Foundations and Recent Trends in Multimodal Mobile Agents: A Survey

Anonymous ACL submission

Abstract

Mobile agents are essential for automating tasks in complex and dynamic mobile environments. As foundation models evolve, they offer increasingly powerful capabilities for understanding and generating natural language, enabling real-time adaptation and processing of multimodal data. This survey provides a comprehensive review of mobile agent technologies, with a focus on recent advancements in foundation models. Our analysis begins by introducing the core components and exploring key representative works in mobile benchmarks and interactive environments, aiming to fully understand the research focuses and their limitations. We then categorize these advancements into two main approaches: prompt-based methods, which utilize large language models (LLMs) for instruction-based task execution, and training-based methods, which fine-tune multimodal models for mobile-specific applications. By discussing key challenges and outlining future research directions, this survey offers valuable insights for advancing mobile agent technologies.

1 Introduction

Mobile agents have achieved notable success in handling complex mobile environments, enabling the automation of task execution across various applications with minimal human intervention (Zhang et al., 2023a; Li et al., 2024b; Bai et al., 2024). These agents are designed to perceive, plan, and execute in dynamic environments, making them highly suitable for mobile platforms that demand real-time adaptability. Over the years, research on mobile agents has evolved from simple rule-based systems to more sophisticated models capable of handling multimodal data and complex decision-making processes (Shi et al., 2017; Rawles et al., 2023). At the same time, foundation models, such as large language models (LLMs) and multimodal models, have become pivotal in enabling mobile agents to better understand, generate, and adapt to their environments, thus expanding their capabilities in mobile systems (Ma et al., 2024; Bai et al., 2024).

The growing complexity of mobile environments and the increasing demand for automation highlight the importance of mobile agents (Deng et al., 2024a). They play a critical role in applications such as mobile interfaces, autonomous navigation, and intelligent assistance, allowing for more efficient and intelligent task execution (Yang et al., 2023). As mobile technologies advance, mobile agents are expected to operate in environments that require continuous adaptation to changing inputs, multimodal data processing, and interaction with various user interfaces (Zhang et al., 2023a). The ability to process and integrate diverse data sources in real-time makes mobile agents essential for enabling seamless user experiences and efficient operations in dynamic mobile platforms.

Despite their progress, mobile agents face several challenges. Traditional evaluation methods often fail to capture real-world mobile tasks' dynamic and interactive nature, limiting their assessment accuracy (Deng et al., 2024a). To address this, recent benchmarks such as AndroidEnv (Toyama et al., 2021) and Mobile-Env (Zhang et al., 2023a) have been developed to evaluate agents in more realistic, interactive mobile environments, focusing on adaptability and task performance. These benchmarks provide a more comprehensive assessment of mobile agents by measuring task completion and their ability to respond to changes in the environment.

Addressing complex tasks while ensuring mobile agents are multimodal, scalable, adaptable, and resource-efficient remains a significant challenge. Recent advancements in multimodal mobile agent research can be categorized into prompt-based and training-based methods. Prompt-based methods leverage large language models (LLMs), such as ChatGPT (OpenAI, 2023) and GPT-4 (OpenAI, 2023), to handle complex tasks by using instruction prompting and chain-of-thought (CoT) reasoning (Zhang et al., 2024c). Notable works such as AppAgent (Yang et al., 2023) and AutoDroid (Wen et al., 2024) have demonstrated the potential of prompt-based systems in interactive mobile environments, although scalability and robustness remain ongoing challenges. On the other hand, training-based methods focus on fine-tuning multimodal models, such as LLaVA (Liu et al., 2023a) and Qwen-VL (Bai et al., 2023), specifically for mobile applications. These models can handle rich, multimodal data by integrating visual and textual inputs, improving their ability to perform tasks like interface navigation and task execution (Ma et al., 2024; Dorka et al., 2024).

This survey provides a comprehensive review of multimodal mobile agent technologies, focusing on recent advancements and ongoing challenges. First, we explore the core components of mobile agents, including perception, planning, action, and memory, which collectively enable agents to operate effectively in dynamic environments. Furthermore, we explore the benchmarks and evaluation methods used to assess mobile agent performance. Next, we categorize mobile agents into prompt-based and training-based approaches, discussing their strengths and limitations in improving agent adaptability, reasoning, and task execution. Finally, we discuss the future development directions of multimodal mobile agents. By providing a clear understanding of the current state of mobile agent research, this survey identifies key areas for future exploration and offers insights into the development of more adaptive, efficient, and capable mobile agents.

2 The Components of Mobile Agents

As shown in Fig. 1, this section outlines the four fundamental components of mobile agents: perception, planning, action, and memory. Together, these components enable agents to perceive, reason, and execute within dynamic mobile environments, adapting their behavior dynamically to improve task efficiency and robustness.

2.1 Perception

Perception is the process through which mobile agents gather and interpret multimodal information from their surroundings. In mobile agents, the perception component focuses on handling multimodal information from different environments, extracting relevant information to aid in planning and task execution.

Early research on mobile agents (Zhang et al., 2021; Sunkara et al., 2022; Song et al., 2023) primarily relied on simple models or tools to convert images or audio into text descriptions. However, these approaches often generate irrelevant and redundant information, hampering effective task planning and execution, especially in content-heavy interfaces. Additionally, the input length limitations of LLMs further amplified these challenges, making it difficult for agents to filter and prioritize information during task processing. Existing visual encoders, mostly pre-trained on general data, are not sensitive to interactive elements in mobile data. To address this, recent studies by Seeclick (Cheng et al., 2024) and CogAgent (Hong et al., 2024) have introduced mobile-specific datasets that enhance visual encoders' ability to detect and process key interactive elements, such as icons, within mobile environments.

In contexts where API calls are accessible, Mind2Web (Deng et al., 2024b) and AutoDroid (Wen et al., 2024) introduces a method for processing HTML-based information. This method ranks key elements of HTML data and filters crucial details to improve LLM perception of interactive components (Li et al., 2024b). Meanwhile, Octopus v2 (Chen and Li, 2024) leverages specialized functional tokens to streamline function calls, significantly enhancing on-device language model efficiency and reducing computational overhead.

2.2 Planning

Planning is central to mobile agents, enabling them to formulate action strategies based on task objectives and dynamic environments. Unlike agents in static settings, mobile agents must adapt to ever-changing inputs while processing multimodal information.

Planning strategies can be categorized as dynamic or static. In static planning, agents break down tasks into sub-goals but do not re-plan if errors occur (Zhang et al., 2024c). In contrast, dynamic planning adjusts the plan based on real-time feedback, enabling agents to revert to earlier states and re-plan (Gao et al., 2023b; Wang et al., 2024a). Recent advances in prompt engineering have further enhanced mobile agent planning. OmniAct (Kapoor et al., 2024) employs prompt-based techniques to structure multimodal inputs and im-

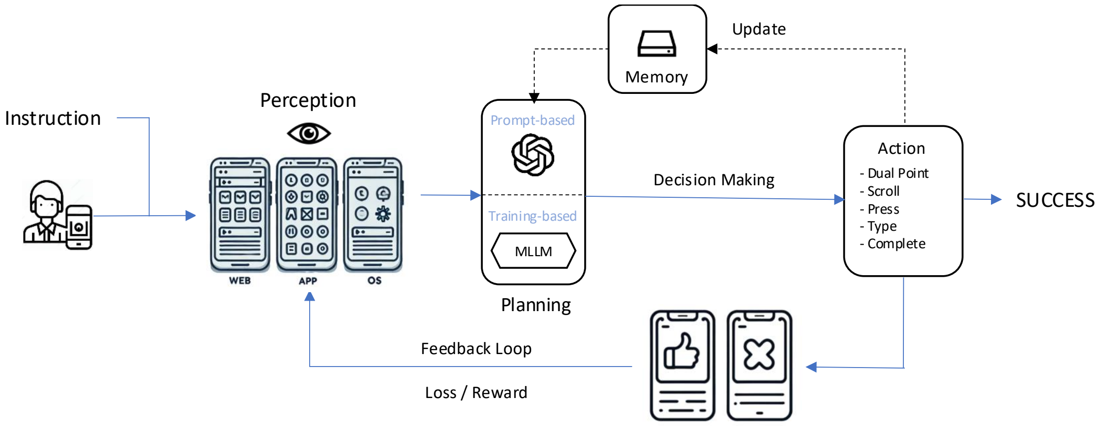

Figure 1: This pipeline shows the decision-making process of mobile agents: User instructions are processed through web, app, and OS interfaces, followed by planning with prompt-based or training-based methods. Actions are taken, and feedback is used to update memory, enabling continuous learning to achieve success.

prove reasoning capabilities. This approach allows agents to integrate external tools and adjust output formats dynamically and efficiently.

2.3 Action

The action component demonstrates how agents execute tasks in a mobile environment by utilizing three key aspects: screen interactions, API calls, and agent interactions. Through screen interactions, agents tap, swipe, or type on GUIs, imitating human behavior to navigate apps. They also make API calls to access deeper system functions, such as issuing commands to automate tasks beyond the GUI (Chen and Li, 2024). Additionally, by collaborating with other agents, they enhance their ability to adapt to complex tasks, ensuring efficient task execution across diverse environments (Zhang et al., 2024c).

Screen Interactions In mobile environments, interactions often involve actions like tapping, swiping, or typing on virtual interfaces. Agents, such as those in AiTW, AITZ, and AMEX (Rawles et al., 2024b; Chai et al., 2024; Zhang et al., 2024c), perform GUI-based actions by mimicking human interactions, ensuring they work smoothly with native apps. These actions go beyond simple gestures, including complex multi-step processes requiring agents to dynamically adapt to changes or new inputs (Lee et al., 2021; Wang et al., 2022).

API Calls Mobile agents rely on various methods to interact with GUIs and perform tasks that require deep integration with mobile operating systems, with API calls serving as the foundation (Chen and Li, 2024; Kapoor et al., 2024). Building on API calls, mobile agents can further leverage HTML and XML data to access core functions, modify device settings, retrieve sensor data, and automate app navigation, extending their capabilities beyond GUI-based inputs (Chai et al., 2024; Chen and Li, 2024; Li et al., 2024b). By integrating these approaches, agents can efficiently complete tasks while gaining a more comprehensive understanding of their environment.

2.4 Memory

Memory mechanisms are crucial for mobile agents, allowing them to retain and use information across tasks. Current research maps in-context learning to short-term and long-term memory to external vector stores.

Short-term Memory Managing task continuity and adaptation effectively requires temporarily storing and reasoning about information, much like human working memory. Recent advancements have focused on improving the memory capabilities of mobile agents. For example, Auto-UI (Zhan and Zhang, 2023) incorporates historical text information to enhance decision-making by retaining past context, while UI-VLM (Dorka et al., 2024) uses image-based memory storage. Unlike single-modality agents, multimodal agents must handle short-term memory across various data types, including text, images, and interactions, ensuring that crucial information from different sources is retained.

Long-term Memory Handling long-term complex information requires a more efficient memory system. While external vector stores can retrieve past experiences, they function differently from human long-term memory, which is highly structured and interconnected. A combination of parametric memory and vector databases is currently used to mimic human-like long-term memory. Parametric memory stores implicit and semantic information, while vector databases hold more recent semantic and episodic memories. To make querying easier, some approaches convert multimodal inputs into a unified text format for storage, simplifying retrieval and integration during task execution (Yang et al., 2023; Wang et al., 2024b; Wen et al., 2024).

3 Mobile Datasets and Benchmarks

Benchmarks establish a standardized testing environment for evaluating and comparing the performance of mobile agents across both static and interactive settings, covering areas such as user interface automation, task completion, and real-world application scenarios.

Currently, as shown in Table 5, many benchmarks for GUI interaction rely on static datasets (Sun et al., 2022; Deng et al., 2024b; Niu et al., 2024; Roßner et al., 2020), which provide fixed ground-truth annotations and evaluate models by comparing their action sequences to predefined solutions. This method is problematic, as it penalizes alternative valid approaches, marking them as failures even if the task is successfully completed. Interactive benchmarks, such as AndroidArena (Xing et al., 2024), also use action sequence similarity as a primary evaluation metric, resulting in an inadequate assessment of agent performance. While recent studies on LLM-based GUI agents (Yang et al., 2023; Wang et al., 2024a; Zhang et al., 2024a) incorporate LLMs or human evaluations, these experiments are often conducted in uncontrolled open environments, leading to issues with reproducibility and comparability of results.

3.1 Static Datasets

Static datasets provide a controlled and predefined set of tasks with annotated ground-truth solutions, making them essential for evaluating the performance of mobile agents in fixed environments. These datasets are primarily used to assess task automation, where agents are required to follow predetermined actions or commands to complete specific tasks.

Early research links referring expressions to UI elements on a screen, with each instance containing a screen, a low-level command, and the corresponding UI element. For example, the RicoSCA dataset (Deka et al., 2017) uses synthetic commands, while MiniWoB++ (Liu et al., 2018) includes sequences of low-level commands for multi-step tasks.

Recent research has shifted towards task-oriented instructions, where each episode contains action-observation pairs, including screenshots and tree-structured representations like Android's View Hierarchy or the Document Object Model in web environments. For instance, the PixelHelp (Li et al., 2020a) dataset contains 187 high-level task goals with step-by-step instructions from Pixel Phone Help pages, while the UGIF (Venkatesh et al., 2022) dataset extends similar queries to multiple languages. Meanwhile, MoTIF(Burns et al., 2021), includes 4.7k task demonstrations, with an average of 6.5 steps per task and 276 unique task instructions. AITW (Rawles et al., 2024b) is much larger, featuring 715,142 episodes and 30,378 unique prompts, some inspired by other datasets.

3.2 Simulation Environments

Simulation environments provide dynamic platforms where agents engage with the environment in real time, receiving feedback and adjusting their actions accordingly. Unlike static datasets, these environments allow for continuous, adaptive interactions, making them critical for evaluating agents in more complex, evolving scenarios.

Before the rise of LLM-based agents, research primarily focused on reinforcement learning (RL)-based agents. A prominent example is Android-Env (Toyama et al., 2021), which provided RL agents with an environment to interact with mobile applications via predefined actions and rewards. However, with advancements in LLMs, the focus has shifted towards agents that can use natural language understanding and generation to perform more flexible and adaptive tasks (Liu et al., 2024b; Sun et al., 2024b,a).

Simulation Environments are a key focus in current research on LLM-based agents, particularly in their ability to explore decision paths autonomously through interactions with the environment (Liu et al., 2024b; Sun et al., 2024b,a). In mobile settings, these agents are designed to handle complex, multi-step tasks and simulate human-like behav-

Dataset	Templates	Attach	Task	Reward	Platform
Static Dataset					
RICOSCA (Deka et al., 2017)	259k	VH	Grounding	-	Android
AndroidHowTo (Deka et al., 2017)	10k	-	Extraction	-	Android
PixelHelp (Li et al., 2020a)	187	VH	Apps	-	Android
Screen2Words (Wang et al., 2021)	112k	VH	Summarization	-	Android
META-GUI (Sun et al., 2022)	1,125	XML	Apps+Web	-	Android
MoTIF (Burns et al., 2021)	4,707	VH	Apps	-	Android
UGIF (Venkatesh et al., 2022)	4184	XML	Grounding	-	Android
AitW (Rawles et al., 2024b)	30k	Layout	Apps+Web	-	Android
AitZ (Zhang et al., 2024c)	2504	CoT	Apps+Web	-	Android
AMEX (Chai et al., 2024)	3k	Layout	Apps+Web	-	Android
Mobile3M (Chen et al., 2024a)	3M	XML	Apps	-	Android
Androidcontrol (Li et al., 2024a)	15283	VH	Apps+Web	-	Android
MobileViews-600K (Gao et al., 2024)	600k	VH	Apps	-	Android
Ferret-UI (You et al., 2024)	120k	VH	Apps	-	IOS
Odyssey (Lu et al., 2024)	7735	-	Apps+Web	-	Multi Platforms
ScreenSpot (Cheng et al., 2024)	1200	Layout	Apps+Web	-	Multi Platforms
GUI-World (Chen et al., 2024a)	12379	Video	Apps+Web	-	Multi Platforms
Interactive Environment					
MiniWoB++ (Liu et al., 2018)	114	-	Web (synthetic)	Sparse Rewards	-
AndroidEnv (Toyama et al., 2021)	100	-	Apps	Sparse Rewards	Android
AppBuddy (Shvo et al., 2021)	35	-	Apps	Sparse Rewards	Android
Mobile-Env (Zhang et al., 2023a)	224	VH	Apps+Web	Dense Rewards	Android
AndroidArena (Wang et al., 2024c)	221	XML	Apps+Web	Sparse Rewards	Android
AndroidWorld (Rawles et al., 2024a)	116	-	Apps+Web	Sparse Rewards	Android
DroidTask (Wen et al., 2024)	158	XML	Apps+Web	-	Android
B-MoCA (Lee et al., 2024)	60	XML	Apps+Web	-	Android
AppWorld (Trivedi et al., 2024)	750	API	Apps+Web	-	Android
Mobile-Bench (Deng et al., 2024a)	832	XML	Apps+Web	-	Android
MobileAgentBench (Wang et al., 2024c)	100	VH	Apps+Web	Dense Rewards	Android
LlamaTouch (Zhang et al., 2024d)	60	-	Apps+Web	-	Android
Spa-Bench (Chen et al., 2024b)	340	-	Apps+Web	Dense Rewards	Android
AndroidLab (Xu et al., 2024b)	138	XML	Apps+Web	Dense Rewards	Android
CRAB (Xu et al., 2024a)	23	XML	Apps+Web	Dense Rewards	Multi Platforms

Table 1: Comparison of various platforms based on parallelization, templates, tasks per template, rewards, and supported platforms. Layout refers to fine-grained annotation of the image content, such as the positional coordinates of elements. VH means View Hierarchy.

iors for app automation (Wen et al., 2023a,b; Liu et al., 2023c; Yao et al., 2022a; Shvo et al., 2021). A notable example is Mobile-Env (Zhang et al., 2023a), created to evaluate how well agents manage multi-step interactions in mobile environments. Ultimately, this research aims to improve the adaptability and flexibility of LLM-based agents, allowing them to function in dynamic, real-world environments with minimal reliance on predefined scripts or manual input.

3.3 Real-world Environments

Real-world Environments present a significant opportunity to address one of the main limitations of closed-reinforcement learning settings: their inability to fully capture the complexity and variability of real-world interactions. While controlled environments are useful for training and testing agents, they often miss the dynamic elements of real-world scenarios, where factors like changing content, unpredictable user behavior, and diverse device configurations are crucial. To overcome these challenges, researchers are increasingly exploring open, real-world environments for LLM-based GUI agents, enabling them to learn and adapt to the intricacies of live systems and evolving situations (Gao et al., 2023a; Wang et al., 2024b; Zhang et al., 2024a; Yang et al., 2023). However, deploying agents in open-world settings introduces several risks. These include safety concerns, irreproducible results, and the potential for unfair comparisons. To mitigate these issues and ensure fair, reproducible evaluations, researchers advocate for strategies such as fixing dynamic online content and employing replay mechanisms during evaluation (Liu et al., 2018; Shi et al., 2017; Zhou et al., 2023). These methods help create a more controlled testing environment, even within the broader scope of open-world deployments.

3.4 Evaluation Methods

In evaluating agent performance, trajectory evaluation, and outcome evaluation are two main methods. Trajectory evaluation focuses on how well agent actions align with predefined paths. In contrast, outcome evaluation emphasizes whether the agent achieves its final goals, focusing on results rather than the specific process. The following sections will explore recent research advancements in these two areas, highlighting how more comprehensive evaluation strategies can enhance our understanding of agent performance in complex environments.

Trajectory Evaluation Recent improvements in GUI interaction benchmarks have focused on step-by-step assessments, comparing predicted actions to reference action trajectories to evaluate agent performance effectiveness (Rawles et al., 2024b; Zhang et al., 2021). While this approach is effective in many cases, task completion often has multiple valid solutions, and agents might explore different paths that do not necessarily follow the predefined trajectories. To improve the flexibility and robustness of these evaluations, Mobile-Env evaluate a subset of signals from the environment of an intermediate state, enabling reliable assessment across a wider range of tasks (Zhang et al., 2023a).

Outcome Evaluation An agent's success is determined by assessing whether it reaches the desired final state, treating task goals as subsets of hidden states, regardless of the path taken to achieve them. These final states can be identified through various system signals. Relying on a single signal type may not capture all relevant state transitions, as certain actions, such as form submissions, may only be visible in the GUI and not in system logs (Toyama et al., 2021) or databases (Rawles et al., 2024a). Shifting to outcome-based evaluation and using multiple signals can make GUI interaction benchmarks more reliable and adaptable, allowing agents to show their full abilities in various scenarios (Wang et al., 2024c; Rawles et al., 2024a).

3.5 Performance Comparison

Due to the limitations of current benchmarks, variations in implementation methods, and changes in platforms, comparing all methods within a unified evaluation environment is challenging. Meanwhile, both prompt-based and training-based approaches suffer from inconsistent evaluation metrics, complicating cross-study comparisons. Methods such as AppAgent (Li et al., 2024b) and AutoDroid (Wen et al., 2024) introduce their own benchmarks and metrics, but only test within these benchmarks and compare against models like GPT-4. These disparities make direct experimental comparisons impractical at this stage. Therefore, after reviewing experimental results from different studies, we compared the AITW and MobileAgentbench benchmarks. AITW measures instruction accuracy (Rawles et al., 2024a), and MobileAgentbench measures Success Rate (Wang et al., 2024c). See tables 3 and 4 in the appendix for more details and the need for standardized benchmarks in future research.

4 The Taxonomy of Mobile Agents

This section introduces a taxonomy of mobile agents, categorizing them into two primary types: prompt-based methods and training-based methods. As shown in Table 6, prompt-based agents take advantage of advancements in LLM to interpret and execute instructions through natural language processing, often focusing on tasks that require dynamic interaction with GUI. Training-based methods involve fine-tuning models or applying reinforcement learning to enhance agents' decision-making and adaptability over time.

4.1 Prompt-based Methods

Recent advancements in LLMs have demonstrated significant potential in developing autonomous GUI agents, particularly in tasks that require instruction following (Sanh et al., 2022; Taori et al., 2023; Chiang et al., 2023) and chain-of-thought (CoT) prompting (Nye et al., 2022; Wei et al., 2022). CoT prompting (Wei et al., 2022; Kojima et al., 2022; Zhang et al., 2023d), in particular, has proven effective in enabling LLMs to handle step-by-step processes, make decisions, and execute actions. These capabilities have shown to be highly beneficial in tasks involving GUI control (Rawles et al., 2023).

GUI Tools Enabling LLMs to interact with GUI is essential, as these models are primarily designed to process natural language rather than visual elements. GUI tools play a crucial role in bridging this gap by allowing LLMs to interpret and interact with visual elements through text-based commands, making it possible for the models to process and respond to graphical interface components. This multimodal integration significantly boosts the efficiency and flexibility of mobile agents in complex environments. Techniques like icon recognition and OCR (Zhang et al., 2021; Sunkara et al.,

Method	Input Type	Model	Training	Memory	Multi-agents
Prompt-based Methods					
DroidGPT (Wen et al., 2023b)	Text	ChatGPT	None	✗	✗
AppAgent (Yang et al., 2023)	Image&Text	GPT-4V	None	✓	✗
MobileAgent (Wang et al., 2024b)	Image&Text	GPT-4V & DINO & OCR	None	✓	✗
MobileAgent v2 (Wang et al., 2024a)	Image&Text	GPT-4V & DINO & OCR	None	✓	✓
AutoDroid (Wen et al., 2024)	Text	GPT-4 & Vicuna-7B	None	✓	✓
AppAgent V2 (Li et al., 2024b)	Image&Text	GPT-4V	None	✓	✗
VLUI (Lee et al., 2024)	Image&Text	GPT-4V	None	✗	✗
MobileExperts (Zhang et al., 2024b)	Image&Text	GPT-4V	None	✗	✓
Mobile-Agent-E (Wang et al., 2025)	Image&Text	GPT-4o & DINO & Qwen-VL-Plus	None	✓	✓
Training-based Methods					
MiniWob (Liu et al., 2018)	Image	DOMNET	RL-based	✗	✗
MetaGUI (Sun et al., 2022)	Image&Text	Faster-RCNN & Transformer	Pre-trained	✗	✗
CogAgent (Hong et al., 2023)	Image&Text	CogVLM	Pre-trained	✗	✗
SeeClick (Cheng et al., 2024)	Image&Text	Qwen-VL	Pre-trained	✗	✗
AutoGUI (Zhang and Zhang, 2023)	Image&Text	BLIP-2-ViT & FLAN-Alpaca	Finetune	✓	✗
ResponsibleTA (Zhang et al., 2023c)	Image&Text	Swin-ViT & BART	Finetune	✓	✗
UI-VLM (Dorka et al., 2024)	Image&Text	Qwen-VL	Finetune	✓	✗
Coco-Agent (Ma et al., 2024)	Image&Text	CLIP-ViT & LLama2-7B	Finetune	✓	✗
DigiRL (Bai et al., 2024)	Image&Text	BLIP-2-ViT & Flan-T5 & RoBerta	RL-based	✗	✗
SphAgent (Chai et al., 2024)	Image&Text	SPHINX-X	Finetune	✗	✗
Octopus v2 (Chen and Li, 2024)	Text	Gemma-2B	Finetune	✗	✗
Octo-planner (Chen et al., 2024d)	Text	Phi-3 Mini	Finetune	✗	✓
MobileVLM (Wu et al., 2024a)	Image&Text	Qwen-VL-Chat	Finetune	✗	✗
OdysseyAgent (Lu et al., 2024)	Image&Text	Qwen-VL	Finetune	✓	✗
AutoGLM (Liu et al., 2024a)	Image&Text	ChatGLM	RL-based	✗	✗
LiMAC (Christianos et al., 2024)	Image&Text	Qwen2-VL-2B & AcT	Finetune	✗	✗
DistRL (Wang et al., 2024e)	Image&Text	BLIP-2-ViT & Flan-T5	RL-based	✗	✗
ShowUI (Qinghong Lin et al., 2024)	Image&Text	Qwen2-VL-2B	Finetune	✗	✗
OmniParser (Wan et al., 2024)	Image&Text	BLIP-2 & OCR & YoloV8	Finetune	✗	✗
OS-Atlas (Wu et al., 2024b)	Image&Text	InternVL-2-4B \ Qwen2-VL-7B	Pre-trained	✗	✗
AppVLM (Papoudakis et al., 2025)	Image&Text	Paligemma-3B-896	RL-based	✗	✗
InfiGUIAgent (Liu et al., 2025)	Image&Text	Qwen2-VL-2B	Finetune	✗	✗

Table 2: Comparison of Mobile Agents: A Detailed Overview of Input Types, Models, Training Methods, Memory Capabilities, and Multi-agent Support.

2022; Song et al., 2023) are used to parse GUI elements, which then converts the parsed elements into HTML layouts. However, this method relies heavily on external tools (Rawles et al., 2023; Wen et al., 2023a) and app-specific APIs (Zhou et al., 2023; Gur et al., 2023), often resulting in inefficiencies and errors during inference. Although some research has investigated multimodal architectures to process different types of inputs (Sun et al., 2022; Yan et al., 2023), these approaches still depend on detailed environment parsing for optimal performance. Given the importance of accurate GUI grounding, newer studies (Cheng et al., 2024; Hong et al., 2023) have begun exploring pre-training methods to improve agent performance in GUI tasks.

Memory Mechanism Effective task execution in prompt-based methods relies on a strong memory mechanism to retain and use relevant information. In agents like AppAgent (Yang et al., 2023), the agent employs an exploration phase for memory, allowing it to learn and adapt to new applications by storing interactions from prior explorations. This approach enables the agent to retain knowledge without needing additional training data. Mobile-Agent (Wang et al., 2024b,a) automates mobile app operations by analyzing screenshots with visual tools, avoiding reliance on system code.

Complex Reasoning In agent systems, complex reasoning refers to the ability of models to process, analyze, and integrate information from multiple sources to solve intricate tasks. This capability enhances decision-making, planning, and adaptability by enabling agents to draw connections between different data inputs, evaluate various outcomes, and execute informed actions in dynamic environments. CoAT (Zhang et al., 2024c) enhances GUI agent performance by integrating semantic information into action generation. It combines screen descriptions, action reasoning, next-action descriptions, and predicted outcomes to improve decision accuracy and consistency.

4.2 Training-based Methods

In contrast to prompt-based methods, training-based approaches involve explicit model optimization. These agents fine-tune large language models like LLama (Zhang et al., 2023b) or multimodal models such as LLaVA (Liu et al., 2023a) by collecting instruction-following data to obtain instruction information.

Pre-trained VLMs In mobile environments, pre-trained VLMs have become powerful tools for decision-making and interaction. Models like LLaVA (Liu et al., 2023a) and Qwen-VL (Bai et al., 2023), pre-trained on large-scale general datasets, capture both visual and language information effectively. However, their applicability in mobile settings is limited by the lack of sensitivity to interactive elements specific to mobile data. To improve the responsiveness of pre-trained models to interactive elements in mobile data, CogAgent (Hong et al., 2023) collected a large-scale mobile dataset for pre-training representations. CogAgent (Hong et al., 2023) integrates visual and textual inputs for GUI agents, improving interaction with complex mobile UIs using VLMs. Spotlight (Li and Li, 2022) is a vision-language model for mobile UI tasks, relying solely on screenshots and specific regions, supporting multi-task and few-shot learning, trained on a large-scale dataset. VUT (Li et al., 2021) employs a dual-tower Transformer for multi-task UI modeling, achieving competitive performance with fewer models and reduced computational costs.

Fine-Tuning The process of fine-tuning pre-trained VLMs with commonsense reasoning capabilities has been facilitated by large-scale mobile datasets, such as AitW (Rawles et al., 2024b), through the Visual Instruction Tuning approach. Existing methods primarily involve two areas: dataset enhancement, and training strategies improvement. ScreenAI (Baechler et al., 2024) and AMEX (Chai et al., 2024) focus on using synthetic data and multi-level annotations to precisely identify and describe UI elements on mobile interfaces, providing high-quality datasets for complex question-answering and navigation tasks. On the other hand, Auto-GUI (Zhan and Zhang, 2023), UI-VLM (Dorka et al., 2024), COCO-Agent (Ma et al., 2024), Octo-planner (Chen et al., 2024d), and AutoDroid (Wen et al., 2024) achieve significant model performance improvements through strategies such as direct interface interaction, task instruction and element layout improvement, and separating planning from execution. These techniques not only optimize automation processes but also enhance the prediction accuracy and operational efficiency of models in practical applications.

Reinforcement Learning A dynamic approach to training mobile agents is offered by reinforcement learning, which enables them to learn from interactions with environments. This method is particularly effective in scenarios where the agent must adapt to sequential decision-making tasks or optimize its actions based on rewards. The WoB (Shi et al., 2017) platform enables reinforcement learning in real environments by allowing agents to interact with websites using human-like actions. Meanwhile (Shi et al., 2017) converts action prediction into question-answering, improving task generalization across different environments. MiniWoB++ (Liu et al., 2018) introduces workflow-guided exploration, which integrates expert workflows with task-specific actions, accelerating learning and improving task efficiency in action prediction tasks. DigiRL (Bai et al., 2024) combines offline and online reinforcement learning to train device control agents. It scales online training using a VLM-based evaluator that supports real-time interaction with 64 Android emulators, enhancing the efficiency of RL-based agent training.

5 Conclusion

This survey provides a comprehensive overview of multimodal mobile agent technologies. Firstly, we discussed the core components—perception, planning, action, and memory—that enable mobile agents to adapt to their environments, forming the foundation of their functionality. Next, we reviewed advancements in mobile agents' benchmarks, which improve mobile agent assessments but still require more comprehensive methods to capture real-world dynamics. We then presented a taxonomy of mobile agents, differentiating between prompt-based and training-based methods, each with strengths and challenges in scalability and adaptability. Finally, we highlighted future research directions, focusing on security, adaptability, and multi-agent collaboration to advance mobile agent capabilities.

6 Limitations

This survey focuses on recent advancements in LLM-based mobile agents but provides limited coverage of traditional, non-LLM-based systems. The lack of discussion on older rule-based agents may limit the broader context of mobile agent technology development.

References

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

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

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A Plummer. 2021. Mobile app tasks with iterative feedback (motif): Addressing task feasibility in interactive visual environments. *arXiv preprint arXiv:2104.08560*.

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

Dongping Chen, Yue Huang, Siyuan Wu, Jingyu Tang, Liuyi Chen, Yilin Bai, Zhigang He, Chenlong Wang, Huichi Zhou, Yiqiang Li, et al. 2024a. Gui-world: A dataset for gui-oriented multimodal llm-based agents. *arXiv preprint arXiv:2406.10819*.

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

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

Wei Chen and Zhiyuan Li. 2024. Octopus v2: On-device language model for super agent. *arXiv preprint arXiv:2404.01744*.

Wei Chen, Zhiyuan Li, Zhen Guo, and Yikang Shen. 2024d. Octo-planner: On-device language model for planner-action agents. *arXiv preprint arXiv:2406.18082*.

Xingyu Chen, Zihan Zhao, Lu Chen, Danyang Zhang, Jiabao Ji, Ao Luo, Yuxuan Xiong, and Kai Yu. 2021. Websrc: A dataset for web-based structural reading comprehension. *arXiv preprint arXiv:2101.09465*.

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

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://vicuna.lmsys.org.

Filippos Christianos, Georgios Papoudakis, Thomas Coste, Jianye Hao, Jun Wang, and Kun Shao. 2024. Lightweight neural app control. *arXiv preprint arXiv:2410.17883*.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 845–854.

Shihan Deng, Weikai Xu, Hongda Sun, Wei Liu, Tao Tan, Jianfeng Liu, Ang Li, Jian Luan, Bin Wang, Rui Yan, et al. 2024a. Mobile-bench: An evaluation benchmark for llm-based mobile agents. *arXiv preprint arXiv:2407.00993*.

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

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

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

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

Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. 2023b. Assistgpt: A general multi-modal assistant that can plan, execute, inspect, and learn. *arXiv preprint arXiv:2306.08640*.

Longxi Gao, Li Zhang, Shihe Wang, Shangguang Wang, Yuanchun Li, and Mengwei Xu. 2024. Mobileviews: A large-scale mobile gui dataset. *arXiv preprint arXiv:2409.14337*.

Yiduo Guo, Zekai Zhang, Yaobo Liang, Dongyan Zhao, and Duan Nan. 2023. Pptc benchmark: Evaluating large language models for powerpoint task completion. *arXiv preprint arXiv:2311.01767*.

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*, abs/2307.12856.

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. *arXiv preprint arXiv:2401.13919*.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. 2023. Cogagent: A visual language model for gui agents. *ArXiv preprint*, abs/2312.08914.

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

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

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *ArXiv preprint*, abs/2205.11916.

Hung-Yi Lee, Mitra Mohtarami, Shang-Wen Li, Di Jin, Mandy Korpusik, Shuyan Dong, Ngoc Thang Vu, and Dilek Hakkani-Tur, editors. 2021. *Proceedings of the 1st Workshop on Meta Learning and Its Applications to Natural Language Processing*. Association for Computational Linguistics, Online.

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

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

Wei Li, William E Bishop, Alice Li, Christopher Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. 2024a. On the effects of data scale on ui control agents. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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

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

Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. 2020b. Widget captioning: Generating natural language description for mobile user interface elements. *arXiv preprint arXiv:2010.04295*.

Yang Li, Gang Li, Xin Zhou, Mostafa Dehghani, and Alexey Gritsenko. 2021. Vut: Versatile ui transformer for multi-modal multi-task user interface modeling. *arXiv preprint arXiv:2112.05692*.

Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. 2024. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. 2018. Reinforcement learning on web interfaces using workflow-guided exploration. *arXiv preprint arXiv:1802.08802*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *ArXiv preprint*, abs/2304.08485.

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

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023b. Agentbench: Evaluating llms as agents. *ArXiv preprint*, abs/2308.03688.

Yuhan Liu, Xiuying Chen, Xiaoqing Zhang, Xing Gao, Ji Zhang, and Rui Yan. 2024b. From skepticism to acceptance: Simulating the attitude dynamics toward fake news. *arXiv preprint arXiv:2403.09498*.

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

Zhe Liu, Chunyang Chen, Junjie Wang, Mengzhuo Chen, Boyu Wu, Xing Che, Dandan Wang, and Qing Wang. 2023c. Chatting with gpt-3 for zero-shot human-like mobile automated gui testing. *arXiv preprint arXiv:2305.09434*.

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

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

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

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

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2022. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop*.

OpenAI. 2023. Chatgpt. `https://openai.com/blog/chatgpt/`. 1, 2.

OpenAI. 2023. Gpt-4 technical report.

Georgios Papoudakis, Thomas Coste, Zhihao Wu, Jianye Hao, Jun Wang, and Kun Shao. 2025. Appvlm: A lightweight vision language model for online app control. *arXiv preprint arXiv:2502.06395*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

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 e-prints*, pages arXiv–2411.

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

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

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy P Lillicrap. 2023. Androidinthewild: A large-scale dataset for android device control. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Daniel Roßner, Claus Atzenbeck, and Daniel Urban. 2020. Weblinks: Augmenting web browsers with enhanced link services. In *Proceedings of the 3rd Workshop on Human Factors in Hypertext*, pages 1–5.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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

Fangxun Shu, Yue Liao, Le Zhuo, Chenning Xu, Lei Zhang, Guanghao Zhang, Haonan Shi, Long Chen, Tao Zhong, Wanggui He, et al. 2024. Llava-mod: Making llava tiny via moe knowledge distillation. *arXiv preprint arXiv:2408.15881*.

Maayan Shvo, Zhiming Hu, Rodrigo Toro Icarte, Iqbal Mohomed, Allan D Jepson, and Sheila A McIlraith. 2021. Appbuddy: Learning to accomplish tasks in mobile apps via reinforcement learning. In *Canadian AI*.

Yunpeng Song, Yiheng Bian, Yongtao Tang, and Zhongmin Cai. 2023. Navigating interfaces with ai for enhanced user interaction. *ArXiv preprint*, abs/2312.11190.

Hongda Sun, Hongzhan Lin, Haiyu Yan, Chen Zhu, Yang Song, Xin Gao, Shuo Shang, and Rui Yan. 2024a. Facilitating multi-role and multi-behavior collaboration of large language models for online job seeking and recruiting. *arXiv preprint arXiv:2405.18113*.

Hongda Sun, Yuxuan Liu, Chengwei Wu, Haiyu Yan, Cheng Tai, Xin Gao, Shuo Shang, and Rui Yan. 2024b. Harnessing multi-role capabilities of large language models for open-domain question answering. In *Proceedings of the ACM on Web Conference 2024*, pages 4372–4382.

Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. 2022. META-GUI: Towards multi-modal conversational agents on mobile GUI. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6699–6712, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Srinivas Sunkara, Maria Wang, Lijuan Liu, Gilles Baechler, Yu-Chung Hsiao, Jindong Chen, Abhanshu Sharma, and James W. W. Stout. 2022. Towards better semantic understanding of mobile interfaces. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5636–5650, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Daniel Toyama, Philippe Hamel, Anita Gergely, Gheorghe Comanici, Amelia Glaese, Zafarali Ahmed, Tyler Jackson, Shibl Mourad, and Doina Precup. 2021. Androidenv: A reinforcement learning platform for android. *arXiv preprint arXiv:2105.13231*.

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

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

Jianqiang Wan, Sibo Song, Wenwen Yu, Yuliang Liu, Wenqing Cheng, Fei Huang, Xiang Bai, Cong Yao, and Zhibo Yang. 2024. Omniparser: A unified framework for text spotting key information extraction and table recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15641–15653.

Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. 2021. Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 498–510.

Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024a. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *arXiv preprint arXiv:2406.01014*.

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

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

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024d. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

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

Xintong Wang, Florian Schneider, Özge Alacam, Prateek Chaudhury, and Chris Biemann. 2022. MOTIF: Contextualized images for complex words to improve human reading. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2468–2477, Marseille, France. European Language Resources Association.

Zhenhailong Wang, Haiyang Xu, Junyang Wang, Xi Zhang, Ming Yan, Ji Zhang, Fei Huang, and Heng Ji. 2025. Mobile-agent-e: Self-evolving mobile assistant for complex tasks. *arXiv preprint arXiv:2501.11733*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv preprint*, abs/2201.11903.

Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. 2023a. Empowering llm to use smartphone for intelligent task automation. *ArXiv preprint*, abs/2308.15272.

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

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

Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. 2023. Webui: A dataset for enhancing visual ui understanding with web semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

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

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

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. 2024c. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*.

Mulong Xie, Sidong Feng, Zhenchang Xing, Jieshan Chen, and Chunyang Chen. 2020. Uied: a hybrid tool for gui element detection. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1655–1659.

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

Mingzhe Xing, Rongkai Zhang, Hui Xue, Qi Chen, Fan Yang, and Zhen Xiao. 2024. Understanding the weakness of large language model agents within a complex android environment. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6061–6072.

Tianqi Xu, Linyao Chen, Dai-Jie Wu, Yanjun Chen, Zecheng Zhang, Xiang Yao, Zhiqiang Xie, Yongchao Chen, Shilong Liu, Bochen Qian, et al. 2024a. Crab: Cross-platfrom agent benchmark for multi-modal embodied language model agents. In *NeurIPS 2024 Workshop on Open-World Agents*.

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

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

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

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022b. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. 2024. Ferret-ui: Grounded mobile ui understanding with multimodal llms.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. 2024. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *arXiv preprint arXiv:2405.10292*.

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

Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei Lin, Saravan Rajmohan, et al. 2024a. Ufo: A ui-focused agent for windows os interaction. *arXiv preprint arXiv:2402.07939*.

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

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

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

Li Zhang, Shihe Wang, Xianqing Jia, Zhihan Zheng, Yunhe Yan, Longxi Gao, Yuanchun Li, and Mengwei Xu. 2024d. Llamatouch: A faithful and scalable testbed for mobile ui automation task evaluation. *arXiv preprint arXiv:2404.16054*.

Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2023b. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *ArXiv preprint*, abs/2303.16199.

Xiaoyi Zhang, Lilian de Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, et al. 2021. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, and Yan Lu. 2023c. Responsible task automation: Empowering large language models as responsible task automators. *arXiv preprint arXiv:2306.01242*.

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

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023d. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

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

A Appendix

A.1 Future Work

In this survey, we have presented the latest advancements in the field of mobile agents. While significant progress has been made, several challenges remain unresolved. Based on the current state of research, we propose the following future research directions:

Model Architecture Optimization : When optimizing mobile agent performance, it is crucial to consider the impact of grounding ability on the action prediction task. To achieve this, the model must enhance its grounding ability for accurate element localization while effectively adapting to action prediction tasks and making efficient decisions. The Mixture of Experts (MOE) architecture plays a key role in this process (Lin et al., 2024). By introducing multiple expert modules, MOE allows the model to dynamically select the most suitable expert based on the task, making it particularly effective in handling multi-domain tasks and improving task adaptability and performance (Shu et al., 2024). Therefore, adopting the MOE architecture enhances grounding ability while ensuring strong decision-making capability in complex tasks, leading to improved performance across multiple domains (Wu et al., 2024c).

Combine with Reinforcement Learning: Enhancing mobile agents' ability to adapt to dynamic and unpredictable environments is crucial. Mobile agent tasks are fundamentally decision-making tasks, not just prediction tasks (Liu et al., 2018). Training through instruction fine-tuning can improve predictions within the action space, but it struggles with decision data influenced by predicted outcomes that cause distribution changes, such as in virtual machines or simulators. These scenarios require the use of reinforcement learning to perform sequential decision-making tasks. However, research in this area is still in its early stages. Current explorations, such as Digirl (Bai et al., 2024), Distrl(Wang et al., 2024e) and RL4VLM (Zhai et al., 2024), have not yet achieved end-to-end alignment in this field. Future research should explore how to utilize reinforcement learning better to integrate changing interactive environments with multimodal large language models for real-time behavior adjustment.

Security and Privacy: Mobile agents face security risks in open environments. Whether it involves tasks that make decisions in latent spaces, such as those found in the AITW (Rawles et al., 2024a) and AMEX (Chai et al., 2024) datasets, or tasks like AITZ (Zhang et al., 2024c) that complete decision-making through chains-of-thought , the security of the model and its ethical aspects can impact decision-making performance (Bai et al., 2022) . Future work should prioritize stronger security mechanisms to prevent malicious behavior and data breaches. It's also necessary to develop privacy protection technologies and ethical improvement mechanisms to ensure safe and ethical operations during agent interactions.

Multi-agent Collaboration: Collective intelligence simplifies complex problems through distributed control, enhances system robustness with redundant designs, and optimizes resource usage through coordinated operations, thereby demonstrating significant efficiency and adaptability in handling large-scale, complex tasks. Improving collaboration among multiple mobile agents remains a key challenge. Current methods exploring multi-agent systems are still limited to role-playing(Li et al., 2024b), standard operating procedures (Zhang et al., 2024b), and collaboration with expert models (Chen et al., 2024d). The overall scale is small, lacking exploration into communication and organizational structures. Future research should focus on efficient communication and collaborative mechanisms that enable agents to dynamically form coalitions and complete tasks more effectively.

Model Lightweighting: The computational resources on mobile devices are limited, which imposes higher requirements for model deployment and inference. Therefore, quantization and accelerating inference are particularly important. Existing methods such as SphAgent (Chai et al., 2024), CogAgent (Hong et al., 2023), and SeeClick (Cheng et al., 2024) still have too large parameter sizes for deployment on mobile devices. The latest research, like LiMAC (Christianos et al., 2024), reduces fine-tuning costs without compressing model parameters. Future research should focus on optimizing the size of mobile agents and speeding up the inference process to ensure high performance under resource constraints. Additionally, refining the inference pipeline to enhance real-time decision-making capabilities is crucial, which involves better computational algorithms and hardware accelerations to achieve faster responses and reduce energy consumption.

A.2 Complementary Technologies

Effective complementary technologies are vital for enhancing the performance and usability of mobile agents, in addition to key components like benchmarks, VLM models, fine-tuning methods, and advanced reasoning skills. These technologies facilitate seamless interactions with mobile environments, allowing agents to adapt, learn, and perform complex tasks efficiently.

UIED (Xie et al., 2020) detects and classifies GUI elements using computer vision and deep learning, supporting interactive editing. WebGPT (Nakano et al., 2021) fine-tunes GPT-3 for web-based question answering using imitation learning and human feedback. WebVLN (Chen et al., 2024c) trains AI agents to navigate websites with question-based instructions, incorporating HTML for deeper understanding.

A.3 Available related technologies

Additionally, OmniACT (Kapoor et al., 2024) offers a comprehensive platform for evaluating task automation across various desktop applications and natural language tasks. WebVoyager (He et al., 2024) introduces an automated evaluation protocol using GPT-4V, capturing screenshots during navigation and achieving an 85.3% agreement with human judgments. Furthermore, Widget Captioning (Li et al., 2020b) sets a benchmark for improving UI accessibility and interaction by providing 162,859 human-annotated phrases that describe UI elements from multimodal inputs, paving the way for advancements in natural language generation tasks. Above all, leveraging a diverse set of system signals provides a more comprehensive and accurate assessment of an agent's performance (Xie et al., 2024).

On desktop platforms, research has focused on evaluating how well LLM-based agents utilize APIs and software tools to complete tasks such as file management and presentations (Qin et al., 2023; Guo et al., 2023). AgentBench (Liu et al., 2023b) offers a flexible, scalable framework for evaluating agent tasks, while PPTC Benchmark (Guo et al., 2023) targets the evaluation of LLM-based agents' performance in PowerPoint-related tasks.

Model	Overall	General	Install	GoogleApps	Single	WebShop.
ChatGPT-COT (Ding, 2024)	7.72	5.93	4.38	10.47	9.39	8.42
GPT-4V ZS+HTML (Ding, 2024)	50.54	41.66	42.64	49.82	72.83	45.73
GPT-4V ZS+History (Ding, 2024)	52.96	43.01	46.14	49.18	78.29	48.18
GPT-4o (Wu et al., 2024a)	55.02	47.06	49.12	52.30	80.28	46.42
MobileAgent (Wang et al., 2024b)	66.92	55.8	74.98	63.95	76.27	63.61
InternVL +History (Wu et al., 2024a)	2.63	1.95	2.88	2.94	3.03	2.71
Qwen-VL +History (Wu et al., 2024a)	3.23	2.71	4.11	4.02	3.89	2.58
PaLM-2 (Zhang and Zhang, 2023)	39.6	–	–	–	–	–
MM-Navigator (Yan et al., 2023)	50.54	41.66	42.64	49.82	72.83	45.73
MM-Navigator$_{\text{w/ text}}$ (Yan et al., 2023)	51.92	42.44	49.18	48.26	76.34	43.35
MM-Navigator$_{\text{w/ history}}$ (Yan et al., 2023)	52.96	43.01	46.14	49.18	78.29	48.18
OmniParser (Wan et al., 2024)	50.54	41.66	42.64	49.82	72.83	45.73
BC (Rawles et al., 2023)	68.7	–	–	–	–	–
BC $_{\text{w/ history}}$ (Rawles et al., 2023)	73.1	63.7	77.5	75.7	80.3	68.5
Qwen-2-VL (Wang et al., 2024d)	67.20	61.40	71.80	62.60	73.70	66.70
Show-UI (Qinghong Lin et al., 2024)	70.00	63.90	72.50	69.70	77.50	66.60
Llama 2 (Zhang and Zhang, 2023)	28.40	28.56	35.18	30.99	27.35	19.92
Llama 2+Plan+Hist (Zhang and Zhang, 2023)	62.86	53.77	69.1	61.19	73.51	56.74
Auto-UI (Zhang and Zhang, 2023)	74.27	68.24	76.89	71.37	84.58	70.26
MobileVLM (Wu et al., 2024a)	74.94	69.58	79.87	74.72	81.24	71.70
SphAgent (Chai et al., 2024)	76.28	68.20	80.50	73.30	85.40	74.00
CoCo-LLAVA (Ma et al., 2024)	70.37	58.93	72.41	70.81	83.73	65.98
SeeClick (Cheng et al., 2024)	76.20	67.60	79.60	75.90	84.60	73.10
CogAgent (Hong et al., 2023)	76.88	65.38	78.86	74.95	93.49	71.73
CoCo-Agent (Ma et al., 2024) $*$	77.82	69.92	80.60	75.76	88.81	74.02

Table 3: Experimental results of different methods on static dataset AITW: Action accuracy across main setups, highlighting overall performance in decision-making tasks. The symbol $*$ indicates that CoCo-Agent incorporates layout information during training, while other models rely solely on image data.

Agent	SR ↑	SE ↓	Latency (s) ↓	Tokens ↓	FN Rate ↓	FP Rate ↓
AndroidArena (Xing et al., 2024)	0.22	1.13	18.61	750.47	0.09	0.33
AutoDroid (Wen et al., 2024)	0.27	3.10	4.85	963.48	0.93	0.01
AppAgent (Yang et al., 2023)	0.40	1.29	26.09	1505.09	0.17	0.40
CogAgent (Hong et al., 2023)	0.08	2.42	6.76	579.84	1.00	0.04
MobileAgent (Ding, 2024)	0.26	1.33	15.91	1236.88	0.19	0.31

Table 4: Experimental results of different methods on simulation environment MobileAgentBench. SR (Success Rate), SE (System Error), Latency, Tokens, FN Rate (False Negative Rate), and FP Rate (False Positive Rate) are the metrics used for comparison.

Dataset	Templates	Attach	Task	Reward	Platform
Static Dataset					
`WebSRC` (Chen et al., 2021)	400k	HTML	Web	-	Windows
`WebUI` (Wu et al., 2023)	400k	HTML	Web	-	Windows
`Mind2Web` (Deng et al., 2024b)	2,350	HTML	Web	-	Windows
`Ferret-UI` (You et al., 2024)	120k	-	Apps	-	IOS
`OmniAct` (Kapoor et al., 2024)	9802	Ocr/Seg	Web	-	Windows
`WebLINX` (Roßner et al., 2020)	2,337	HTML	Web	-	Windows
`ScreenAgent` (Niu et al., 2024)	3005	HTML	Web	-	Windows
Interactive Environment					
`WebShop` (Yao et al., 2022a)	12k	-	Web	Product Attrs Match	Windows
`WebArena` (Zhou et al., 2023)	241	HTML	Web	url/text-match	Windows
`VisualWebArena` (Koh et al., 2024)	314	HTML	Web	url/text/image-match	Windows
`Ferret-UI` (You et al., 2024)	314	HTML	Web	url/text/image-match	Windows
`OSWorld` (Xie et al., 2024)	369	-	Web	Device/Cloud state	Linux

Table 5: Comparison of various platforms based on parallelization, templates, tasks per template, rewards, and supported platforms.

Method	Input Type	Training	Memory	Task	Multi-agents
Prompt-based Methods					
ReAct (Yao et al., 2022b)	Text	None	✓	Web	✗
MM-Navigator (Yan et al., 2023)	Image&Text	None	✗	Apps+Web	✗
MindAct (Deng et al., 2024b)	Text	None	✓	Apps+Web	✗
OmniAct (Kapoor et al., 2024)	Text	None	✗	Apps+Web	✗
Training-based Methods					
VUT (Li et al., 2021)	Image&Text	Pre-trained	✗	Web	✗
Spotlight (Li and Li, 2022)	Image&Text	Pre-trained	✗	Web	✗
ScreenAI (Baechler et al., 2024)	Image&Text	Pre-trained	✗	Web	✗
ScreenAgent (Niu et al., 2024)	Image&Text	Pre-trained	✓	Web	✗
SeeClick (Cheng et al., 2024)	Image&Text	Pre-trained	✗	Web	✗

Table 6: Comparison of Mobile Agents: A Detailed Overview of Input Types, Models, Training Methods, Memory Capabilities, Tasks, and Multi-agent Support. Web* means synthesized web data.