ClickAgent: Enhancing UI Location Capabilities of Autonomous Agents

Jakub Hoscilowicz*, **Bartosz Maj***, **Bartosz Kozakiewicz**
Oleksii Tymoshchuk and **Artur Janicki***
Samsung R&D Poland
*Warsaw University of Technology

Abstract

With the growing reliance on digital devices with graphical user interfaces (GUIs) like computers and smartphones, the demand for smart voice assistants has grown significantly. While multimodal large language models (MLLM) like GPT-4V excel in many areas, they struggle with GUI interactions, limiting their effectiveness in automating everyday tasks. In this work, we introduce ClickAgent, a novel framework for building autonomous agents. ClickAgent combines MLLM-driven reasoning and action planning with a separate UI location model that identifies relevant UI elements on the screen. This approach addresses a key limitation of current MLLMs: their inability to accurately locate UI elements. Evaluations conducted using both an Android emulator and a real smartphone show that ClickAgent outperforms other autonomous agents (DigiRL, CogAgent, AppAgent) on the AITW benchmark.

1 Introduction

The current generation of voice assistants (e.g., Google Assistant, Siri, Alexa) relies on established spoken language understanding methods such as Joint NLU (Goo et al., 2018). However, as these approaches have plateaued, the industry is shifting its focus toward AI Agents. These systems, designed to interact with graphical user interfaces (GUIs) autonomously, are becoming critical for automating tasks on digital devices such as smartphones and computers (Kapoor et al., 2024).

Researchers have begun developing agent-oriented large language models (LLMs) (Chen et al., 2023a; Zeng et al., 2023), but the scope of language-only agents is limited in voice assistant applications, where interaction with GUIs is often needed to perform complex tasks. MLLMs and visual language models (VLMs) offer a promising solution to these limitations (You et al., 2023; Rahman et al., 2024; Gur et al., 2024; Baechler et al., 2024). Unlike language-based agents that rely solely on textual data such as HTML (Nakano et al., 2021) or OCR outputs (Rawles et al., 2023), MLLM-based agents directly interpret visual signals from GUIs. However, while current-generation MLLMs demonstrate reasonable abilities in screen understanding, reasoning, and action planning, they struggle to locate specific UI elements on screens accurately (Liu et al., 2024). Previous works (Yang et al., 2023; Fan et al., 2024; Ma et al., 2024a) attempt to bypass this issue, for example, by using an XML file that details the interactive elements or by using a separate OCR model. Yet, such multi-module approaches are error-prone due to the inherent complexity of GUIs and the inconsistencies in XML/HTML files.

Our contribution lies in the development of ClickAgent, a hybrid autonomous agent that combines MLLM-driven reasoning with a specialized UI location model. Specifically, ClickAgent leverages the InternVL2.0 MLLM (Chen et al., 2023b) for reasoning and the TinyClick UI location model for identifying the coordinates of relevant UI elements (Pawlowski et al., 2024). This approach significantly improves performance on the AITW benchmark (Rawles et al., 2024), surpassing prompt-based agents (D-PoT (Zhang et al., 2024b), CogAgent (Hong et al., 2024)) and DigiRL (Bai et al., 2024), a reinforcement learning-based solution.

2 Method

Although models like SeeClick (Cheng et al., 2024a) and Auto-UI (Zhan and Zhang, 2023) excel at identifying UI elements, they lack robust action planning, leading to low success rates in real-world smartphone tasks. To overcome these challenges, ClickAgent integrates InternVL2.0 for reasoning, while a dedicated UI location model identifies the

*Correspondence to: jakub.hoscilowicz.dokt@pw.edu.pl, artur.janicki@pw.edu.pl.

Proceedings of the 26th Annual Meeting of SIGDIAL, pages 471–476
Aug 25–27, 2025.

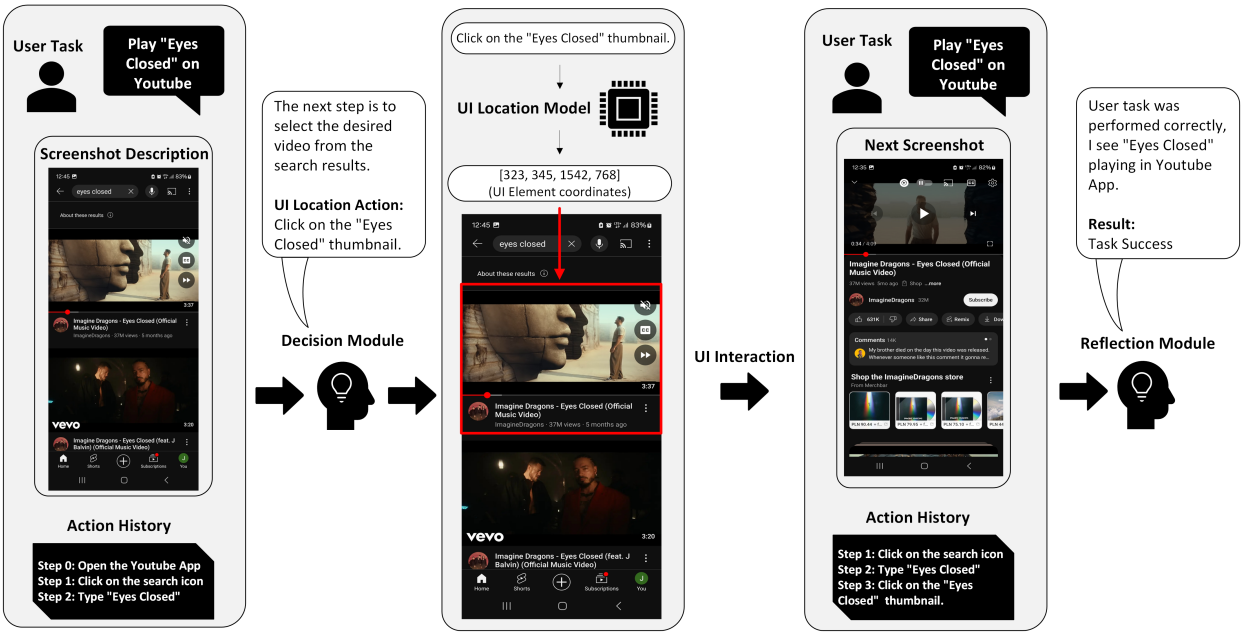

Figure 1: In ClickAgent, the MLLM is responsible for reasoning, reflection and action planning. In this example, the MLLM generates a UI command, and a specialized UI location model identifies the coordinates of the corresponding icon on the screen.

exact coordinates of the target UI elements. The inputs to the UI location model are a screenshot and a natural language command corresponding to the desired UI element. ClickAgent's hybrid approach addresses the limitations of current MLLMs, which struggle to locate UI elements accurately (Liu et al., 2024).

ClickAgent consists of three main components: Decision, UI Location, and Reflection. In the Decision module, the MLLM is asked to analyze the current screenshot, review the action history, and determine the next step to complete the user's task. The Decision module selects one of the predefined actions:

- `Click`: The MLLM generates a natural language command for the UI location model (e.g., "click on the Gmail icon."). The command and screenshot are passed to the UI location model, which returns the coordinates of the relevant UI element.
- `Type`: The MLLM generates the text to be typed into the text field.
- `OpenApp`: If this action is chosen, an additional query is made to the MLLM to select an app from the list retrieved from the Android device.
- `Swipe`: The agent swipes in the specified direction (up, down, left, or right).

For example, as illustrated in Figure 1, the MLLM chooses the Click action and issues a UI action command ("Click on the Eyes Closed Official Video") which, along with a screenshot, serves as input to the UI location model. The model then returns the bounding box coordinates of the relevant UI element.

After the action is executed on Android, the next screenshot is captured. In the Reflection module, the MLLM is asked to analyze the screenshot content and the entire action history. Reflection evaluates whether the user's task was successfully completed, returning either a "success" or "failure" status. If the task is marked as successful, the autonomous agent is stopped; otherwise, the agent proceeds with the next decision and reflection cycle.

InternVL2.0 was selected as the primary MLLM for the Decision and Reflection modules due to its strong performance on agentic benchmarks and ease of deployment. SeeClick and TinyClick were chosen for UI location tasks because of their high performance on benchmarks such as ScreenSpot (Cheng et al., 2024b) and Omni-Act (Kapoor et al., 2024).

Table 1: Performance comparison of autonomous agents on the AITW General and AITW WebShopping benchmark subsets. Task completion rates [%] were calculated based on manual evaluations. The main ClickAgent results were obtained using an actual Android smartphone, while the results with cache removal were performed on an Android smartphone emulator to simulate a first-time user experience.

	AITW General	AITW WebShopping	Overall
AppAgent (GPT-4V)	15.6	13.5	14.0
AutoUI	12.5	18.8	17.2
CogAgent	25.0	42.6	38.3
D-PoT	42.2	36.6	-
DigiRL	70.0	68.8	69.6
ClickAgent (ours)	72.5	**75.8**	**73.5**
w/ Android cache removal	**73.1**	69.9	72.0

3 Evaluation Method

We evaluated ClickAgent on both an Android smartphone emulator and a real Android smartphone, using the task completion rate (in percentage) as the primary performance metric. This metric assesses whether the agent successfully executed the user's task, making it the most critical measure in autonomous agent evaluation. Unlike metrics such as step success rate or action accuracy (Zhan and Zhang, 2023; Ma et al., 2024b; Shen et al., 2024), task completion rate provides a clear assessment of the agent's ability to execute user commands on the device (Zhang et al., 2024a).

Our evaluation was performed manually due to the imprecision of current automatic evaluation methods (Pan et al., 2024). We utilized 4 x NVIDIA A100 80GB GPU for running InternVL-2.0 and one NVIDIA A100 40GB GPU for the UI location models (SeeClick, TinyClick). Each experiment was repeated three times, showing no significant deviation in accuracy across runs. We report the average of these three runs. Except for D-PoT (Zhang et al., 2024b), all results of baseline agents were taken from (Bai et al., 2024).

3.1 Test Environment

The tests were conducted in two scenarios: with and without cache removal. In the cache removal scenario, the emulator's cache was cleared before each test case. This ensured that popups and first-time user interactions appeared for each website, simulating a first-time user experience. In the no-cache removal scenario, conducted on the real Android smartphone, the cache was retained to mimic a user who had previously visited the websites, thereby minimizing or eliminating popups and other initial distractions.

We conducted an evaluation of ClickAgent on a subset of the AITW dataset (Rawles et al., 2023), specifically curated by DigiRL[1], to ensure comparability with this main baseline. The AITW General consists of tasks related to interacting with everyday smartphone applications, while WebShopping focuses on tasks specific to e-commerce platforms.

4 Main Results

Table 1 presents the main results from the AITW benchmark. ClickAgent consistently outperforms other agents (DigiRL, AppAgent and CogAgent), achieving a significantly higher task completion rate, regardless of whether the Android cache was cleared or not. As shown in Table 2, the accuracy of the UI location model plays a crucial role in determining the overall task completion rate, making it a key factor in ClickAgent's performance.

4.1 UI Location Model Analysis

Our primary insight is that TinyClick excels in the OCR-related UI location. Therefore, we adjusted the prompt to encourage the MLLM to generate UI commands that incorporate textual information when possible. For instance, rather than producing commands like "Click on the first email," the MLLM is prompted to return more specific commands such as "Click on the email with the subject 'Meeting Agenda.'". This single prompt modification led to an improvement of around 10 % in performance on the AITW.

4.2 ClickAgent Fails Analysis

On the AITW, the most common failures of ClickAgent were distributed across the following areas,

[1] https://github.com/DigiRL-agent/digirl/tree/master/digirl/environment/android/assets/task_set

Table 2: Performance comparison of ClickAgent using different UI location models on the AITW General and AITW WebShopping (Task completion rates [%]).

	AITW General	AITW WebShopping
InternVL2-76B	0	0
SeeClick-9.6B	47.6	48.8
TinyClick-0.27B	72.5	75.8

with the percentages indicating the proportion of total errors attributed to each component:

- `Reflection Module` (47 %): In most cases, the agent stops the action too early, even though the task has not been completed. From our observation the quality of Reflection is directly linked to the general reasoning capabilities of the MLLM used. Therefore, even reinforcement learning-based agents like DigiRL rely on proprietary MLLMs (Gemini) for Reflection[2].

- `UI Location Model` (15 %): Some UI elements are unique to specific applications, and certain web pages have outdated designs, making it challenging for the UI location model to identify desired elements accurately.

- `Decision Module` (38 %): Similarly to UI location, most decision failures arise from the MLLM's limited understanding of specific UIs and their functionalities.

5 Ablation Study

We conduct an ablation study to understand the impact of two main components (MLLM and UI location model) on the overall performance of ClickAgent. Table 2 evaluates the UI location model's impact on ClickAgent's performance, comparing three different models. As expected, the MLLM (InternVL-2.0-76B) shows poor performance in the UI location task, resulting in ClickAgent failing all test cases. The most significant improvement comes from using the recently released TinyClick, which results in a substantially higher success rate than SeeClick.

Figure 2 illustrates the effect of MLLM general quality on ClickAgent's performance by evaluating four versions of InternVL-2.0 (1B, 7B, 26B, and 76B), alongside Qwen2.0-VL-72B (Wang et al., 2024). The results show that the quality of the MLLM plays a critical role in ClickAgent's performance. Larger models, such as InternVL-2.0-76B, result in significantly higher success rates compared to the smaller variants. Further improvements in the MLLM quality should continue to enhance ClickAgent's performance (especially in terms of the accuracy of the Reflection module). On the other hand, Figure 2 highlights that agents directly deployable on user devices achieve significantly worse performance, rendering them far from ready for commercialization.

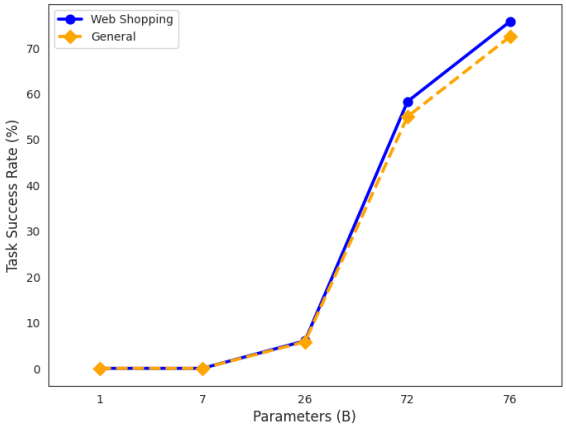

Figure 2: ClickAgent performance on the AITW General and AITW WebShopping using different MLLMs. In all cases, the TinyClick model is employed for UI location.

6 Conclusion

In this paper, we introduced ClickAgent, a hybrid autonomous agent that combines MLLM-driven reasoning with a specialized UI location model. By addressing the limitations of previous approaches in identifying UI elements, ClickAgent achieves a task completion rate of 73.5 %, outperforming state-of-the-art agents like DigiRL (69.6 %) on the AITW benchmark. Notably, it surpasses existing methods without relying on proprietary MLLMs such as Gemini or GPT-4V. The failures observed in the Reflection and Decision modules highlight the need for further advancements in MLLM capabilities, particularly in understanding UIs of less popular apps and websites. One of the most important future directions is research on On-Device Agents, as maintaining GPU infrastructure is costly, and on-device deployment enables better personalization while preserving user privacy.

[2] https://github.com/DigiRL-agent/digirl/blob/master/digirl/environment/android/evaluate.py#L161

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. In *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*.

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*.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. 2023a. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2023b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Computer Vision and Pattern Recognition*.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. 2024a. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Annual Meeting of the Association for Computational Linguistics*.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. 2024b. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Annual Meeting of the Association for Computational Linguistics*.

Yue Fan, Lei Ding, Ching-Chen Kuo, Shan Jiang, Yang Zhao, Xinze Guan, Jie Yang, Yi Zhang, and Xin Eric Wang. 2024. Read anywhere pointed: Layout-aware gui screen reading with tree-of-lens grounding. *arXiv preprint arXiv:2406.19263*.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2024. A real-world webagent with planning, long context understanding, and program synthesis. *Preprint*, arXiv:2307.12856.

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*.

Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. 2024. Visualwebbench: How far have multimodal llms evolved in web page understanding and grounding? *arXiv preprint arXiv:2404.05955*.

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

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2024b. Comprehensive cognitive llm agent for smartphone gui automation. *arXiv preprint arXiv:2402.11941*.

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*.

Jiayi Pan, Yichi Zhang, Nicholas Tomlin, Yifei Zhou, Sergey Levine, and Alane Suhr. 2024. Autonomous evaluation and refinement of digital agents. In *First Conference on Language Modeling*.

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

Abdur Rahman, Rajat Chawla, Muskaan Kumar, Arkajit Datta, Adarsh Jha, Mukunda NS, and Ishaan Bhola. 2024. V-zen: Efficient gui understanding and precise grounding with a novel multimodal llm. *Preprint*, arXiv:2405.15341.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2023. Android in the wild: a large-scale dataset for android device control. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 59708–59728.

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

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

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

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*.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. 2023. Ferret: Refer and ground anything anywhere at any granularity. *Preprint*, arXiv:2310.07704.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*.

Zhuosheng Zhan and Aston Zhang. 2023. You only look at screens: Multimodal chain-of-action agents. In *Annual Meeting of the Association for Computational Linguistics*.

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

Shaoqing Zhang, Zhuosheng Zhang, Kehai Chen, Xinbei Ma, Muyun Yang, Tiejun Zhao, and Min Zhang. 2024b. Dynamic planning for llm-based graphical user interface automation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1304–1320.