"Are We Done Yet?": A Vision-Based Judge for Autonomous Task Completion of Computer Use Agents

Marta Sumyk[1], **Oleksandr Kosovan**[1],

[1]Ukrainian Catholic University
sumyk.pn@ucu.edu.ua, o.kosovan@ucu.edu.ua

Abstract

Computer Use Agents (CUAs) are designed to autonomously operate digital interfaces, yet they often fail to reliably determine whether a given task has been successfully completed. We present an autonomous evaluation and feedback framework that leverages Vision–Language Models (VLMs) to assess task completion directly from screenshots and task descriptions. Our dataset covers 42 built-in macOS applications and $1{,}260$ human-labeled tasks, covering a wide range of scenarios. Our framework achieves up to 73% classification accuracy in task success detection and yields an average relative improvement of 27% in the overall task success rate of CUAs when evaluator feedback is applied. These results demonstrate that vision-based evaluation can serve as an actionable feedback mechanism that significantly improves the reliability and self-correction of autonomous computer-use agents.

Introduction

In recent years, Computer Use Agents (CUAs) (Saunders et al. 2022; OpenAI 2025; Mei et al. 2025) have emerged as a promising paradigm for enabling AI systems to autonomously interact with digital environments, perceiving screen states and performing actions such as clicking, typing, and executing commands to accomplish user-specified goals. Despite their generality and service-agnostic design, a key limitation remains: CUAs often struggle to reliably determine whether a task has been successfully completed. This shortcoming manifests in two critical ways:

- The agent declares a task complete when it is not, undermining user trust and overall reliability (Sun et al. 2025; Sager et al. 2025).
- The agent successfully completes the task but fails to recognize this, leading to redundant actions and unnecessary computational overhead (Sager et al. 2025).

To address these challenges, this work proposes a method for autonomous evaluation of task completion for macOS CUAs, aimed at improving both the success rate and the reliability of CUAs.

We focus on the macOS environment for two main reasons. First, it remains an underexplored domain in the study of screen representations and CUAs (Muryn et al. 2025). While prior work has primarily examined web and mobile environments (Pan et al. 2024; Li et al. 2024; Humphreys et al. 2024), desktop operating systems have received considerably less attention. Desktop interfaces are also inherently more complex to interpret: they typically contain a larger number of visual elements than mobile UIs and lack structured unified representations such as HTML trees available in web environments (Muryn et al. 2025). Second, we choose macOS as an ideal starting point because it offers a controlled yet diverse collection of built-in applications that enable systematic benchmarking of task execution and evaluation. In the future, we aim to extend our framework to other desktop operating systems, including Windows, Linux, and cross-platform web interfaces, toward developing general-purpose and reliable CUAs.

To sum up, the contributions of this work are as follows:

- We introduce a diverse, human-labeled dataset comprising $1{,}260$ tasks across 42 built-in macOS applications.
- We propose a methodology for autonomous evaluation of task completion, achieving up to 73% accuracy and improving success rate of CUAs by an average of 27% relative percentage points on average.

Related Works

Computer Use Agents

CUAs are autonomous agents designed to interact directly with Graphical User Interfaces (GUIs), performing actions such as clicking, typing, scrolling, or navigating web pages. Early examples include browser-based assistants and recent prototypes such as OpenAI's Computer Use tool[1], which integrate vision-language reasoning with low-level action execution. Similar lines of research explore autonomous UI navigation (Gur et al. 2023), multi-modal planning (Li et al. 2024), and end-to-end web automation benchmarks (Humphreys et al. 2024).

Unlike API-based or function-calling agents that require explicit integration with each service, CUAs operate in a service-agnostic manner: they perceive and act directly on the screen, enabling interaction with any digital environment without additional engineering effort. This design paradigm

[1]https://openai.com/index/computer-using-agent/

makes CUAs inherently more flexible and scalable, capable of generalizing across software systems and interfaces (Sun et al. 2025; Sager et al. 2025). However, this generality also introduces new challenges in reasoning and verification. Because CUAs rely solely on visual observations, they can fail silently or partially when confronted with unexpected interface states, visual occlusions, or distribution shifts (Gur et al. 2023; Humphreys et al. 2024; Li et al. 2024). This highlights the need for reliable mechanisms to assess whether the intended task has actually been completed, especially when the task cannot be trivially reduced to log-based success signals.

Autonomous Evaluation of Task Completion

A persistent challenge across all types of agentic systems is evaluating whether a goal has truly been achieved (Zhou et al. 2025; Bhonsle et al. 2025; Zhuge et al. 2024). Reliable evaluation is fundamental for measuring performance, enabling self-improvement, and establishing user trust of agents.

In this work, we adapt this broader evaluation challenge to the domain of CUAs. The problem of determining whether a CUA has successfully completed a task has not been extensively explored (Sager et al. 2025). While prior work has focused primarily on improving action planning and interface understanding (Gur et al. 2023; Li et al. 2024; Humphreys et al. 2024), it has paid little attention to how to tell when a task is actually completed.

A notable exception is script-based evaluation, used in the OSWorld benchmark (Xie et al. 2024). However, this approach relies on manually written verification scripts for each task, which severely limits scalability and makes real-time evaluation impractical. Maintaining reliable automated evaluation across hundreds of GUI tasks requires substantial manual effort, since even minor interface or environment changes can break the scripts and invalidate results.

Also, a closely related effort is the work of Pan et al. (2024), which proposes an autonomous evaluation and refinement framework for digital agents. Their method focuses on automatically assessing and improving web-based and simulated agents by reasoning over structured representations of page elements and textual feedback. Although their approach demonstrates that model-based evaluators can significantly accelerate agent learning, it operates primarily in browser and synthetic environments where the interface semantics and success states are explicitly defined. In contrast, our setting involves real desktop interfaces, specifically macOS, where the screens are harder to parse due to variety and number of elements and also have no universal way of representation as in the web HTML (Muryn et al. 2025). This makes our work a complementary extension of autonomous evaluation to unstructured, multimodal environments that better mirror real-world computer use.

In contrast, the question of task completion by an agent has been more systematically studied in other domains, particularly in robotics. In robotics, recent work such as AutoEval (Zhou et al. 2025) introduces autonomous evaluation frameworks for manipulation policies, reducing the reliance on human annotators or scripted success detectors. It reaches both high agreement with human annotations and reduces the human annotation time by 99%.

Our work builds on this line of research but adapts it to the domain of CUAs. Unlike physical robotics tasks, task completion in digital environments often lacks a straightforward ground-truth signal: for instance, whether "sending an email" was completed correctly may not be directly observable from logs alone. Inspired by AutoEval, we propose to use vision-language models as evaluators that judge whether the current desktop state corresponds to the intended outcome, providing CUAs with reliable, automated feedback.

Methodology

Dataset

Our dataset[2] covers 42 built-in macOS applications, spanning functionality productivity, communication, multimedia, system utilities, and developer tools. For each application, we define 30 concrete tasks, resulting in a total of $1{,}260$ tasks (in comparison, the OSWorld (Xie et al. 2024) constains 369 tasks). The task set is deliberately diverse, ranging from simple actions (e.g., *"Open Calendar app"*) to more complex, multi-step interactions (e.g., *"Filter apps by free in App Store and open the first result"*).

This design ensures coverage across varying levels of difficulty and interaction types, allowing us to evaluate CUAs both on basic GUI navigation skills and on higher-level reasoning about application states. The dataset is intended to simulate realistic end-user goals that CUAs may encounter, while avoiding tasks that require private user data or external configuration (e.g., importing files or logging into accounts).

Autonomous Evaluation

We propose a zero-shot method based on VLMs to automatically evaluate whether a task has been successfully completed [3]. Our pipeline consists of three main steps: 1. The CUA attempts to complete the given task; 2. A VLM receives the final screenshot along with the original task description and predicts whether the task has been successfully completed, providing a natural language justification for its decision; 3. If the VLM judges the task as incomplete, its reasoning is fed back into the CUA. The agent then uses this feedback to attempt the task again, starting from its current state rather than restarting from scratch.

This feedback loop enables CUAs not only to receive automated success signals but also to adapt their behavior dynamically, reducing both task failure and redundant actions. The illustration of our proposed pipeline is shown in Figure 1.

Task Execution. For our experiments, we evaluated three CUAs: Claude Computer Use[4], OpenAI Operator, and UI-TARS (Qin et al. 2025). The first two are proprietary systems, while UI-TARS is open-source. We selected these

[2] https://zenodo.org/records/17696742

[3] https://github.com/martasumyk/vision-based-judge

[4] https://docs.claude.com/en/docs/agents-and-tools/tool-use/computer-use-tool

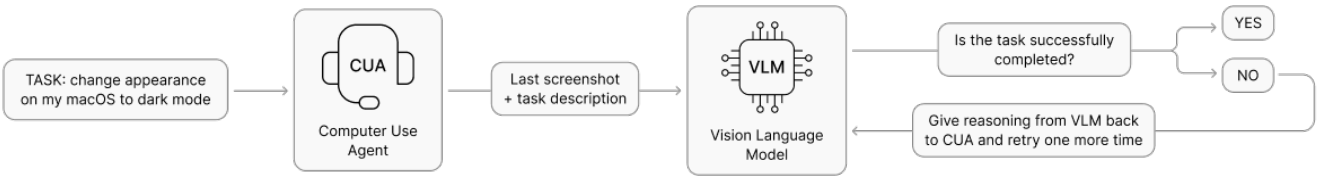

Figure 1: Overview of the proposed evaluation-feedback pipeline. CUA executes a user-defined task (e.g., *"Change appearance on macOS to dark mode"*) and produces a final screenshot of the desktop state. The VLM receives the screenshot and task description, then judges whether the task was successfully completed. If the task is deemed incomplete, the VLM provides reasoning that is passed back to the CUA, which reattempts the task based on this feedback.

agents because they currently achieve leading performance on the OSWorld benchmark (Xie et al. 2024).

Each CUA is provided with a task description and attempts to complete the task within the macOS environment. During execution, the full trajectory is recorded, including step-by-step screenshots, the actions performed at each step (where the action space consists of clicks, double-clicks, typing text, pressing keys, and waiting), and the agent's reasoning at each step.

Outcome Evaluation. In this stage, the task description and the final screenshot of the desktop state are provided to a VLM, which is prompted in a zero-shot setting to determine whether the task has been successfully completed. The VLM produces both a binary judgment (*done/not done*) and a short natural-language rationale explaining its decision.

We evaluate five VLMs that represent both proprietary and open-source families. Among proprietary evaluators, we use GPT-4o[5] and Claude 3.5 Sonnet[6], chosen for their state-of-the-art multimodal reasoning capabilities. For open-source models, we employ LLaVA-v1.5-7B (Liu, Li et al. 2024), InternVL 2-8B (Chen et al. 2024), and Qwen2-VL-7B (Bai et al. 2024), which provide competitive performance. This selection covers a broad spectrum of parameter scales and training paradigms, allowing us to compare evaluation consistency across architectures and accessibility tiers.

This setup enables task evaluation to be performed independently of the acting CUA, reducing bias and ensuring that success is judged solely from observable interface states rather than internal model assumptions. Leveraging general-purpose vision–language reasoning allows the evaluator to robustly handle diverse applications and task types without requiring task-specific rules or heuristics.

Feedback and Retry. If the VLM determines that the task has not been successfully completed, its rationale is passed back to the CUA as feedback. The agent then uses this reasoning to replan and reattempt the task, resuming from its current state rather than restarting the entire trajectory. This feedback loop enables independent correction: the CUA can interpret evaluator feedback, and adjust its strategy accordingly. This feedback mechanism enables the agent to continue from its current state rather than restarting, leading to more efficient retries and higher overall task success rates.

[5]https://openai.com/index/hello-gpt-4o/

[6]https://www.anthropic.com/claude

Results

Evaluator Accuracy Across CUAs

The results in Table 1 show that accuracy of task completion classification, measured against human-annotated ground truth, is consistently high for both proprietary and open-source evaluators. Even in a zero-shot setting, most models demonstrate strong alignment with human judgments, confirming that vision-language models can reliably assess task success.

Effect of Evaluator Feedback on Success Rate

Figure 2 illustrates the effect of evaluator feedback on task success rate across the three CUAs. All evaluated VLM feedback mechanisms lead to measurable performance gains compared to the baseline without feedback. Proprietary evaluators (GPT-4o and Claude 3.5 Sonnet) yield the largest improvements, achieving up to 61% relative success rate gains, while open-source evaluators such as Qwen2-VL-7B also provide consistent boosts in success rate. Notably, agents with lower baseline success rate like Anthropic CU benefit the most from visual feedback, indicating that automated screen-based reasoning helps agents detect and correct incomplete actions. These findings highlight that VLM evaluators not only reliably assess task completion but also enhance the self-correction ability of CUAs through interpretable, vision-grounded feedback.

Future Work

Our task completion evaluation framework opens several promising directions for future research.

First, we plan to expand the framework beyond macOS to additional operating systems such as Linux and Windows. Since fundamental interface components such as windows, menus, buttons, and text are common to Windows, Linux, and macOS, our vision-based approach eliminates the need for OS-specific instrumentation and enables straightforward transfer of the same evaluation logic to new environments. This cross-platform expansion would enable broader support of CUAs across all desktop environments.

Second, our current evaluation employs a binary success metric: a task is considered complete only when the final

Table 1: Task completion classification accuracy (*done/not done*) across proprietary and open-source VLM-based evaluators for three CUAs. **Claude 3.5 Sonnet** achieves the highest proprietary performance, while **Qwen2-VL-7B** leads among open-source models.

Evaluator Model	OpenAI Operator	Anthropic CU	UI-TARS
Proprietary Evaluators			
GPT-4o	0.61	0.69	0.64
Claude 3.5 Sonnet	**0.69**	**0.71**	**0.73**
Open-Source Evaluators			
LLaVA-v1.5-7B	0.56	0.61	0.52
InternVL 2-8B	0.62	**0.67**	0.61
Qwen2-VL-7B	**0.68**	0.66	**0.70**

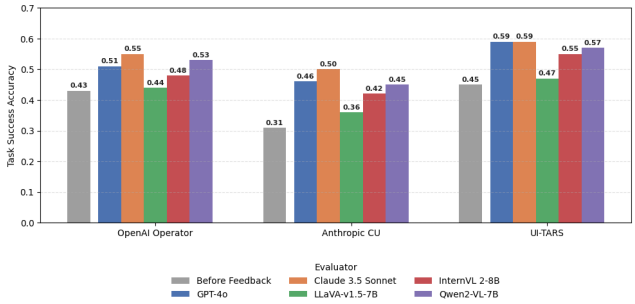

Figure 2: Task success rates before and after evaluator feedback across three CUAs. Gray bars represent baseline success rates before feedback, while colored bars indicate post-feedback (after only one retry) success rate for five VLM evaluators. Proprietary evaluators (GPT-4o and Claude 3.5 Sonnet) achieve the largest relative gains, whereas open-source models (LLaVA-v1.5-7B, InternVL 2-8B, and Qwen2-VL-7B) provide consistent improvements across all agents.

goal is reached. A natural extension is to develop step-level evaluation, in which each intermediate action is judged based on whether it moves the agent closer to the final objective. We also plan to conduct an ablation study to determine how many screenshots or temporal observations are most informative for reliable evaluation.

Third, we plan to conduct inter-model agreement analysis, calibration measurements, and consistency studies across different VLMs. These analyses will quantify how sensitive evaluations are to model choice and provide confidence intervals for task-success predictions. ncorporating techniques such as temperature scaling, conformal prediction, or ensemble averaging may further improve the robustness of judgments.

Forth, the evaluator's output can be used directly as a reward signal within Reinforcement Learning (RL) pipelines for CUAs, providing interpretable, vision-grounded feedback that may improve exploration efficiency and stabilize long-horizon training. By replacing heuristic or human-provided rewards, this approach also reduces the need for large-scale human-labeled datasets in RL pipelines, enabling more scalable and autonomous agent training.

Finally, we aim to extend this work toward multi-agent frameworks (Zhuge et al. 2024; Bhonsle et al. 2025), where the evaluator continuously monitors CUA actions and delivers real-time feedback on each step. Such integration would allow agents to adapt their strategies dynamically, reduce redundant actions, and improve robustness. This could further enable the development of multi-agent systems in which specialized evaluators and actors collaborate to achieve complex computer-use goals.

Conclusion

We presented a framework that autonomously checks whether a CUA has completed its task using only the final screenshot and the task description. Instead of relying on hand-written scripts or system logs, our method uses VLMs to judge task success and provide short feedback that the agent can use to try the task completion again, resuming from the current state. We also provide a diverse, human-labeled dataset of 1,260 tasks across 42 built-in macOS applications to enable reproducible evaluation and support future research.

Across three CUAs and five VLM evaluators, our approach achieves up to **73%** accuracy in identifying completed tasks and improves overall success rates by **27%** on average. We find that weaker agents benefit the most, showing that external visual feedback can make CUAs more reliable and efficient.

Beyond improving accuracy, our framework provides a simple and general way to verify what agents actually achieve on screen. In the future, we plan to extend this work to other operating systems, explore step-by-step evaluation instead of only final results, and use the evaluator as a reward signal in reinforcement learning or multi-agent systems. This moves us closer to building CUAs that can not only act but also correctly recognize when their goals are accomplished.

References

Bai, S.; et al. 2024. Qwen2-VL: A Versatile Vision-Language Model for Understanding and Generation. *arXiv preprint arXiv:2409.12191*.

Bhonsle, R.; Dutta, R.; Vavilapalli, S.; Seth, H.; Jaye, A.; Chang, Y.; Rungta, M.; Boateng, E. A.; Hasan, S.; Nosakhare, E.; and Srinivasan, S. 2025. Auto-Eval Judge:

Towards a General Agentic Framework for Task Completion Evaluation. arXiv:2508.05508.

Chen, X.; et al. 2024. InternVL 2.0: Scaling Up Vision-Language Pretraining and Benchmarking. *arXiv preprint arXiv:2405.07961*.

Gur, I.; Pal, A.; Li, T.; Brockschmidt, M.; Chaudhuri, S.; Riedl, M.; and Andreas, J. 2023. BrowserGym: A Benchmark for Browser Agents. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Humphreys, P.; Ni, A.; Pan, H.; Gur, I.; Zhong, V.; and Andreas, J. 2024. WebArena: A Realistic Web Environment for Building Autonomous Agents. *arXiv preprint arXiv:2307.13854*.

Li, Z.; Zhao, Y.; Chen, Q.; Zhao, Y.; Zhang, H.; Yuan, L.; Lin, B. Y.; Wang, Y.; and Zhang, W. 2024. SeeAct: A Multi-Modal Agent for Web Interaction via Visual Grounding and Action Generation. *arXiv preprint arXiv:2404.05719*.

Liu, H.; Li, C.; et al. 2024. LLaVA 1.5: Improved Multimodal Reasoning and Instruction Following. *arXiv preprint arXiv:2401.02410*.

Mei, K.; Zhu, X.; Gao, H.; Lin, S.; and Zhang, Y. 2025. LiteCUA: Computer as MCP Server for Computer-Use Agent on AIOS. *arXiv preprint arXiv:2505.18829*.

Muryn, V.; Sumyk, M.; Hirna, M.; Garkot, S.; and Shamrai, M. 2025. Screen2AX: Vision-Based Approach for Automatic macOS Accessibility Generation. *arXiv preprint arXiv:2507.XXXXXX*.

OpenAI. 2025. Computer-Using Agent (CUA). Research preview; OpenAI Operator. Combines GPT-4o vision, GUI perception, mouse/keyboard actions; benchmarks: OSWorld, WebArena, WebVoyager.

Pan, J.; Zhang, Y.; Tomlin, N.; Zhou, Y.; Levine, S.; and Suhr, A. 2024. Autonomous Evaluation and Refinement of Digital Agents. arXiv:2404.06474.

Qin, Y.; Ye, Y.; Fang, J.; Wang, H.; Liang, S.; Tian, S.; Zhang, J.; Li, J.; Li, Y.; Huang, S.; Zhong, W.; Li, K.; Yang, J.; Miao, Y.; Lin, W.; Liu, L.; Jiang, X.; Ma, Q.; Li, J.; Xiao, X.; Cai, K.; Li, C.; Zheng, Y.; Jin, C.; Li, C.; Zhou, X.; Wang, M.; Chen, H.; Li, Z.; Yang, H.; Liu, H.; Lin, F.; Peng, T.; Liu, X.; and Shi, G. 2025. UI-TARS: Pioneering Automated GUI Interaction with Native Agents. arXiv:2501.12326.

Sager, P. J.; Meyer, B.; Yan, P.; von Wartburg-Kottler, R.; Etaiwi, L.; Enayati, A.; Nobel, G.; Abdulkadir, A.; Grewe, B. F.; and Stadelmann, T. 2025. A Comprehensive Survey of Agents for Computer Use: Foundations, Challenges, and Future Directions. arXiv:2501.16150.

Saunders, W.; Yeh, C.; Wu, J.; Hilton, J.; Bowman, S.; Glaese, A.; McAleese, N.; Aslanides, J.; et al. 2022. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*.

Sun, Z.; Liu, Z.; Zang, Y.; Cao, Y.; Dong, X.; Wu, T.; Lin, D.; and Wang, J. 2025. SEAgent: Self-Evolving Computer Use Agent with Autonomous Learning from Experience. arXiv:2508.04700.

Xie, T.; Zhang, D.; Chen, J.; Li, X.; Zhao, S.; Cao, R.; Hua, T. J.; Cheng, Z.; Shin, D.; Lei, F.; Liu, Y.; Xu, Y.; Zhou, S.; Savarese, S.; Xiong, C.; Zhong, V.; and Yu, T. 2024. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. arXiv:2404.07972.

Zhou, Z.; Atreya, P.; Tan, Y. L.; Pertsch, K.; and Levine, S. 2025. AutoEval: Autonomous Evaluation of Generalist Robot Manipulation Policies in the Real World. arXiv:2503.24278.

Zhuge, M.; Zhao, C.; Ashley, D.; Wang, W.; Khizbullin, D.; Xiong, Y.; Liu, Z.; Chang, E.; Krishnamoorthi, R.; Tian, Y.; Shi, Y.; Chandra, V.; and Schmidhuber, J. 2024. Agent-as-a-Judge: Evaluate Agents with Agents. arXiv:2410.10934.