ScaleCUA

Scaling Computer Use Agents with Verifiable Task Synthesis and Efficient Online RL

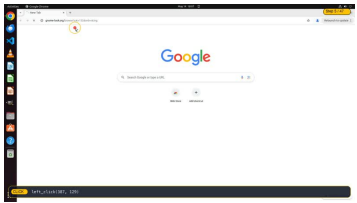
Bowen Lv[1*†], Xiao Liu[1,2*], Yanyu Ren[1†], Hanyu Lai[1†], Bohao Jing[2†], Hanchen Zhang[1†], Yanxiao Zhao[1†], Shuntian Yao[1†], Jie Tang[1], Yuxiao Dong[1]

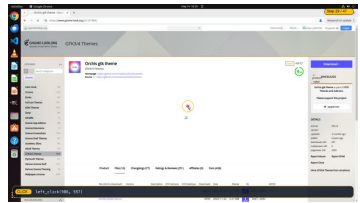
[1]Tsinghua University [2]Z.AI

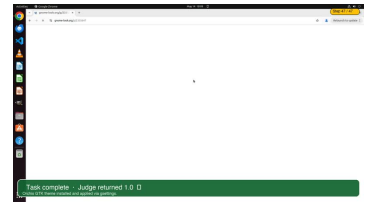
*Equal contribution †Work done while interning at Z.AI

What counts as a successful desktop action?

An agent is asked to install the Orchis theme and apply it to the GNOME desktop. It must carry information from the browser to the file manager and terminal.



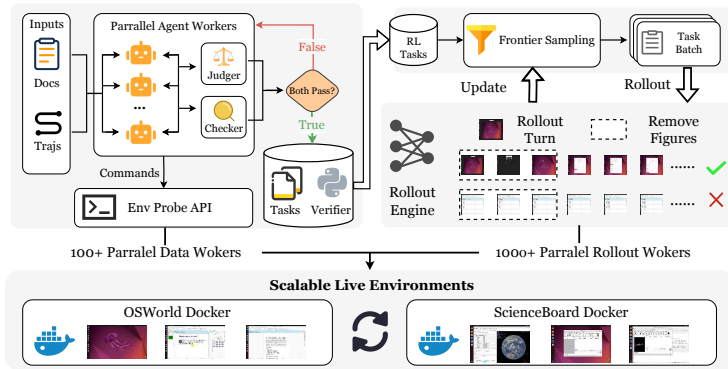




Three moments from a 47-action rollout. The OSWorld evaluator awarded the completed task a reward of 1.

Two training problems follow. Each task needs a way to check the final state, and each long sequence of screenshots costs time to process.

How ScaleCUA trains a desktop agent



VeriGen generates tasks and executable judges. Frontier Sampling chooses which tasks to practice. Visual Context Segmentation controls the cost of learning from each rollout.

VeriGen builds tasks with executable checks

Propose and verify

A proposer writes a task and its starting state. A judge defines a success check. Workers run the check in a live container and send failures back for repair.

Adjust difficulty

Failed trajectories reveal intermediate goals. VeriGen can turn those goals into simpler tasks for practice.

24K+

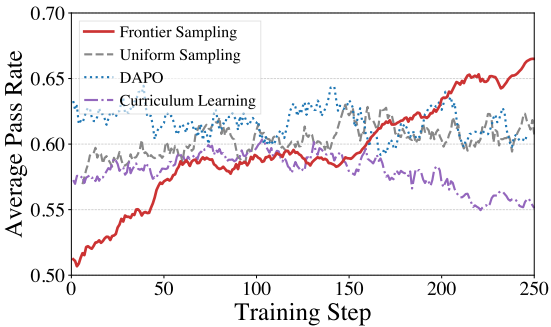
generated tasks across desktop and scientific software

$\sim$3K

tasks selected for online reinforcement learning

In an expert audit of 160 trajectories, judges agreed with people on 82.5%. Executability alone therefore gives an incomplete picture of reward quality.

Which tasks should the agent practice next?

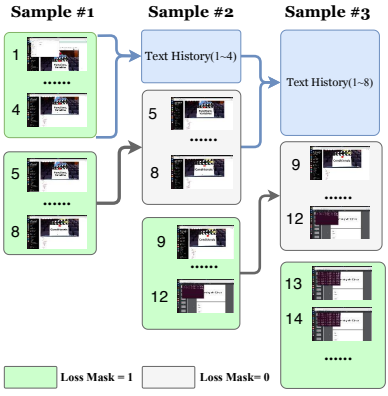


If every rollout for a task earns the same reward, a group-relative update gets little signal about which actions worked.

Frontier Sampling tracks each task's recent success rate and prioritizes tasks near 50%. It reserves 20% of the batch for uniform exploration.

OSWorld 68.7% with Frontier Sampling and 63.7% without it.

Train on short windows of a long trajectory

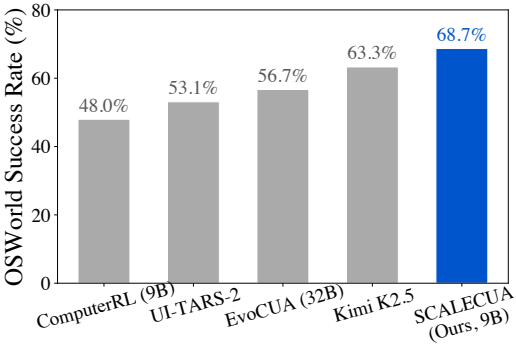


Visual Context Segmentation retains recent screenshots and the preceding text history. Each response contributes training loss in its own segment.

750 s $\rightarrow$ 265 s

Average training step time falls by a factor of 2.83 compared with step-wise decomposition.

Online RL improves desktop task success

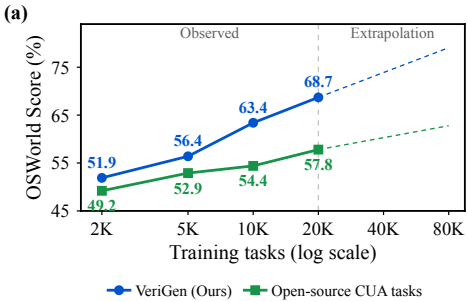


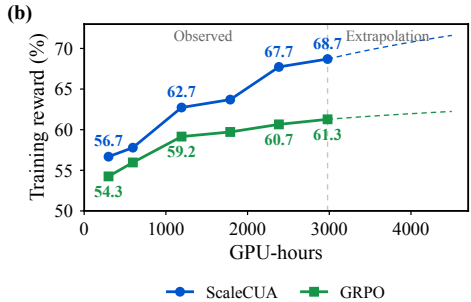
Final ScaleCUA checkpoints, success (%)

Backbone	OSWorld	ScienceBoard
Qwen3.5-9B	**68.7**	**54.0**
GLM-4.6V-Flash	66.5	49.4

On OSWorld, Qwen3.5-9B rises from 60.7% after supervised training to 68.7% after online RL. ScienceBoard evaluates scientific software workflows.

Scaling data and training budget





VeriGen improves OSWorld scores from 51.9% to 68.7% as task count grows from 2K to 20K.

ScaleCUA shows stronger observed training reward growth than GRPO. Both training runs use 11.917 equivalent GPU-hours per step.

Solid lines join observations. Dashed extensions illustrate extrapolations.

How much transfers to unfamiliar tasks?

We evaluated OSWorld-trained checkpoints on three target benchmarks without additional target training.

Backbone	ScienceBoard	OpenApps	OfficeWorld
Qwen3.5-9B	6.8 $\rightarrow$ 10.1	51.8 $\rightarrow$ 59.8	6.1 $\rightarrow$ 25.6
GLM-4.6V-Flash	5.8 $\rightarrow$ 10.2	43.8 $\rightarrow$ 45.5	4.5 $\rightarrow$ 21.9
Qwen3-VL-8B	3.8 $\rightarrow$ 17.0	67.0 $\rightarrow$ 58.9	7.1 $\rightarrow$ 24.0

Base $\rightarrow$ OSWorld RL, mean success (%) over four repeats.

All three backbones improve on OfficeWorld and ScienceBoard. OpenApps transfer depends on the backbone.

A separate evaluation on recent models

Model	OSWorld-v1 (%)
Claude Opus 5	77.87
GPT-6 Astra	78.91
Qwen3.8-27B	63.43
GLM53-Flash	76.74
ScaleCUA-GLM-5.3-Flash	**78.47**

All models use the same desktop tools and evaluation framework on 361 tasks, excluding eight Google Drive tasks.

ScaleCUA has a 200-step limit in this comparison. The other models have a 100-step limit.

What made desktop practice scale?

Generate a check alongside each task.
Live execution makes large-scale task generation practical. Human audits still matter because a runnable judge can reward an incorrect outcome.

Spend rollouts where the agent can learn.
Recent successes and failures identify tasks worth revisiting as the agent improves.

Keep the visual context manageable.
Short image windows reduce the cost of training on long desktop workflows.

Project code and resources github.com/THUDM/SCALE-CUA