Self-Improving Computer Use Agents: A Reinforcement Learning Approach

AI Research Assistant
Generative AI Laboratory
ai.assistant@research.org

Abstract

The development of Computer Use Agents (CUAs) capable of autonomously operating Graphical User Interfaces (GUIs) represents a significant step towards general-purpose AI assistants. Current state-of-the-art approaches, such as ScaleCUA, rely on large-scale, human-curated datasets for supervised training. While effective, this paradigm is data-intensive and limits the agent's ability to adapt and improve beyond the scope of its training data. In this paper, we propose a novel framework, the Self-Improving Computer Use Agent (SI-CUA), which leverages reinforcement learning (RL) to enable agents to learn from their own interactions with the environment. We formulate the GUI task automation problem as a Partially Observable Markov Decision Process (POMDP) and introduce a dense reward shaping mechanism that utilizes a powerful Vision-Language Model (VLM) as a judge to provide fine-grained feedback. By combining pre-training on existing datasets with online RL finetuning, our proposed framework allows the agent to continuously explore, learn from its mistakes, and discover novel strategies for task completion, thereby reducing the reliance on static datasets and paving the way for more adaptive and robust CUAs.

1 Introduction

The ability to interact with digital environments through Graphical User Interfaces (GUIs) is a fundamental aspect of modern computing. The automation of these interactions through Computer Use Agents (CUAs) holds immense potential for increasing productivity, accessibility, and efficiency across a wide range of personal and professional tasks. Recent advancements in Vision-Language Models (VLMs) have significantly propelled the capabilities of CUAs, enabling them to interpret visual screen content and natural language instructions to perform complex actions.

A landmark contribution in this area is the ScaleCUA framework [1], which introduced a large-scale, cross-platform dataset and a family of powerful base models. The ScaleCUA models, trained through supervised learning (behavioral cloning) on this vast corpus of human demonstrations, have demonstrated state-of-the-art performance on a variety of GUI interaction benchmarks. However, this reliance on supervised learning presents several inherent limitations:

- **Data Scalability:** The creation and annotation of large-scale demonstration datasets are expensive and labor-intensive, posing a significant bottleneck to further progress.
- **Static Knowledge:** Agents trained on fixed datasets are limited to the knowledge contained within them. They struggle to adapt to new applications, updated UI layouts, or tasks that require novel sequences of actions not present in the training data.
- **Lack of Optimality:** Behavioral cloning learns to mimic human actions, but these actions may not always be the most efficient or optimal way to complete a task. The agent has no mechanism to discover better strategies on its own.

To overcome these challenges, we propose a paradigm shift from purely supervised learning to a hybrid approach that incorporates reinforcement learning (RL). Our framework, the Self-Improving Computer Use Agent (SI-CUA), enables an agent to learn and refine its skills through direct interaction with the GUI environment. By learning from the consequences of its own actions, the SI-CUA can move beyond simple imitation and towards true goal-oriented problem-solving.

The main contributions of this proposed research are:

1. A novel framework (SI-CUA) that applies deep reinforcement learning to the domain of computer use agents, enabling continuous self-improvement and adaptation.
2. A sophisticated reward shaping mechanism that leverages a pre-trained VLM as a "judge" to provide dense, goal-oriented feedback, addressing the challenge of sparse rewards in complex GUI environments.
3. A hybrid training methodology that combines the benefits of supervised pre-training (for rapid initial learning) with RL fine-tuning (for optimization and discovery of new skills).

This approach promises to create more robust, adaptable, and efficient CUAs, significantly reducing the dependency on manually curated datasets and advancing the development of truly autonomous AI agents.

2 Related Work

2.1 Computer Use Agents

The concept of agents that can operate GUIs has been a long-standing goal in AI. Early approaches relied on structured data like Document Object Models (DOM) or accessibility trees. However, recent progress has been dominated by VLMs that operate directly on raw pixel data, making them more generalizable. The ScaleCUA paper [1] is a prime example of this trend, demonstrating how large-scale, diverse datasets can be used to train highly capable agents via supervised learning. Their work provides a strong foundation and a valuable resource for our proposed research, particularly for pre-training our agent.

2.2 Reinforcement Learning for Complex Tasks

Reinforcement learning has achieved remarkable success in various domains, from game playing [2] to robotics [3]. Applying RL to GUI interaction is challenging due to the high-dimensional state space (raw pixels) and the complex, often hybrid (discrete-continuous) action space. Policy-based algorithms, such as Proximal Policy Optimization (PPO) [4], have proven to be effective in such large-scale environments. PPO's stability and sample efficiency make it a strong candidate for training our SI-CUA.

A critical challenge in applying RL to real-world tasks is reward design. Sparse rewards (e.g., a single reward upon task completion) often lead to intractable exploration problems. Reward shaping [5] is a technique used to provide more frequent, intermediate rewards to guide the learning process. Our work draws inspiration from recent advancements in using large language models to provide reward signals in complex environments [6].

3 Methodology: The Self-Improving CUA (SI-CUA) Framework

We propose the Self-Improving Computer Use Agent (SI-CUA), a framework designed to learn GUI tasks through a combination of supervised pre-training and reinforcement learning fine-tuning.

3.1 Problem Formulation

We formulate the task of a CUA as a Partially Observable Markov Decision Process (POMDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$:

- **State Space ($\mathcal{S}$):** The true, unobserved state of the environment (e.g., the complete state of the operating system and applications).
- **Action Space ($\mathcal{A}$):** A unified, hybrid action space, as defined in ScaleCUA, including actions like click(x, y), type(text), and scroll(d).
- **Transition Function ($\mathcal{T}$):** The dynamics of the environment, $s_{t+1} = \mathcal{T}(s_t, a_t)$.
- **Reward Function ($\mathcal{R}$):** The reward $r_t = \mathcal{R}(s_t, a_t)$ received by the agent.

- **Observation Space (Ω):** The space of possible observations. In our case, an observation $o_t \in \Omega$ is a screenshot of the GUI.
- **Observation Function ($\mathcal{O}$):** The function that maps a state to an observation, $o_t = \mathcal{O}(s_t)$.
- **Discount Factor (γ):** A value in $[0, 1]$ that discounts future rewards.

The agent's goal is to learn a policy $\pi_\theta(a_t|o_t, g)$ conditioned on the current observation o_t and a natural language goal g, which maximizes the expected discounted return $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$.

3.2 Model Architecture

The SI-CUA policy and value functions are parameterized by a single VLM backbone, similar to ScaleCUA. The model takes the current screenshot o_t and the goal description g as input. The VLM encoder produces a joint representation of the visual and textual information. This representation is then fed into two separate heads:

1. **Actor Head:** Outputs the parameters for a probability distribution over the action space $\mathcal{A}$, representing the policy $\pi_\theta(a_t|o_t, g)$.
2. **Critic Head:** Outputs a scalar value $V_\phi(o_t, g)$, which estimates the expected return from the current state (value function).

3.3 Reward Design

The core innovation of our framework is the dense reward shaping mechanism. A sparse reward, only given at the end of a task, would make learning prohibitively difficult. We propose a composite dense reward function:

$$r_t = w_{goal} \cdot r_{goal} + w_{prog} \cdot r_{prog} + r_{step} \tag{1}$$

where w_{goal} and w_{prog} are weighting coefficients.

- **Goal-Conditioned Reward (r_{goal}):** We employ a powerful, pre-trained VLM (e.g., GPT-4o) as a reward model or "judge". At each step t, this VLM is prompted with the current screenshot o_t and the goal g, and asked to output a score from 0 to 1 indicating how close the current state is to achieving the goal.
$$r_{goal} = \text{VLM}_{\text{judge}}(o_t, g) \tag{2}$$
- **Progress-Based Reward (r_{prog}):** To encourage meaningful interactions, we reward the agent for making tangible progress. This is defined as the positive change in the goal-conditioned reward:
$$r_{prog} = \max(0, r_{goal}(o_{t+1}, g) - r_{goal}(o_t, g)) \tag{3}$$
This prevents the agent from being rewarded for simply reaching a good state and staying there without finishing the task.

- **Step Penalty (r_{step}):** A small negative constant is given for each time step. This encourages the agent to find the most efficient solution.

3.4 Training Algorithm

The training process for SI-CUA consists of two phases:

1. **Supervised Pre-training (Behavioral Cloning):** The model is first trained on the ScaleCUA dataset using a standard supervised learning objective (cross-entropy loss). This initializes the policy to a reasonable starting point, significantly accelerating the subsequent RL phase.

$$\mathcal{L}_{BC} = -\mathbb{E}_{(o,g,a)\sim\mathcal{D}}[\log \pi_\theta(a|o, g)] \tag{4}$$

   where $\mathcal{D}$ is the expert demonstration dataset.

2. **RL Fine-tuning (PPO):** After pre-training, the agent is placed in an interactive environment and fine-tuned using Proximal Policy Optimization (PPO). The agent collects trajectories of experience by interacting with the environment. The policy is then updated using the clipped surrogate objective function of PPO, and the value function is updated by minimizing the mean squared error against the empirical returns.

4 Proposed Experiments

4.1 Environments and Baselines

We will evaluate the SI-CUA framework on a suite of benchmarks used in the ScaleCUA paper, including OSWorld for Ubuntu, WebArena-Lite-v2 for web tasks, and WindowsAgentArena for Windows tasks. This will allow for a direct comparison of performance. We will compare our model against two key baselines:

1. **ScaleCUA (Supervised Only):** The original ScaleCUA model trained solely with behavioral cloning. This will establish the performance of the current state-of-the-art.

2. **SI-CUA with Sparse Rewards:** A version of our agent trained with our RL framework but only a binary +1/-1 reward for task success/failure. This will demonstrate the importance of our dense reward shaping mechanism.

4.2 Metrics and Expected Results

The primary metric for evaluation will be the **Task Success Rate**. We will also measure **Sample Efficiency** (how many interactions are required to reach a certain performance level) and **Generalization** (performance on unseen applications and tasks).

We hypothesize that:

- The fully-featured SI-CUA will achieve a significantly higher task success rate than both baselines, demonstrating its ability to optimize beyond simple imitation.
- The SI-CUA with dense rewards will learn much more efficiently than the version with sparse rewards.
- The SI-CUA will exhibit better generalization to novel tasks because it has learned a more robust, goal-oriented problem-solving strategy rather than just memorizing specific action sequences.

5 Conclusion and Future Work

This paper proposes the Self-Improving Computer Use Agent (SI-CUA), a novel framework that integrates reinforcement learning to overcome the limitations of purely supervised approaches for training CUAs. By designing a dense reward signal based on a VLM judge and employing a hybrid training scheme, we aim to develop agents that can learn from experience, adapt to new challenges, and discover optimal strategies for task completion.

Future work could extend this framework in several exciting directions. Incorporating a memory module would allow the agent to handle long-horizon tasks that require recalling past information. Hierarchical reinforcement learning could be used to break down complex goals into manageable sub-tasks. Finally, exploring multi-agent scenarios where multiple CUAs collaborate to solve complex problems presents a fascinating avenue for future research. The development of self-improving agents is a critical step towards creating truly general and autonomous AI systems that can seamlessly assist humans in their daily digital lives.

References

[1] Zhaoyang Liu, Jingjing Xie, Zichen Ding, et al. ScaleCUA: Scaling Open-Source Computer Use Agents with Cross-Platform Data. *arXiv preprint arXiv:2509.15221*, 2025. http://arxiv.org/abs/2509.15221v1

[2] David Silver, Aja Huang, Chris J. Maddison, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[3] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[5] Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[6] Tse-hsun (Allen) Chen, Htoo Htoo, and Y-Lan Boureau. Taming Large Language Models with Human-aligned Reward Functions. *arXiv preprint arXiv:2305.18735*, 2023.