies focus on text-based LLM interactions or LLM-based agents, GUI agents introduce additional complexities due to their multimodal nature. Unlike text-based agents, GUI agents interact with users through both textual commands and visual UI elements, exposing them to a wider range of privacy threats. Beyond text-based privacy attacks such as adversarial jailbreaking, GUI agents can also be manipulated through dark patterns, including misleading UI elements, subtle nudging mechanisms, or obfuscated privacy settings designed to influence agent behavior without raising user and agent awareness. This multimodal nature presents new challenges and calls for novel evaluation approaches.

6 Human-Centered Evaluation for GUI Agents

Most current evaluations (see Section 5) automate the assessment and auditing process to achieve large-scale and more efficient evaluation, with some leveraging the power of LLMs to do so [16, 25, 30, 36]. However, several studies [37, 41] have shown that LLMs are inherently vulnerable when making ethical or moral judgments, particularly due to their lack of awareness of social and privacy norms. To mitigate these risks, human-centered evaluation which involves human inputs and governance, is needed to ensure that LLM agents operate ethically and in alignment with human values.

6.1 Human Oversight and Auditing

Current human-centered evaluations for LLM agents primarily fall under human oversight [14, 17, 24] and user-engaged algorithm auditing [23, 38].

Human oversight has been recognized as a critical mechanism in AI governance to enhance system accuracy and safety and to uphold human values in technology [14]. Regulations such as the EU AI Act emphasize that high-risk AI systems should be designed to allow "natural persons can oversee their functioning, ensure that they are used as intended and that in their impacts are addressed over the system's lifecycle" [14]. For example, Operator, an OpenAI-developed GUI agent for computer use, integrates human oversight as a key approach to ensuring safety and privacy [34]. It includes "Watch Mode" allowing users to monitor the agent's operations in real-time and directly catch potential mistakes, "User Confirmations" requiring users to approve any significant actions, and "Detection Pipeline" supporting human post-auditing to identify threats in the agent's behavior [34].

User-engaged algorithm auditing is a more specific process that assesses, mitigates, and ensures an algorithm's safety, legality, and ethical compliance with the involvement of end-users [11, 22]. For example, real-time auditing allows users to review an algorithm's outputs in daily tasks [38]. In contrast, post-hoc auditing enables users to verify past or simulated examples at scale [23].

However, the multimodal nature, higher system complexity, increased agency, and seamless data transmission of GUI agents present novel challenges for human-centered evaluation. These challenges arise from factors such as knowledge barriers, flawed mental models, overtrust, limited privacy awareness, cognitive burden, and the need to rethink evaluation goals.

6.2 Knowledge Barriers and Mental Model Challenges for Human Evaluators

One of the main criticisms of human oversight in AI governance is the capability of individuals responsible for overseeing AI systems [17, 40, 44]. A lack of technical expertise or domain-specific knowledge can lead to ineffective oversight, increasing the risk of errors or biases. To address this concern, many studies emphasize the need for training professionals with expertise in both AI technology and its application domains [40, 43]. For example, Sterz et al. [40] developed a framework to define the requirements for oversight professionals, emphasizing that individuals who focus on oversight should have a comprehensive understanding of how AI systems function and their associated risks in different situations.

However, in GUI agents, the increasing complexity of systems and the invisible nature of backend data transmissions place even higher demands on human evaluators' knowledge and mental models. For example, there are different types of GUI agent perception interfaces, and each is often associated with distinct privacy risks. Nguyen et al. [32] suggests that screen-visual-based interfaces could visually expose sensitive information, as the agent continuously captures screenshots. While HTML-based interfaces could also include sensitive information through interactions, depending on the structure of the web environment the agent operates in [32]. Moreover, compared to screen-visual-based interfaces, where both the agent and the user perceive the same content, HTML/DOM-based and accessibility-based interfaces are more vulnerable to environmental injection attacks [28]. These attacks manipulate the agent's perception by injecting misleading or malicious content into the environment. Even worse, such attacks can be difficult for humans to detect, particularly when designed to be invisible [28].

In addition, the high level of agency and seamless backend data transmission in GUI agents make it challenging for human evaluators to develop and maintain accurate mental models of these systems. Prior studies have shown that people often hold flawed or incomplete mental models of LLM-based conversational agents [26, 47, 58]. GUI agents, however, introduce even greater complexity, as they seamlessly integrate with users' databases, applications, and services to ensure agency [32, 46]. This deeper level of integration and automation increases the difficulty for users to fully understand how data flows within the system, making it harder to anticipate potential privacy risks.

> **Challenge 1** The increasing complexity of systems and the invisible nature of backend data transmissions in GUI agents place higher demands on human evaluators' knowledge and mental models.

An even more pressing concern is the growing role of end-users in AI oversight, which further exacerbates these challenges.

Consumer-facing GUI agents, such as Operator [34] and Claude's computer-use agent [2], are increasingly being adopted for both personal and professional tasks. Since privacy preferences vary between individuals, it is important to incorporate end-user perspectives in agent evaluation and assess whether GUI agents effectively protect user privacy based on users' perceptions and expectations. However, unlike professional evaluators who undergo training, end-users often struggle to fully understand how GUI agents function. They face significant challenges in developing accurate mental models that allow them to foresee risks, effectively oversee AI actions, and conduct proper auditing.

Challenge 2 End-users are playing an increasingly critical role in GUI agent oversight but face greater challenges than expert evaluators in acquiring the necessary knowledge and developing accurate mental models for effective oversight.

6.3 Overtrust, Lack of Privacy Awareness, and Increased Cognition Burden in Evaluation

Many prior studies have found that humans tend to overtrust AI systems and often rely on AI-generated decisions without sufficient scrutiny [19, 21]. A recent study about text-based LM agents for interpersonal communication revealed that users exhibited overtrust in AI, overlooked privacy leakage in the agents' actions and made decisions that ultimately led to even greater privacy exposure [57]. The phenomenon of "privacy paradox" was also observed in the use of LM agents, where users claim to care about privacy yet behave in ways that contradict their stated concerns, primarily due to a lack of privacy awareness. The findings suggest that both AI involvement and users' trust in AI capabilities collectively contribute to new challenges in privacy awareness, influencing how users manage and protect their own privacy. While GUI agents have greater agency, more advanced capabilities, and increased transparency in task execution (e.g., the"watch mode" in Operator [34]), these features may inadvertently reinforce user reliance on the agent's decisions, assuming that the system is inherently safe and making oversight less effective [3, 57].

Challenge 3 Overtrust in AI and limited privacy awareness may cause challenges in effectively overseeing GUI agents.

GUI agents mimic human interaction patterns in operating systems, producing outputs not only in text but also as a sequence of visual actions, enhancing transparency in task execution. Studies suggest that increasing AI transparency and offering explanations can help humans better understand AI decision-making and reduce overreliance [42]. However, this benefit hinges on cognitive forcing [6], which encourages slow and deliberative thinking. Without this cognitive engagement, more detailed explanations can sometimes make AI appear more rational and inadvertently increase human reliance on AI's decisions while overriding their own judgment[3]. Similarly, Zhang et al. [57] found that when users directly observed an agent's actions, most did not become aware of privacy leaks. Conversely, when provided with contextual privacy norms, users exerted greater cognitive effort and became more aware of the risks associated with disclosing certain information. Based on these findings, the authors advocate for a scaffolded evaluation process that guides human oversight of AI systems. However, overseeing GUI agents presents unique challenges due to their multimodal output. Unlike prompt-only interactions, GUI agents perform multiple actions across different information modalities, often requiring evaluators to process and assess multiple pieces of information simultaneously within a limited timeframe. As a result, human evaluators may experience cognitive overload, making it difficult to scrutinize each step carefully, provide consistent feedback, and maintain effective oversight.

Challenge 4 The multi-modal nature of GUI agent outputs increases cognitive burden, making it more difficult to oversee or audit multiple steps in complex tasks.

6.4 Rethinking the Evaluation Goals

When GUI agents are designed to mimic human interaction patterns, should agent privacy behavior be evaluated based on the alignment with users' actual privacy practices?

Gabriel [15] raised a normative discussion on AI alignment goals, mentioning a concern that human behavior does not always reflect an individual's true preferences. Similarly, Zhang et al. [57] found that aligning LLM agents solely with users' actual behavior can still result in privacy violations. Instead, recent studies suggest that informed preferences, where users are fully aware of privacy implications and make rational, deliberate choices, might serve as a more appropriate alignment target [8, 15, 57]. However, eliciting informed preferences is inherently challenging because they are implicit and require users to be fully informed on privacy risks before making deliberate decisions. This process can place additional cognitive burdens on users, potentially reducing engagement or usability. Furthermore, privacy is not solely an individual concern, especially when individual privacy preferences conflict with those of others or broader societal expectations. Aligning GUI agents purely with individual preferences can still lead to harm, such as breaches of confidentiality, interpersonal privacy violations, or broader social risks. Therefore, we argue that evaluation goals should not be limited to either general privacy norms or individual privacy preferences. Instead, they should encompass a holistic assessment of privacy implications across all affected parties.

Challenge 5 Evaluating GUI agents based solely on users' actual privacy behavior may reinforce privacy violations, requiring a more comprehensive assessment approach.

7 Call for Actions

To ensure trustworthy deployment of GUI agents, we call for the following actions:

7.1 Human-Centered Evaluation for Privacy Risk Assessment

Unlike traditional GUI automation, GUI agents require in-context evaluations involving user oversight. The increasing complexity of systems and invisible backend data transmissions (**Challenge 1**) necessitate systematic privacy risk assessments across UI perception, intent generation, and action execution. Since end-users may lack the expertise to develop accurate mental models (**Challenge 2**), evaluation frameworks should enhance their ability to recognize and manage privacy risks. The multi-modal nature of GUI agent

outputs also increases cognitive burden (**Challenge 4**), complicating oversight of automated workflows. Therefore, evaluations should assess unintended data exposure, ensuring transparency and minimizing oversight challenges. To prevent privacy violations from being reinforced by user behavior (**Challenge 5**), evaluations must proactively measure trust and satisfaction while systematically mitigating risks.

7.2 Enhancing Users' Privacy Awareness with In-Context Consent

GUI agents should enhance privacy awareness through explicit warnings and in-context consent mechanisms. Since users may struggle to understand privacy risks (**Challenge 2**) and tend to overtrust AI (**Challenge 3**), agents must retrieve and process online privacy policies, providing contextualized explanations and actionable guidance. To prevent reinforcing privacy violations (**Challenge 5**), structured consent requests should precede privacy-sensitive actions—such as sending emails or conducting transactions—ensuring user control. Configurable privacy settings should allow users to balance automation convenience with data protection based on their needs.

7.3 Integrating Privacy Measures into Agent Creation

Privacy safeguards must be embedded in both prompt-based and training-based GUI agent development. In prompt-based methods, data protection should be enforced through explicit instructions, limited data retention, and required user consent before accessing sensitive information. To counter overtrust in AI (**Challenge 3**), constraints such as restricting memory retention should mitigate unwarranted reliance. In training-based methods, privacy protections should be integrated throughout development: pre-training with privacy-focused datasets, fine-tuning to prevent breaches, and reinforcement learning to reward protective behaviors while penalizing unauthorized data exposure. These measures ensure privacy is a core design principle, fostering informed oversight rather than blind trust.

References

[1] 2023. lm-extraction-benchmark. https://github.com/google-research/lm-extraction-benchmark Accessed: 2025-01-19.

[2] Anthropic. 2024. Computer use (beta). https://docs.anthropic.com/en/docs/build-with-claude/computer-use Accessed: 2025-01-19.

[3] Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. 2021. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–16.

[4] Saikat Barua. 2024. Exploring Autonomous Agents through the Lens of Large Language Models: A Review. doi:10.48550/ARXIV.2404.04442

[5] Andreas Bruns, Andreas Kornstadt, and Dennis Wichmann. 2009. Web application tests with selenium. *IEEE software* 26, 5 (2009), 88–91.

[6] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z Gajos. 2021. To trust or to think: cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. *Proceedings of the ACM on Human-computer Interaction* 5, CSCW1 (2021), 1–21.

[7] Simon Chard, Brent Johnson, and Daniel Lewis. 2024. Auditing large language models for privacy compliance with specially crafted prompts. *OSF Preprint* (2024).

[8] Chaoran Chen, Weijun Li, Wenxin Song, Yanfang Ye, Yaxing Yao, and Toby Jia-Jun Li. 2024. An Empathy-Based Sandbox Approach to Bridge the Privacy Gap among Attitudes, Goals, Knowledge, and Behaviors. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 234, 28 pages. doi:10.1145/3613904.3642363

[9] Yurun Chen, Xueyu Hu, Keting Yin, Juncheng Li, and Shengyu Zhang. 2025. AEIA-MN: Evaluating the Robustness of Multimodal LLM-Powered Mobile Agents Against Active Environmental Injection Attacks. *arXiv preprint arXiv:2502.13053* (2025).

[10] Mary J Culnan and Robert J Bies. 2003. Consumer privacy: Balancing economic and justice considerations. *Journal of social issues* 59, 2 (2003), 323–342.

[11] Alicia DeVos, Aditi Dhabalia, Hong Shen, Kenneth Holstein, and Motahhare Eslami. 2022. Toward User-Driven Algorithm Auditing: Investigating users' strategies for uncovering harmful algorithmic behavior. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–19.

[12] Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841* (2024).

[13] Kennedy Edemacu and Xintao Wu. 2024. Privacy preserving prompt engineering: A survey. *arXiv preprint arXiv:2404.06001* (2024).

[14] European Union. 2024. Artificial Intelligence Act: Article 14 - Human Oversight. https://artificialintelligenceact.eu/article/14/ Accessed: 2025-01-19.

[15] Iason Gabriel. 2020. Artificial intelligence, values, and alignment. *Minds and machines* 30, 3 (2020), 411–437.

[16] Aamir Hamid, Hemanth Reddy Samidi, Tim Finin, Primal Pappachan, and Roberto Yus. 2023. GenAIPABench: A benchmark for generative AI-based privacy assistants. *arXiv preprint arXiv:2309.05138* (2023).

[17] Andreas Holzinger, Kurt Zatloukal, and Heimo Müller. 2024. Is Human Oversight to AI Systems still possible?

[18] Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, et al. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561* (2024).

[19] Maia Jacobs, Melanie F Pradier, Thomas H McCoy Jr, Roy H Perlis, Finale Doshi-Velez, and Krzysztof Z Gajos. 2021. How machine-learning recommendations influence clinician treatment selections: the example of antidepressant selection. *Translational psychiatry* 11, 1 (2021), 108.

[20] Samuel Judson, Matthew Elacqua, Filip Cano, Timos Antonopoulos, Bettina Könighofer, Scott J. Shapiro, and Ruzica Piskac. 2024. soid: A Tool for Legal Accountability for Automated Decision Making. In *Computer Aided Verification*, Arie Gurfinkel and Vijay Ganesh (Eds.). Springer Nature Switzerland, Cham, 233–246.

[21] Artur Klingbeil, Cassandra Grützner, and Philipp Schreck. 2024. Trust and reliance on AI—An experimental study on the extent and costs of overreliance on AI. *Computers in Human Behavior* 160 (2024), 108352.

[22] Adriano Koshiyama, Emre Kazim, Philip Treleaven, Pete Rai, Lukasz Szpruch, Giles Pavey, Ghazi Ahamat, Franziska Leutner, Randy Goebel, Andrew Knight, et al. 2024. Towards algorithm auditing: managing legal, ethical and technological risks of AI, ML and associated algorithms. *Royal Society Open Science* 11, 5 (2024), 230859.

[23] Michelle S Lam, Mitchell L Gordon, Danaë Metaxa, Jeffrey T Hancock, James A Landay, and Michael S Bernstein. 2022. End-user audits: A system empowering communities to lead large-scale investigations of harmful algorithmic behavior. *proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–34.

[24] Markus Langer, Kevin Baum, and Nadine Schlicker. 2024. Effective Human Oversight of AI-Based Systems: A Signal Detection Perspective on the Detection of Inaccurate and Unfair Outputs. *Minds and Machines* 35, 1 (2024), 1.

[25] Qinbin Li, Junyuan Hong, Chulin Xie, Jeffrey Tan, Rachel Xin, Junyi Hou, Xavier Yin, Zhun Wang, Dan Hendrycks, Zhangyang Wang, Bo Li, Bingsheng He, and Dawn Song. 2024. LLM-PBE: Assessing Data Privacy in Large Language Models. arXiv:2408.12787 [cs.CR] https://arxiv.org/abs/2408.12787

[26] Tianshi Li, Sauvik Das, Hao-Ping Lee, Dakuo Wang, Bingsheng Yao, and Zhiping Zhang. 2024. Human-Centered Privacy Research in the Age of Large Language Models. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–4.

[27] Youwei Li, Yangyang Li, and Yangzhao Yang. 2024. Test-Agent: A Multimodal App Automation Testing Framework Based on the Large Language Model. In *2024 IEEE 4th International Conference on Digital Twins and Parallel Intelligence (DTPI)*. 609–614. doi:10.1109/DTPI61353.2024.10778901

[28] 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).

[29] 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).

[30] Jeffrey Lund, Sean Macfarlane, and Brooke Niles. 2024. Privacy audit of commercial large language models with sophisticated prompt engineering. *Preprint* (2024).

[31] Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. 2023. Can llms keep a secret? testing privacy

implications of language models via contextual integrity theory. *arXiv preprint arXiv:2310.17884* (2023).

[32] Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, Xintong Li, Jing Shi, Hongjie Chen, Viet Dac Lai, Zhouhang Xie, Sungchul Kim, Ruiyi Zhang, Tong Yu, Mehrab Tanjim, Nesreen K. Ahmed, Puneet Mathur, Seunghyun Yoon, Lina Yao, Branislav Kveton, Thien Huu Nguyen, Trung Bui, Tianyi Zhou, Ryan A. Rossi, and Franck Dernoncourt. 2024. GUI Agents: A Survey. arXiv:2412.13501 [cs.AI] https://arxiv.org/abs/2412.13501

[33] Helen Nissenbaum. 2004. Privacy as contextual integrity. *Wash. L. Rev.* 79 (2004), 119.

[34] OpenAI. 2025. Introducing Operator-Safety and privacy. https://openai.com/index/introducing-operator/ Accessed: 2025-01-19.

[35] Ashwinee Panda, Xinyu Tang, Milad Nasr, Christopher A Choquette-Choo, and Prateek Mittal. 2024. Privacy auditing of large language models. In *ICML 2024 Next Generation of AI Safety Workshop*.

[36] Amazon Science. 2024. PrivLM-Bench: A Multi-Level Privacy Evaluation Benchmark for Language Models. *Amazon Science* (2024). https://www.amazon.science/publications/privlm-bench-a-multi-level-privacy-evaluation-benchmark-for-language-models

[37] Yijia Shao, Tianshi Li, Weiyan Shi, Yanchen Liu, and Diyi Yang. 2024. Privacylens: Evaluating privacy norm awareness of language models in action. *arXiv preprint arXiv:2409.00138* (2024).

[38] Hong Shen, Alicia DeVos, Motahhare Eslami, and Kenneth Holstein. 2021. Everyday algorithm auditing: Understanding the power of everyday users in surfacing harmful algorithmic behaviors. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–29.

[39] Yueqi Song, Frank Xu, Shuyan Zhou, and Graham Neubig. 2025. Beyond Browsing: API-Based Web Agents. arXiv:2410.16464 [cs.CL] https://arxiv.org/abs/2410.16464

[40] Sarah Sterz, Kevin Baum, Sebastian Biewer, Holger Hermanns, Anne Lauber-Rönsberg, Philip Meinel, and Markus Langer. 2024. On the Quest for Effectiveness in Human Oversight: Interdisciplinary Perspectives. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*. 2495–2507.

[41] Adriana Tiron-Tudor and Delia Deliu. 2022. Reflections on the human-algorithm complex duality perspectives in the auditing process. *Qualitative Research in Accounting & Management* 19, 3 (2022), 255–285.

[42] Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael S Bernstein, and Ranjay Krishna. 2023. Explanations can reduce overreliance on ai systems during decision-making. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (2023), 1–38.

[43] Maria Virvou and George A Tsihrintzis. 2023. Pre-made Empowering Artificial Intelligence and ChatGPT: The Growing Importance of Human AI-Experts. In *2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 1–8.

[44] Johannes Walter. 2023. *Human oversight done right: The AI Act should use humans to monitor AI only when effective.* Technical Report. ZEW policy brief.

[45] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2023. DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models.. In *NeurIPS*.

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

[47] Xingyi Wang, Xiaozheng Wang, Sunyup Park, and Yaxing Yao. 2025. Users' Mental Models of Generative AI Chatbot Ecosystems. *arXiv preprint arXiv:2501.19211* (2025).

[48] Hao Wen, Shizuo Tian, Borislav Pavlov, Wenjie Du, Yixuan Li, Ge Chang, Shanhui Zhao, Jiacheng Liu, Yunxin Liu, Ya-Qin Zhang, and Yuanchun Li. 2024. AutoDroid-V2: Boosting SLM-based GUI Agents via Code Generation. arXiv:2412.18116 [cs.AI] https://arxiv.org/abs/2412.18116

[49] Hui Yang, Sifu Yue, and Yunzhong He. 2023. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224* (2023).

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

[51] 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. *arXiv preprint arXiv:2402.07939* (2024).

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

[53] Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. 2024. Agent Security Bench: A Benchmark for Evaluating Security and Privacy in LLM-Based Agents. *OpenReview* (2024). https://openreview.net/forum?id=V4y0CpX4hK

[54] Li Zhang, Shihe Wang, Xianqing Jia, Zhihan Zheng, Yunhe Yan, Longxi Gao, Yuanchun Li, and Mengwei Xu. 2024. LlamaTouch: A Faithful and Scalable Testbed for Mobile UI Task Automation. arXiv:2404.16054 [cs.HC] https://arxiv.org/abs/2404.16054

[55] 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.

[56] Yanzhe Zhang, Tao Yu, and Diyi Yang. 2024. Attacking Vision-Language Computer Agents via Pop-ups. *arXiv preprint arXiv:2411.02391* (2024).

[57] Zhiping Zhang, Bingcan Guo, and Tianshi Li. 2024. Privacy Leakage Overshadowed by Views of AI: A Study on Human Oversight of Privacy in Language Model Agent. *arXiv preprint arXiv:2411.01344* (2024).

[58] Zhiping Zhang, Michelle Jia, Hao-Ping Lee, Bingsheng Yao, Sauvik Das, Ada Lerner, Dakuo Wang, and Tianshi Li. 2024. "It's a Fair Game", or Is It? Examining How Users Navigate Disclosure Risks and Benefits When Using LLM-Based Conversational Agents. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–26.

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