LabOSWorld: Benchmarking Computer Use Agents towards Scientific Instrument Control

Anonymous ACL submission

Abstract

Current computer-use benchmarks primarily focus on software operation tasks in virtualized systems, whereas scientific instrumentation scenarios require coordinated control over complex interfaces, and feedback-driven parameter adjustment. However, directly evaluating agents on physical high-precision instruments is impractical due to high cost, safety risks, limited accessibility, and difficulty in ensuring reproducible evaluation. This motivates the need for a simulated yet realistic testbed that preserves the operational challenges of scientific instruments while enabling scalable and safe benchmarking. To this end, we introduce **LabOSWorld**, a challenging benchmark for multimodal GUI agents built on a suite of web-based scientific-instrument simulators. Operating directly via a browser, LabOSWorld avoids resource-heavy OS virtualization while supporting flexible task configuration and execution-based evaluation. Specifically, LabOSWorld constructs 96 subtasks across eight instrument simulators, covering workflows from sample loading, alignment, parameter tuning, and data acquisition to result inspection. We evaluate general-purpose vision-language models, specialized GUI agent models, and advanced agentic frameworks at both subtask and end-to-end levels. Our experiments reveal that while existing agents can complete many structured GUI subtasks, they still struggle with feedback-driven operations and long-horizon workflow execution. Overall, LabOSWorld provides a reproducible, low-cost testbed for advancing computer-using agents toward scientific-instrument control.

1 Introduction

Humans perform many computational tasks through graphical user interfaces (GUIs) and command-line interfaces (CLIs), including web browsing, file management, data analysis, and software development. Recent advances in large vision-language models (VLMs) have enabled autonomous digital agents that follow natural-language instructions and interact with GUIs through reasoning-and-acting loops (Yao et al., 2022; Qin et al., 2025), offering a promising path toward simplifying human–computer interaction.

Despite this progress, developing multimodal agents for domain-specific professional settings remains challenging. Existing benchmarks such as *OSWorld* provide full operating-system environments (Xie et al., 2024; Bonatti et al., 2024; Rawles et al., 2025), but rely on heavy virtualization frameworks such as VMware or Docker, making them costly to deploy and difficult to scale. Web-based benchmarks such as *WebArena* and *Mind2Web* focus on general web navigation (Zhou et al., 2024; Deng et al., 2023), and thus do not capture scientific-instrument interfaces with intricate layouts, long-horizon procedures, and continuous parameter tuning. Moreover, many prior benchmarks rely on static demonstrations or offline evaluation, limiting their ability to assess interactive learning, exploration, and alternative valid action sequences.

Scientific instrument interfaces pose unique challenges for multimodal agents. Benchmarking this setting is difficult because it requires realistic procedural workflows, dense professional interfaces, and feedback-driven parameter tuning, while avoiding the cost, safety risks, and limited accessibility of physical instruments. Although recent work has explored scientific agents, autonomous laboratories, and intelligent instrumentation systems (Tom et al., 2024; Szymanski et al., 2023; Boiko et al., 2023; Wang et al., 2022; Jansen et al., 2024; Deng et al., 2026), existing benchmarks provide limited support for evaluating GUI-based instrument control. Prior work mainly focuses on scientific reasoning, experiment planning, autonomous discovery, or instrument-specific intelligence (Tom et al., 2024; Szymanski et al., 2023; Boiko et al., 2023;

Wang et al., 2022; Jansen et al., 2024), whereas our work evaluates existing computer-use agents on scientific-instrument GUIs when direct device APIs are unavailable.

To address this gap, we introduce **LabOSWorld**, a lightweight, web-based, executable benchmark for evaluating multimodal agents on scientific instrument simulators. LabOSWorld runs entirely in a standard browser environment, eliminating the need for OS-level virtualization while preserving realistic GUI interactions. It supports native mouse and keyboard control, configurable initial states, execution-based evaluation.

LabOSWorld includes 96 subtasks across eight instrument simulators, covering workflows such as X-ray diffraction scanning, focused ion beam milling, light/fluorescence microscopy imaging, and scanning probe microscopy operation. We evaluate state-of-the-art LLM/VLM-based agents, specialized GUI models (Qin et al., 2025), and agentic frameworks (Han et al., 2026). Results show that strong multimodal models can complete many structured subtasks, but still struggle with feedback-driven adjustment, scientific-state interpretation, and long-horizon execution. Our analysis further reveals failures in visual grounding, action localization, recovery strategy, and instrument-specific GUI understanding.

Our contributions are threefold: (i) we introduce LabOSWorld, a lightweight executable benchmark for scientific-instrument GUI control; (ii) we design 96 subtasks across eight instrument simulators, covering procedural operation, visual feedback, and scientific-state adjustment; and (iii) we systematically evaluate general-purpose multimodal models, specialized GUI models, and agentic frameworks, revealing substantial gaps in feedback-driven and end-to-end scientific workflows.

2 Related Work

Computer-Using Agents (CUAs). CUAs have evolved from DOM-based web agents to vision-based computer-use agents that operate directly on screenshots (Deng et al., 2023; Zhou et al., 2024; He et al., 2024; Zheng et al., 2024; Xie et al., 2024; Davydova et al., 2025; Rawles et al., 2025; Qin et al., 2025; Liu et al., 2025; Wu et al., 2025; Li et al., 2026). Screen parsing methods (Lu et al., 2024b; Li et al., 2025) improve agents' ability to identify actionable elements in complex visual observations. However, current agents still struggle with long-horizon planning, precise localization, and state-dependent interaction dynamics.

Agents for Science. Scientific-agent systems have studied reasoning, experiment planning, chemistry tool use, machine-learning experimentation, and autonomous discovery (Tom et al., 2024; Szymanski et al., 2023; Boiko et al., 2023; Wang et al., 2022; Jansen et al., 2024; Bran et al., 2023; Huang et al., 2023; Lu et al., 2024a). MyScope[1] provides browser-based microscope simulators for human education. In contrast, LabOSWorld adds executable evaluation for computer-use agents, including natural-language task specifications, subtask-level success checkers, episode logging, browser execution, and instrument-specific metrics.

Computer-Using Benchmarks. Existing benchmarks evaluate agents in web, desktop, mobile, and scientific workflow environments (Deng et al., 2023; Zhou et al., 2024; Koh et al., 2024; Xie et al., 2024; Davydova et al., 2025; Rawles et al., 2025; Yang et al., 2026; Sun et al., 2025). Web-based benchmarks are lightweight but focus mainly on general navigation, while OS-level benchmarks improve realism at the cost of heavy virtualization. More recently, *ScienceBoard* evaluates multimodal autonomous agents in realistic scientific workflows with professional software and multi-domain research tasks (Sun et al., 2025). However, existing benchmarks still provide limited coverage of scientific-instrument GUI operation, where agents must manipulate dense control panels, track instrument states, perform continuous parameter adjustment, and respond to state-dependent visual or numerical feedback. In contrast, LabOSWorld focuses on executable, browser-based scientific instrument simulators for evaluating GUI-based instrument control when direct device APIs are unavailable.

3 The LabOSWorld Benchmark

We introduce LabOSWorld, a benchmark suite for evaluating computer-use agents on scientific-instrument interfaces. Unlike general web or desktop tasks, scientific-instrument control requires procedural dependencies, domain-specific controls, state transitions, and ordered acquisition or export operations. LabOSWorld complements general

[1] https://myscope.training/

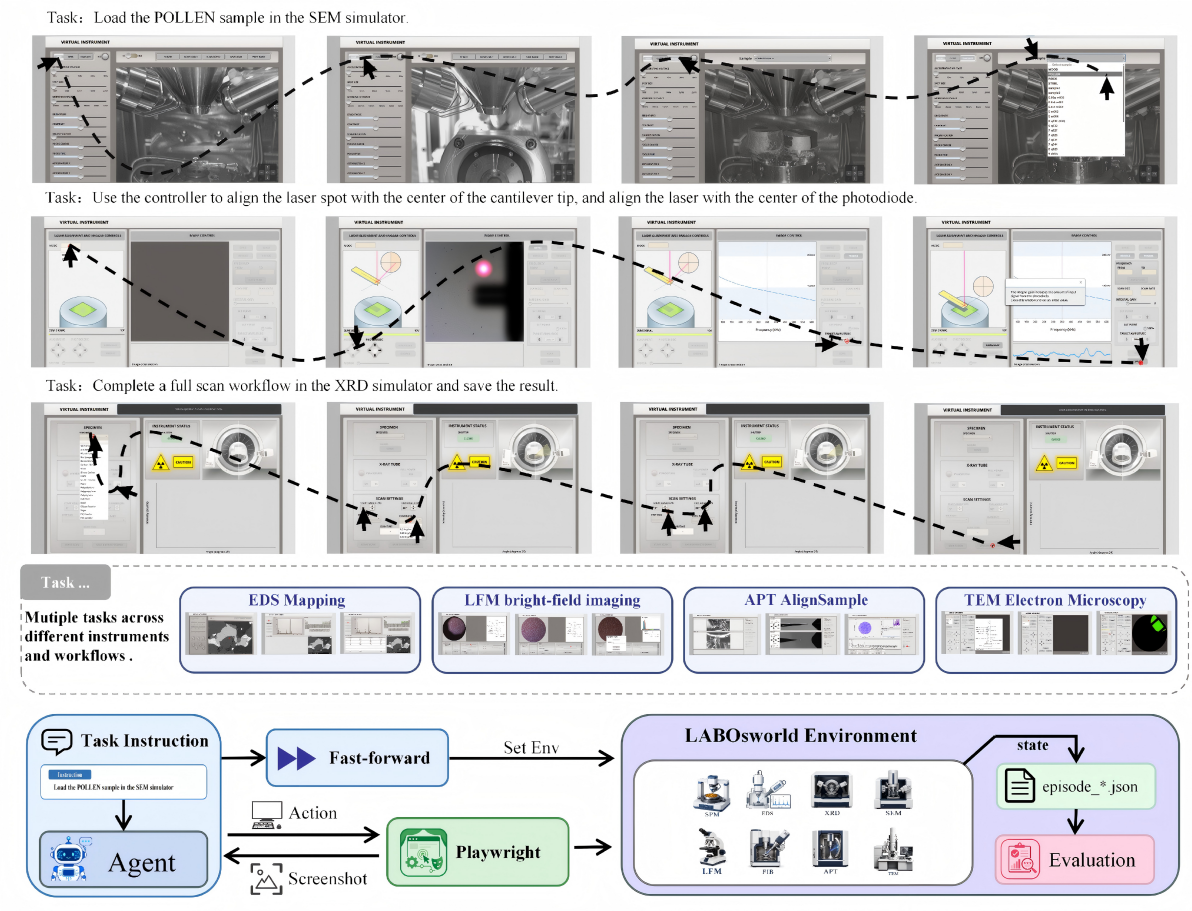


Figure 1: Overview of LabOSWorld. The benchmark evaluates computer use agents on eight types of scientific instrument web interfaces, where agents receive natural-language instructions and screenshots, execute GUI actions, and are scored using episode-level and subtask-level metrics across 96 subtasks.

computer-use benchmarks (Xie et al., 2024; Zhou et al., 2024; Deng et al., 2023; Rawles et al., 2025) with domain-specific web interfaces and laboratory-style workflows.

Each run is launched from a lightweight specification consisting of command-line arguments, environment variables, and a natural-language instruction. As shown in Fig. 2, a browser-backed coordinator executes agent actions, exports episode logs through in-page benchmark hooks, and aggregates instrument-specific metrics.

3.1 Scientific Instrument Tasks

LabOSWorld covers eight simulated scientific instruments that span three foundation experimental paradigms: (i) advanced microscopy imaging, including Scanning Electron Microscopy (**SEM**), Transmission Electron Microscopy (**TEM**), and Light/Fluorescence Microscopy (**LFM**); (ii) Diffraction and Spectroscopic Analysis, including X-Ray Diffraction (**XRD**), Energy-Dispersive Spectroscopy (**EDS**) and Atom Probe Tomography (APT); (iii) precision micro-nanofacrication and manipulation, including Focused Ion Beam (**FIB**) and Scanning Probe Microscopy (**SPM**). Each instrument is implemented as a high-fidelity, interactive web environment that meticulously preserves domain-specific control modalities, such as multi-dimensional sliders, discrete toggles, real-time dynamic plots, and multi-state status indicators.

Each task within our benchmark is operationalized via a natural-language instruction describing the target scientific operation. Rather than navigating standardized, linear web forms, agents in LabOSWorld must master highly heterogeneous, instrument-specific operational logic. For example, XRD tasks require agents to select a specimen, configure scan parameters, run the scan, and save the diffraction result, while SEM tasks involve chamber control, sample selection, vacuum preparation, high-voltage activation, imaging adjustment, scanning, and image saving. Consequently, successfully

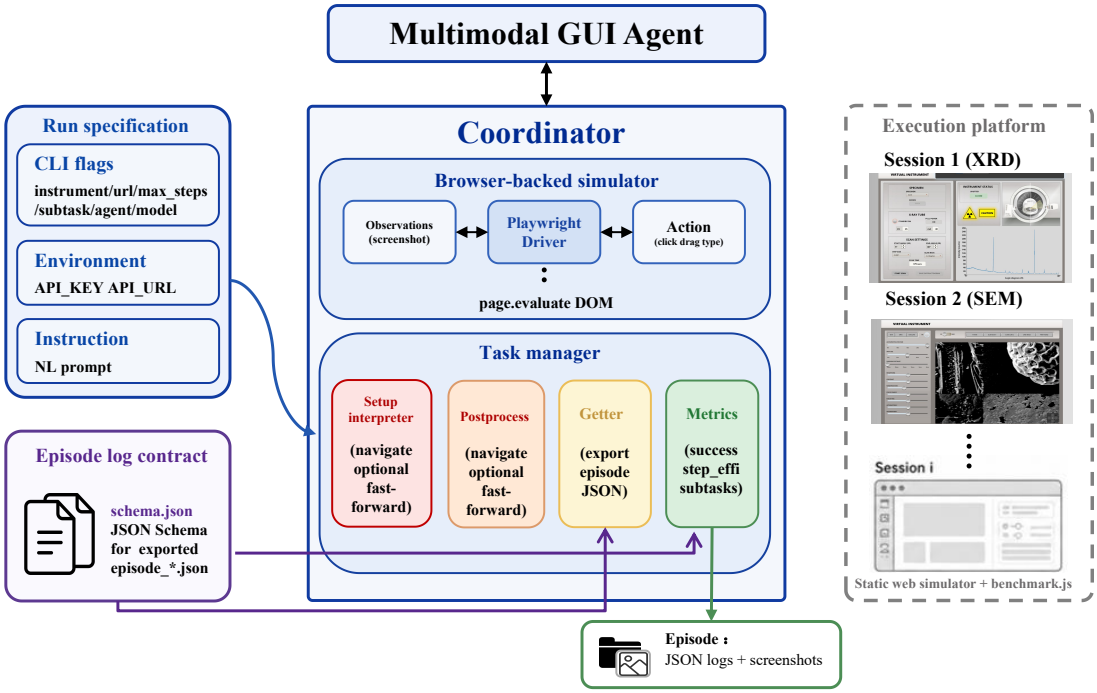


Figure 2: Evaluation infrastructure of LabOSWorld. Each run is specified by command-line arguments, environment variables, and a natural-language instruction. A browser-backed coordinator executes agent actions in scientific instrument web simulators, exports episode logs through in-page benchmark hooks, and aggregates metrics according to instrument-specific JSON Schemas.

solving these tasks demands an integration of fine-grained cross-modal visual-grounding, non-linear procedural reasoning under physical dependencies, and robust state tracking across extended interaction horizons.

3.2 Workflow and Subtask Decomposition

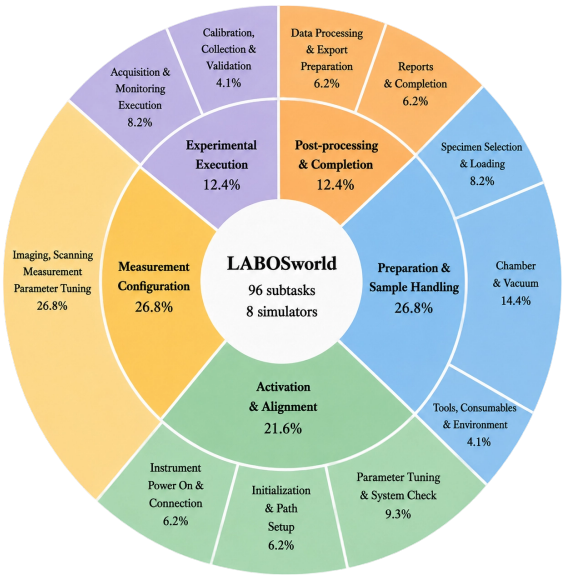
To facilitate structured evaluation, we structure the workflow of each instrument as a sequence of subtasks. Each subtask corresponds to a meaningful operation stage detectable from simulator states or interactions with specific GUI controls. This decomposition follows actual scientific operating procedures rather than arbitrary or heuristic screen-region segmentations.

Across instruments, subtasks fall into recurring structural categories: (i) *Sample preparation* (e.g., specimen selection and mounting); (ii) *Environment conditioning* (e.g., vacuum evacuation); (iii) *Power or beam activation* (e.g., power-up or beam alignment); (iv) *Parameter configuration* (e.g., focus adjustment and astigmatism correction); (v) *Data acquisition* (e.g., sample scanning and signal capturing); This design enables interpretable evaluation beyond binary task success and supports targeted diagnosis of failures at specific workflow stages.

3.3 Browser-Based Evaluation Infrastructure

Unlike traditional environments that depend on heavy virtual machines, LabOSWorld runs natively in a browser. We use Playwright to open simulator pages, capture screenshots, and execute agent actions. At each step, the agent receives the instruction and current screenshot, then outputs an executable GUI action. The action space covers clicking, double-clicking, dragging, typing, selecting dropdown options, pressing keys, scrolling, and waiting for simulator state transitions. The browser driver parses and executes these actions, using DOM-level event dispatch when possible and mouse events otherwise for JavaScript-based controls.

Each simulator page is instrumented with an in-page benchmark script, such as `benchmark_xrd.js` or `benchmark_sem.js`. These scripts record subtask completion, control attempts, episode metadata, and step-level traces.



Inst.	#Sub.	Goal
XRD	8	Diffractogram
SEM	12	Imaging
TEM	10	Imaging
FIB	20	Preparation
SPM	14	Scanning
LFM	12	Brightfield
EDS	8	Spectrum
APT	12	Reconstruction
Total	**96**	–

Figure 3: Overview of scientific-instrument workflows and GUI task taxonomy in LabOSWorld. The left panel summarizes the distribution of 96 subtasks across high-level and fine-grained GUI operation categories, while the right panel lists the eight instrument simulators, their subtask counts, and workflow goals.

Finally, a Python-based coordinator exports the in-page episode object and step screenshots, following instrument-specific JSON Schemas with fields such as `success`, `summary`, `subtasks`, and `steps`.

3.4 Full-Episode and Subtask-Level Evaluation

We evaluate agents in two complementary modes. In the full-episode setting, the agent starts from the initial simulator state and must complete the entire workflow within a pre-defined step budget. This setting measures long-horizon task completion and captures compounding failures caused by early mistakes, missed state transitions, or incorrect parameter settings.

In the subtask-level setting, the simulator is initialized directly to the canonical state immediately preceding a target subtask, and the agent is evaluated solely on that individual subtask. This mode isolates local capabilities such as locating a domain-specific widget, selecting a menu item, dragging a parameter slider, or recognizing operation completion. For simulators that support programmatic state initialization, such as XRD, we implement fast-forward functions exposed on the browser window, e.g., `XRD_fast_forward_to_subtask(`S_k`)`. These functions mark preceding subtasks as completed and position the UI at the beginning of the target subtask. Such fast-forwarding is utilized exclusively for diagnostic subtask evaluation and is disabled during full-episode evaluation.

Together, the two settings distinguish local GUI grounding failures from long-horizon workflow failures. An agent that succeeds on subtasks but fails in full episodes may suffer from limitations in memory, sequential ordering, or error recovery, whereas failure on individual subtasks indicates weak grounding of domain-specific scientific controls.

3.5 Metrics

We report episode-level and subtask-level metrics. An episode is successful if all required subtasks are completed within the step budget. For each subtask, we record whether the corresponding operation is completed, revealing where an agent fails in full episodes and enabling localized success-rate analysis.

Let $\mathcal{S}_d$ denote the set of subtasks associated with instrument d, and let $R_{d,s}$ denote the number of repeated runs available for subtask s. In our main subtask-level evaluation, $R_{d,s} = 2$. For each subtask $s \in \mathcal{S}_d$, we compute:

$$\mathrm{SR}_{d,s} = \frac{1}{R_{d,s}} \sum_{r=1}^{R_{d,s}} \mathbf{1}\left[\mathrm{success}_{d,s,r}\right],$$

where $\mathrm{success}_{d,s,r}$ indicates whether the agent successfully completes subtask s of instrument d in run r.

The instrument-level score is then computed by

averaging over all subtasks:

$$\mathrm{Score}_d = \frac{1}{|\mathcal{S}_d|} \sum_{s \in \mathcal{S}_d} \mathrm{SR}_{d,s}.$$

The reported results in Table 1 correspond to Score_d for each instrument. We also log interaction attempts and grounding-oriented statistics for diagnostic analysis. For image-producing tasks, we compute pixel-level quality metrics when a reference image or canonical view is available. In SEM, the saved or current micrograph is compared against a reference view after display-region normalization. We report peak signal-to-noise ratio (PSNR) to measure the visual fidelity of the resulting micrograph. This metric assesses whether parameter adjustments produce a scientifically usable visual state beyond merely triggering the correct GUI action.

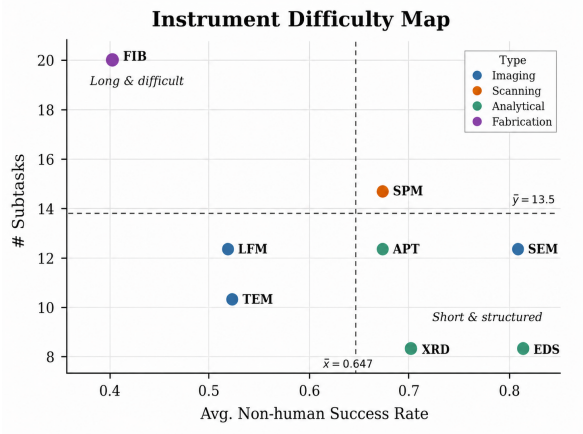


Figure 4: Instrument difficulty map of the eight scientific-instrument simulators in LabOSWorld. The x-axis shows the average non-human success rate, the y-axis shows the number of subtasks, and bubble size denotes the human–model gap. Dashed lines indicate instrument-level means.

4 Experiments

In this section, we present the experimental settings and main results for representative GUI agent baselines on LabOSWorld, including general-purpose multimodal models, specialized GUI models, and agentic frameworks.

4.1 Baseline Models and Agentic Frameworks

We evaluate three groups of baselines on LabOSWorld. The first group includes general-purpose multimodal models, including Qwen3VL-32B (Bai et al., 2025), EvoCUA-8B (Xue et al., 2026), Claude Sonnet-4.5 (Anthropic, 2025b), Kimi-K2.5 (Kimi Team et al., 2026), Seed-1.6 (ByteDance Seed Team, 2025), GPT-5.5 (OpenAI, 2026), and Claude Opus-4.5 (Anthropic, 2025a). The second group consists of specialized GUI models, including UI-TARS-1.5-7B (Qin et al., 2025) and GUI-Owl-7B (Deng et al., 2026). The third group contains agentic frameworks built on strong foundation models, including GTA1 w/ GPT-5.5, VLAA-GUI w/ Opus-4.5 (Han et al., 2026), and Hippo Agent w/ Opus-4.5.

For each task, the agent receives a natural-language instruction and the current screen observation, and then outputs an executable GUI action, such as clicking, typing, scrolling, selecting a widget, or triggering an instrument-specific operation. We use a unified interaction protocol across all instruments and models. For subtask-level evaluation, each model is given at most 50 interaction steps per subtask. Unless otherwise specified, we run each model–subtask pair twice (`runs=2`) and report subtask-normalized success rates.

For each instrument d, we first compute the success rate of each subtask over repeated runs and then average across all subtasks, ensuring that each subtask contributes equally. The instrument-level scores reported in Table 1 correspond to this subtask-normalized score.

Benchmark-level difficulty. Before comparing individual models, we first analyze the structural difficulty of LabOSWorld at the instrument level. Figure 4 visualizes each simulator according to its average non-human model success rate, workflow length, and human–best model gap. This view shows that the difficulty of scientific-instrument GUI control is multi-dimensional. FIB is challenging because it combines the longest workflow with low average model performance, indicating strong long-horizon dependency and error-accumulation pressure. LFM has a shorter workflow than FIB but a larger human–model gap, suggesting that feedback-driven optical adjustment and visual-state interpretation remain difficult for current agents. In contrast, EDS and XRD are located in the short-workflow and high-performance region, reflecting their more structured operation sequences.

4.2 Results and Analysis

Category-level analysis. Table 2 reveals distinct strengths and limitations across method families. General-purpose multimodal models benefit from

Table 1: Main results of average subtask-level scores on the LabOSWorld benchmark across eight scientific instrument simulators. Scores are reported in the range $[0, 1]$. The Avg. column reports the macro-average over the eight instrument simulators. Bold values indicate the best non-human method for each column.

Category	Model	SEM	SPM	TEM	XRD	LFM	FIB	APT	EDS	Avg.
General Models	Qwen3VL-32B	0.833	0.643	0.500	0.750	0.500	0.275	0.667	0.750	0.615
	EvoCUA-8B	0.833	0.571	0.500	0.750	0.500	0.350	0.708	0.562	0.597
	Claude Sonnet-4.5	0.833	0.714	0.400	0.625	0.500	0.250	0.667	**1.000**	0.624
	Kimi-K2.5	0.625	0.357	0.500	**0.875**	0.667	0.525	0.667	0.562	0.597
	Seed-1.6	0.792	0.786	0.650	**0.875**	0.667	0.650	0.875	0.812	0.763
	GPT-5.5	0.833	0.786	0.500	0.625	0.500	0.650	**0.917**	**1.000**	0.726
	Claude Opus-4.5	0.833	0.643	0.400	0.750	0.500	0.250	0.583	**1.000**	0.620
Specialized Models	UI-TARS-1.5-7B	0.500	0.750	0.600	0.812	0.542	0.650	0.667	0.750	0.659
	GUI-Owl-7B	0.833	0.714	0.500	0.750	0.333	0.325	0.667	**1.000**	0.640
Agentic Frameworks	GTA1 w/ GPT-5.5	**0.875**	**0.821**	**0.850**	**0.875**	**0.708**	**0.675**	0.833	0.875	**0.814**
	VLAA-GUI w/ Opus-4.5	0.333	0.429	0.400	0.625	0.333	0.150	0.500	0.875	0.456
	Hippo Agent w/ Opus-4.5	0.833	0.714	0.500	0.750	0.583	0.300	0.583	**1.000**	0.658
Human	Human	0.917	0.857	0.900	1.000	0.917	0.800	1.000	1.000	0.924

strong visual perception and instruction following: Seed-1.6 performs well on experimental execution and post-processing, while GPT-5.5 is strong in measurement configuration and execution. This suggests that general VLMs can handle instruction understanding, visual feedback, and parameter selection, but still struggle with long-horizon tracking and fine-grained closed-loop adjustment.

Specialized GUI models show advantages in conventional interface grounding. UI-TARS-1.5-7B and GUI-Owl-7B are competitive on widget selection, panel operation, and structured GUI procedures. However, their gains are less consistent for tasks requiring scientific-state interpretation, such as alignment, focusing, or visually guided adjustment, indicating that GUI localization alone is insufficient for scientific-instrument control.

Agentic frameworks achieve the strongest overall results, but their benefits depend on reliable grounding and domain-valid recovery. GTA1 w/ GPT-5.5 obtains the best overall average and leads on Activation & Alignment and Measurement Configuration, suggesting that planning, verification, and execution monitoring can improve complex workflows. In contrast, the weaker performance of VLAA-GUI w/ Opus-4.5 shows that simply adding an agent loop is not enough: wrong clicks, incorrect control adjustments, and invalid recovery actions can still derail scientific procedures.

Scientific-state understanding. Table 2 suggests that scientific-instrument GUI control cannot be reduced to discrete widget grounding. Tasks involving alignment, focusing, and image adjustment require agents to interpret scientific visual states and perform feedback-driven control. We therefore analyze the SEM focus-adjustment task, where the agent must iteratively adjust focus and judge image clarity. As shown in Fig. 6, GPT-5.5 achieves both the highest focus success rate and the highest best PSNR, showing that LabOSWorld can evaluate intermediate scientific-state quality rather than only whether a task is eventually completed.

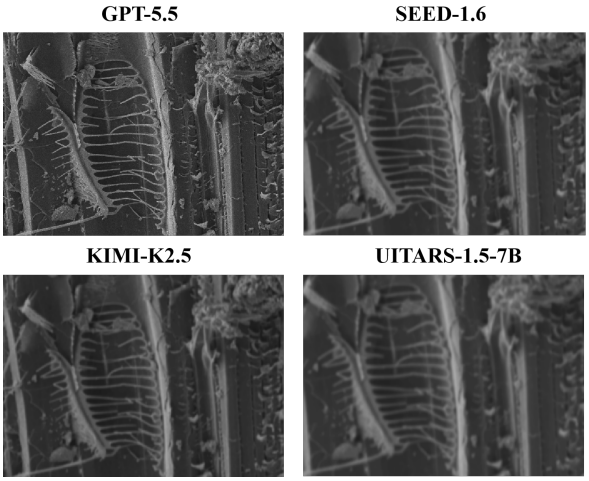


Figure 5: Qualitative comparison of final outputs on the SEM focus-adjustment task under the same task setting.

End-to-end workflow difficulty. Although our main evaluation is subtask-level, we further perform an end-to-end pilot evaluation using GPT-5.5 with 10 runs per workflow. GPT-5.5 achieves non-zero success on EDS, APT, and XRD, with success rates of 70.0%, 80.0%, and 60.0%, respectively, but fails on SEM, SPM, TEM, LFM, and FIB. The average end-to-end success rate is only 26.3%, much lower than its subtask-level performance, indicating severe error accumulation in complete work-

Table 2: Success rates across five scientific-instrument task categories. Scores are reported as percentages. Each category-level score is computed by averaging subtask-level success rates over all operation-level subtasks assigned to that category.

Model	Preparation & Sample Handling	Activation & Alignment	Measurement Configuration	Experimental Execution	Post-processing & Completion
Qwen3VL-32B	82.4	38.6	61.1	46.4	62.5
Hippo Agent w/ Opus-4.5	88.2	43.2	66.7	42.9	65.6
UI-TARS-1.5-7B	91.2	40.9	64.8	64.3	71.9
GUI-Owl-7B	85.3	34.1	64.8	50.0	68.8
EvoCUA-8B	85.3	40.9	64.8	50.0	43.8
Kimi-K2.5	76.5	45.5	53.7	57.1	62.5
Seed-1.6	**97.1**	59.1	66.7	**82.1**	**81.2**
VLAA-GUI w/ Opus-4.5	82.4	18.2	29.6	28.6	56.2
Claude Sonnet-4.5	76.5	45.5	63.0	35.7	68.8
Claude Opus-4.5	82.4	40.9	63.0	35.7	62.5
GTA1 w/ GPT-5.5	94.1	**63.6**	**87.0**	78.6	75.0
GPT-5.5	82.4	50.0	81.5	71.4	75.0

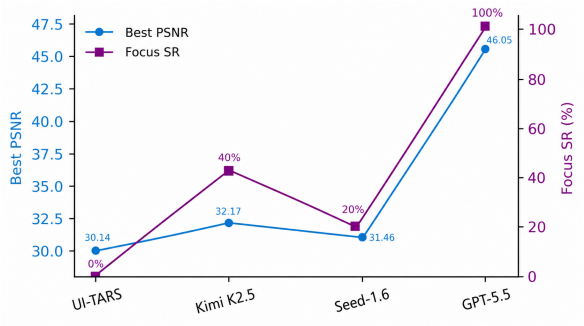


Figure 6: Scientific-state quality on the SEM focus-adjustment task. The two lines show best PSNR and focus success rate (SR) across models. GPT-5.5 achieves the highest values on both metrics.

flows.

Instrument-level difficulty. Performance varies substantially across instruments. EDS and XRD are relatively easier due to structured operations, clear feedback, and short action sequences. In contrast, SEM, FIB, and LFM are more challenging because they involve feedback-driven focusing, image-quality adjustment, long procedures, fine-grained operations, and visual target localization. These differences show that LabOSWorld evaluates not only generic GUI navigation, but also visual grounding, scientific-domain understanding, long-horizon reasoning, and closed-loop control.

Summary. The results highlight four findings. First, subtask-level evaluation reveals limitations of current GUI agents, while end-to-end evaluation exposes severe error accumulation. Second, general-purpose multimodal models outperform specialized GUI models overall, but still struggle with feedback-intensive and long-horizon tasks. Third, agentic frameworks help only when their recovery strategies align with scientific-instrument operation semantics. Fourth, LabOSWorld challenges agents on GUI grounding, scientific-state understanding, sequential operation, and continuous visual adjustment.

5 Conclusion

We presented LabOSWorld, a lightweight executable benchmark for scientific-instrument GUI control built on web-based simulators. By avoiding full OS virtualization while preserving dense interfaces, procedural dependencies, continuous parameter tuning, and feedback-driven visual adjustment, LabOSWorld provides a practical testbed for evaluating multimodal GUI agents in scientific workflows. Our evaluation shows that current agents remain far from reliable scientific-instrument operation: strong multimodal models can solve many discrete subtasks, and agentic frameworks can improve robustness, but failures persist in scientific-state interpretation, precise spatial grounding, closed-loop adjustment, and long-horizon workflows. The SEM focus analysis and end-to-end study further demonstrate that successful agents must not only complete GUI actions, but also maintain valid intermediate scientific states and avoid error accumulation. Overall, LabOSWorld highlights scientific-instrument control as a challenging and underexplored setting that connects computer-use agent evaluation (Xie et al., 2024; Zhou et al., 2024; Qin et al., 2025) with autonomous laboratory systems (Tom et al., 2024; Szymanski et al., 2023; Boiko et al., 2023), and we hope it encourages future work on domain-aware GUI agents with accurate grounding, state-aware planning, and scientific feedback interpretation.

Limitations

Although LabOSWorld provides a lightweight and executable testbed for scientific-instrument GUI control, it still has several limitations. First, it is built on web-based simulators rather than physical instruments, so it cannot fully capture hardware latency, calibration uncertainty, safety constraints, or real laboratory failure modes. Second, the current benchmark covers a selected set of instruments and workflows, including microscopy, spectroscopy, diffraction, and tomography, but does not yet include broader laboratory scenarios such as wet-lab protocols, robotic manipulation, chemical synthesis, or multi-instrument experimental planning. Finally, the evaluation mainly relies on screenshots and logged simulator states, while some scientific decisions may require richer observations, domain knowledge, or multimodal sensor signals beyond the current environment.

LabOSWorld is designed for evaluating agents in simulated scientific-instrument environments and does not provide direct control over real laboratory hardware. Nevertheless, scientific-instrument automation may involve safety-critical operations if deployed on physical devices. Future systems should include human supervision, permission control, operation logging, and safety interlocks before being connected to real instruments. The benchmark is intended to support reproducible research on GUI agents and should not be interpreted as a recommendation to deploy autonomous agents on physical laboratory equipment without appropriate safety validation.

References

Anthropic. 2025a. Claude opus 4.5 system card. https://www.anthropic.com/claude-opus-4-5-system-card. Official system card. Accessed: 2026-05-22.

Anthropic. 2025b. Claude sonnet 4.5 system card. https://www.anthropic.com/claude-sonnet-4-5-system-card. Official system card. Accessed: 2026-05-22.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578.

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, and 1 others. 2024. Windows agent arena: Evaluating multi-modal os agents at scale. *arXiv preprint arXiv:2409.08264*.

Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2023. ChemCrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*.

ByteDance Seed Team. 2025. Seed1.6: Bytedance seed general-purpose model series. https://seed.bytedance.com/en/seed1_6. Official technical page. Accessed: 2026-05-22.

Mariya Davydova, Daniel Jeffries, Patrick Barker, Arturo Márquez Flores, and Sinéad Ryan. 2025. Osuniverse: Benchmark for multimodal gui-navigation ai agents. *arXiv preprint arXiv:2505.03570*.

Han Deng, Anqi Zou, Hanling Zhang, Ben Fei, Chengyu Zhang, Haobo Wang, Xinru Guo, Zhenyu Li, Xuzhu Wang, Peng Yang, and 1 others. 2026. Owl-auraid 1.0: An intelligent system for autonomous scientific instrumentation and scientific data analysis. *arXiv preprint arXiv:2603.29828*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2Web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Qijun Han, Haoqin Tu, Zijun Wang, Haoyue Dai, Yiyang Zhou, Nancy Lau, Alvaro A Cardenas, Yuhui Xu, Ran Xu, Caiming Xiong, and 1 others. 2026. VLAA-GUI: Knowing when to stop, recover, and search, a modular framework for GUI automation. *arXiv preprint arXiv:2604.21375*.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2023. MLAgentBench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. 2024. DiscoveryWorld: A virtual environment for developing and evaluating automated scientific discovery agents. *Advances in Neural Information Processing Systems*, 37:10088–10116.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng

Chen, Guanduo Chen, and 1 others. 2026. Kimi K2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. 2025. ScreenSpot-Pro: GUI grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 8778–8786.

Zehao Li, Zhenyu Wu, Yibo Zhao, Bowen Yang, Jingjing Xie, Zhaoyang Liu, Zhoumianze Liu, Kaiming Jin, Jianze Liang, Zonglin Li, and 1 others. 2026. Os-themis: A scalable critic framework for generalist gui rewards. *arXiv preprint arXiv:2603.19191*.

Zhaoyang Liu, JingJing Xie, Zichen Ding, Zehao Li, Bowen Yang, Zhenyu Wu, Xuehui Wang, Qiushi Sun, Shi Liu, Weiyun Wang, and 1 others. 2025. Scalecua: Scaling open-source computer use agents with cross-platform data. *arXiv preprint arXiv:2509.15221*.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024a. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. 2024b. OmniParser for pure vision based GUI agent. *arXiv preprint arXiv:2408.00203*.

OpenAI. 2026. Gpt-5.5 system card. https://openai.com/index/gpt-5-5-system-card/. Official system card. Accessed: 2026-05-22.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, and 1 others. 2025. UI-TARS: Pioneering automated GUI interaction with native agents. *arXiv preprint arXiv:2501.12326*.

Chris Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, and 1 others. 2025. Androidworld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, volume 2025, pages 406–441.

Qiushi Sun, Zhoumianze Liu, Chang Ma, Zichen Ding, Fangzhi Xu, Zhangyue Yin, Haiteng Zhao, Zhenyu Wu, Kanzhi Cheng, Zhaoyang Liu, and 1 others. 2025. Scienceboard: Evaluating multimodal autonomous agents in realistic scientific workflows. *arXiv preprint arXiv:2505.19897*.

Nathan J Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E Kumar, Tanjin He, David Milsted, Matthew J McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, and 1 others. 2023. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*, 624(7990):86.

Gary Tom, Stefan P Schmid, Sterling G Baird, Yang Cao, Kourosh Darvish, Han Hao, Stanley Lo, Sergio Pablo-García, Ella M Rajaonson, Marta Skreta, and 1 others. 2024. Self-driving laboratories for chemistry and materials science. *Chemical Reviews*, 124(16):9633–9732.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298.

Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Qiushi Sun, Zhaoyang Liu, Zhoumianze Liu, Yu Qiao, Xiangyu Yue, Zun Wang, and 1 others. 2025. Os-oracle: A comprehensive framework for cross-platform gui critic models. *arXiv preprint arXiv:2512.16295*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, and 1 others. 2024. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.

Taofeng Xue, Chong Peng, Mianqiu Huang, Linsen Guo, Tiancheng Han, Haozhe Wang, Jianing Wang, Xiaocheng Zhang, Xin Yang, Dengchang Zhao, and 1 others. 2026. EvoCUA: Evolving computer use agents via learning from scalable synthetic experience. *arXiv preprint arXiv:2601.15876*.

Pei Yang, Hai Ci, and Mike Zheng Shou. 2026. macOSWorld: A multilingual interactive benchmark for GUI agents. *Advances in Neural Information Processing Systems*, 38:134014–134056.

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

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. GPT-4V(ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. 2024. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pages 15585–15606.

A Benchmark Task Details

This appendix provides the complete list of 96 subtasks used in LabOSWorld. Each scientific-instrument simulator is decomposed into a sequence of operation-level subtasks. For each subtask, we report its name, difficulty level, and natural-language description. The difficulty levels are assigned according to the type of GUI interaction, the degree of scientific-state interpretation required, and the amount of feedback-driven adjustment involved.

A.1 Task Construction Principles

The subtasks in LabOSWorld are constructed according to scientific workflow stages rather than arbitrary GUI regions. Each subtask corresponds to an operation-level unit that changes the simulator state, configures an instrument parameter, performs a scientific action, or finalizes an output. This design preserves the scientific meaning of the original workflow while enabling diagnostic subtask-level evaluation.

A.2 Complete Subtask List

The following eight tables summarize the complete subtask lists for all scientific-instrument simulators. These subtasks are used for the subtask-level evaluation reported in the main paper.

A.3 Difficulty Level Definitions

We annotate each subtask with a coarse difficulty level to characterize the interaction complexity. EASY subtasks typically involve a single explicit GUI action, such as clicking a clearly labeled button or selecting an item from a dropdown menu. MEDIUM subtasks require a short sequence of actions, parameter adjustment, or waiting for a simulator state transition. HARD subtasks require feedback-driven control, precise spatial grounding, scientific-state interpretation, or multi-step manipulation under visual feedback.

The difficulty labels are intended for analysis and interpretation rather than for changing the evaluation protocol. All subtasks are evaluated using the same action interface and success-checking mechanism.

A.4 Mapping to High-Level Task Categories

For category-level analysis, each subtask is further mapped to one of five high-level operation categories: Preparation & Sample Handling, Activation & Alignment, Measurement Configuration, Experimental Execution, and Post-processing & Completion. The mapping is based on the primary scientific operation required by the subtask rather than the low-level GUI action. For example, a click on a button may be categorized differently depending on whether it loads a sample, activates a beam, starts a scan, or saves a result. Table 11 defines the five high-level categories and lists the operation-level subtasks assigned to each category.

B Evaluation Protocol and Implementation Details

This appendix provides additional details about the evaluation protocol and implementation of LabOSWorld. The goal is to make the browser-based execution process, action interface, subtask initialization, step budget, and logging format explicit. All models and agentic frameworks are evaluated through the same interaction protocol unless otherwise specified.

B.1 Observation and Action Space

At each interaction step, the agent receives a natural-language task instruction and the current screenshot of the simulator interface. The agent then outputs one executable GUI action. We use a unified action interface across all simulators so that the same evaluation runner can be applied to different scientific-instrument workflows.

The action space includes clicking, double-clicking, dragging, typing, selecting dropdown options, pressing keys, scrolling, and waiting for simulator state transitions. Click and double-click actions are used for buttons, panels, and visual target regions; drag actions support sliders, stage movement, and pattern placement; type and select actions handle numeric inputs and menus; wait actions allow instrument animations, scans, reconstructions, or state transitions to complete.

For coordinate-based actions, the runner maps the agent-produced coordinate to the current browser viewport. When an action targets a standard HTML element, the runner uses DOM-level event dispatch when possible. For canvas-like or JavaScript-rendered controls, it falls back to mouse events. This hybrid execution strategy allows LabOSWorld to support both conventional web widgets and simulator-specific interactive controls.

B.2 Interaction Loop

Each evaluation episode follows a closed-loop observe–act–execute protocol. At the beginning

Table 3: XRD subtasks.

Subtask	Level	Description
1. Select Specimen	EASY	Select a specimen from the dropdown menu.
2. Load Sample	MEDIUM	Click DOORS to open the chamber, load the sample, and then close the chamber.
3. Power Up	MEDIUM	Click STANDBY to power up the instrument and wait until it is ready.
4. Set Angles	MEDIUM	Use the angle adjustment buttons to set the start and end angles.
5. Set Step Size	MEDIUM	Select the step size from the dropdown menu.
6. Set Scan Rate	MEDIUM	Select the scan rate from the dropdown menu.
7. Run Scan	EASY	Click START SCAN to begin scanning.
8. Save Result	EASY	Click SAVE DIFFRACTOGRAM to save the scan result.

Table 4: FIB subtasks.

Subtask	Level	Description
1. Vent Chamber	EASY	Click VENT once and wait for the chamber animation to complete.
2. Pump Down	EASY	Click PUMP once and wait for the vacuum process to complete.
3. Select Sample	EASY	Select Si Wafer from the SAMPLE dropdown menu.
4. E-beam On	EASY	Click the electron-beam HT switch.
5. E-beam Live Focus	MEDIUM	Focus the electron-beam image.
6. WD 7 mm	MEDIUM	Adjust the WD slider to approximately 7 mm.
7. Tilt 10°	MEDIUM	Set the stage tilt to approximately 10°.
8. Stage Z Center	MEDIUM	Move along the Z axis to a suitable height.
9. Ion Beam Live Center	MEDIUM	Center the target region in the ion-beam field of view.
10. First Rect Start	EASY	Click START once on the correctly selected pattern.
11. Delete Pattern	EASY	Click DELETE PATTERN once to remove the current pattern.
12. Second Rect Start	MEDIUM	Select the second rectangle and click START.
13. Beam Current 10 pA	MEDIUM	Set the beam current and use DELETE PATTERN.
14. Pt Needle In	EASY	Click to extend the Pt needle.
15. Pt Deposition Start	HARD	Select the Pt Deposition pattern, place it at the correct position, and click START.
16. Ion Snapshot $5000\times$	MEDIUM	Set the magnification to approximately $5000\times$ and capture an ion-beam snapshot.
17. Cross Section Cut Start	HARD	Select the cross-section cutting pattern and click in the correct region.
18. Cleaning Section Start	HARD	Select the cutting pattern, set the beam current to 0.1 nA, drag the pattern to the target region, and click START.
19. Tilt 0°	EASY	Check Cross Section for imaging and set the stage tilt to 0°.
20. Task Complete	EASY	Click SAVE to store the cut image.

of an episode, the runner launches the target simulator page, applies the required initial state, and provides the agent with the task instruction and the initial screenshot. At each step, the agent predicts the next GUI action based on the instruction and the current observation. The runner parses the action, executes it in the browser, waits for the simulator state to update, captures a new screenshot, and queries the in-page benchmark state.

The loop terminates when the target subtask or workflow is successfully completed, when the step budget is exhausted, or when the runner detects an unrecoverable execution error. Invalid or unparsable actions are recorded in the episode log and counted as failed interaction attempts. This protocol allows the benchmark to capture both successful task completion and intermediate failure behavior, such as repeated clicks, wrong coordinate selection, ineffective waiting, or incorrect parameter manipulation.

B.3 Subtask Initialization

For subtask-level evaluation, each simulator is initialized to a canonical state immediately before the target subtask. This initialization isolates the local capability required by the target operation and avoids conflating subtask performance with failures accumulated in earlier workflow stages. For example, when evaluating a parameter-setting subtask, the simulator is first placed into the state where the relevant control is available and all prerequisite subtasks have already been completed.

When a simulator supports programmatic state control, we use instrument-specific fast-forward functions exposed through the browser window. These functions mark prerequisite subtasks as completed and restore the GUI to the canonical pre-subtask state. Fast-forwarding is used only for diagnostic subtask-level evaluation. It is not used in full-episode evaluation, where the agent must complete the entire workflow from the initial simulator state.

This design allows LabOSWorld to distinguish local GUI-control ability from long-horizon workflow execution. An agent that succeeds under subtask initialization but fails in full-episode evaluation may suffer from planning, ordering, memory, or recovery problems. In contrast, failure under

Table 5: LFM subtasks.

Subtask	Level	Description
1. Drag Sample	EASY	Drag the H&E-stained kidney specimen onto the microscope stage.
2. Select Brightfield	EASY	Check the `SETUP & BRIGHTFIELD` mode.
3. Select Halogen	EASY	Click `HALOGEN LAMP` to select the halogen lamp.
4. Select 10× Focus	MEDIUM	Select the 10× objective and use the `FOCUS` slider.
5. Field Diaphragm Close	EASY	Push the slider fully right to close the field diaphragm.
6. Field Diaphragm Focus	HARD	Adjust the slider so that the diaphragm edge is sharply focused.
7. Field Diaphragm Center	HARD	Use the up, down, left, and right buttons to center the diaphragm.
8. Field Diaphragm Open	EASY	Push the slider left to open the diaphragm until the blades just exceed the field of view.
9. Aperture Diaphragm	HARD	Click to remove the right eyepiece, adjust the aperture diaphragm, and reinstall the eyepiece.
10. White Balance	HARD	Click the target, move to a blank area, click to set white balance, and then move back.
11. Exposure Adjust	MEDIUM	Adjust the exposure slider for optimal exposure and click `LUT`.
12. BF Capture	EASY	Click `LUT` to check the dynamic range and then click `CAPTURE`.

Table 6: SEM subtasks.

Subtask	Level	Description
1. Vent Chamber	EASY	Click `VENT` to release vacuum and allow chamber access.
2. Open Chamber	EASY	Open the chamber door to load the sample.
3. Close Chamber	MEDIUM	Close the chamber door after loading.
4. Evacuate Chamber	EASY	Click `PUMP` to evacuate the chamber.
5. Select Sample	EASY	Select the specimen from the SAMPLE dropdown menu.
6. Turn On HT	MEDIUM	Click the `HT` button to turn on the electron beam.
7. Set Acc Voltage	HARD	Set the acceleration voltage from the dropdown menu.
8. Set Contrast	HARD	Adjust the contrast slider for optimal image quality.
9. Adjust Clarity	HARD	Use the focus control to achieve a sharp image.
10. Start Scan	EASY	Click `START SCAN` to begin acquisition.
11. Save Image	EASY	Click `SAVE` to store the acquired image.
12. Mosaic ROI Region Save	HARD	In the 2×2 mosaic view, pan or double-click to adjust the field of view, use the MAGNIFICATION slider to zoom into vertically aligned parallel tubular or stripe-like structures, center and sharpen the region, and click `SAVE IMAGE`.

subtask initialization indicates that the agent cannot reliably complete the local scientific-instrument operation itself.

B.4 Step Budget and Termination

For subtask-level evaluation, each model is allowed up to 50 interaction steps per subtask. A subtask episode is marked as successful if the corresponding success condition is satisfied within the step budget. Otherwise, the episode is marked as failed after the budget is exhausted.

For full-episode evaluation, the agent starts from the initial simulator state and must complete all required subtasks in order. A full episode is successful only if all required workflow stages are completed within the specified workflow-level step budget. This setting evaluates long-horizon execution and captures error accumulation across the entire scientific workflow.

An episode may terminate under three conditions: (i) all target success conditions are satisfied, (ii) the maximum number of allowed interaction steps is reached, or (iii) the runner encounters an unrecoverable execution error. All termination causes are recorded in the episode log.

B.5 Logging Format

After each evaluation, the runner exports a structured JSON log for each simulator and model. The log records the number of runs per subtask, subtask-level outcomes, run-level episode metadata, diagnostic grounding metrics, and overall aggregated statistics. This format allows us to compute the reported subtask success rates while preserving additional diagnostic information for failure analysis.

At the top level, each log contains `num_runs_per_subtask`, a list of `subtasks`, and an `overall_metrics_summary`. Each subtask entry stores the subtask identifier, subtask name, number of runs, success count, success rate, averaged diagnostic metrics, and the detailed records of individual runs. Each run record stores whether the run succeeds, the directory of the corresponding episode trace, and run-level metrics such as actual interaction steps, grounding accuracy, target-subtask attempts, and target-subtask completion.

The official subtask-level score is computed from `success_count` and `success_rate`. Other fields, such as `actual_steps`, `target_subtask_attempts`,

Table 7: EDS subtasks.

Subtask	Level	Description
1. Select Point Mode	EASY	Click `Point` to select point-analysis mode.
2. Pick Sample Point	EASY	Mark the sampling point for subsequent analysis.
3. Open Label	EASY	Open the label of the sampling point or spectrum.
4. Open Semi Quant	MEDIUM	Open the semi-quantitative analysis window.
5. Open Maps	EASY	Open the element distribution map.
6. Toggle Si Mapping	EASY	Turn on or off the silicon element distribution map layer.
7. Open Composite	EASY	Open the composite diagram.
8. Return to Main	EASY	Return to the main control panel.

Table 8: SPM subtasks.

Subtask	Level	Description
1. Select Tapping Mode	EASY	Select TAPPING from the MODE dropdown menu.
2. Laser Alignment	HARD	Align the laser spot onto the cantilever using the alignment controls.
3. Photodiode Alignment	HARD	Center the laser spot on the photodiode using the photodiode controls.
4. Set Target Amplitude	EASY	Set the target amplitude to 500 mV.
5. Set Frequency	MEDIUM	Set the frequency range to 200–500 kHz.
6. Auto Tune	EASY	Click the `AUTO TUNE` button to initiate automated tuning.
7. Set Scan Size	EASY	Select the desired scan size from the SCAN SIZE menu.
8. Set Integral Gain	MEDIUM	Set the integral gain value for the feedback loop.
9. Set Scan Rate	EASY	Select the scan rate from the SCAN RATE menu.
10. Set Set Point	MEDIUM	Adjust the set point slider to control the imaging force.
11. Motor Approach	MEDIUM	Use the MOTOR slider to bring the probe toward the sample surface.
12. Engage	EASY	Click the `ENGAGE` button to begin the automated approach.
13. Scan	EASY	Click the `SCAN` button to initiate the topographic scan.
14. Save	EASY	Click the `SAVE` button to export the acquired image.

`widget_grounding_accuracy`, `text_grounding_accuracy`, `state_grounding_accuracy`, and `target_subtask_success`, are used as diagnostic signals rather than as the primary score. This distinction allows the benchmark to report final task success while still analyzing intermediate grounding behavior and failure modes. For anonymity, machine-specific absolute paths in `episode_dir` are removed or anonymized before release.

B.6 Implementation Notes

The evaluation runner is implemented as a lightweight browser-based pipeline. It launches local simulator pages, captures screenshots, dispatches GUI actions, and exports structured logs after each episode. This design avoids full operating-system virtualization while retaining executable interaction with realistic scientific-instrument interfaces. Because the simulator state is logged at each step, the same traces can be used for quantitative scoring, failure localization, and qualitative case analysis.

C Subtask-level Results

This appendix reports the subtask-level success rates of all evaluated models on the LabOSWorld benchmark. Tables 13–20 provide the complete results grouped by scientific-instrument simulator. Each row represents an operation-level subtask, and all scores are reported in the range $[0, 1]$.

Table 9: APT subtasks.

Subtask	Level	Description
1. Select Sample	EASY	Select a sample from the Sample menu.
2. Align Sample	HARD	Click `ALIGN SAMPLE` and wait for the alignment process to finish.
3. Set Temperature	MEDIUM	Set the specimen temperature.
4. Set Detection	MEDIUM	Set the detection rate.
5. Set Frequency	MEDIUM	Set the voltage pulse frequency.
6. Set Pulse	MEDIUM	Set the voltage pulse fraction.
7. Start Experiment	HARD	Click `START` to run the experiment.
8. Wait Completion	HARD	Wait for the experiment to finish and click again if needed.
9. Start Reconstruction	HARD	Click `RECONSTRUCT` to enter the reconstruction view.
10. Set ICF	EASY	Choose an ICF parameter.
11. Set K-Factor	EASY	Choose a K-factor parameter.
12. Finish Experiment	EASY	Click `Finish` and then close the popup window.

Table 10: TEM subtasks.

Subtask	Level	Description
1. Select Sample	EASY	Select a sample from the SAMPLE dropdown menu.
2. Remove Holder	EASY	Click `REMOVE` to take out the empty holder.
3. Pump Airlock	EASY	Click `PUMP` in the AIRLOCK panel to evacuate the airlock.
4. Insert Specimen	EASY	Click `INSERT` to insert the sample.
5. Set Voltage	EASY	Choose an accelerating voltage from the ACC menu.
6. Enable Beam	HARD	Turn the beam on and adjust the beam intensity.
7. Adjust Magnification	HARD	Select a magnification and adjust image brightness.
8. Move Stage	HARD	Move the stage in X, Y, and Z to center the region of interest.
9. Focus Objective	HARD	Adjust the OBJECTIVE LENS FOCUS to sharpen the image.
10. Acquire Image	EASY	Insert the camera and click `ACQUIRE` to capture a TEM image.

Table 11: Definitions of the five high-level task categories and their assigned operation-level subtasks.

Task Category	Definition	Assigned Subtasks
Preparation & Sample Handling	Operations that prepare the simulator, select the target object, or load the required sample before measurement or manipulation.	SEM-S1 VentChamber; SEM-S2 OpenChamber; SEM-S3 CloseChamber; SEM-S4 EvacuateChamber; SEM-S5 SelectSample; TEM-T1 Select sample; TEM-T2 Remove empty holder; TEM-T3 Airlock pump; TEM-T4 Insert specimen; XRD-S1 SelectSpecimen; XRD-S2 Doors; LFM-L1 DragSample; FIB-F1 VentChamber; FIB-F2 PumpDown; FIB-F3 SelectSample; APT-S1 SelectSample; EDS-S2 PickSamplePoint.
Activation & Alignment	Operations that activate instrument components or align the system into a valid operating state.	SEM-S6 TurnOnHT; SPM-S1 SelectTappingMode; SPM-S2 LaserAlignment; SPM-S3 PhotodiodeAlignment; SPM-S11 MotorApproach; SPM-S12 Engage; TEM-T5 Select acceleration voltage; TEM-T6 Beam on and filament current; XRD-S3 PowerUp; LFM-L2 SelectBrightfield; LFM-L3 SelectHalogen; LFM-L4 Select10xFocus; LFM-L5 FieldDiaphragmClose; LFM-L6 FieldDiaphragmFocus; LFM-L7 FieldDiaphragmCenter; LFM-L8 FieldDiaphragmOpen; LFM-L9 ApertureDiaphragm; FIB-F4 EbeamOn; FIB-F5 EbeamLiveFocus; FIB-F8 StageZCenter; FIB-F9 IonBeamLiveCenter; APT-S2 AlignSample.
Measurement Configuration	Operations that configure parameters before acquisition, scanning, fabrication, or reconstruction.	SEM-S7 SetAccVoltage; SEM-S8 SetContrast; SEM-S9 AdjustClarity; SPM-S4 SetTargetAmplitude; SPM-S5 SetFrequency; SPM-S6 AutoTune; SPM-S7 SetScanSize; SPM-S8 SetIntegralGain; SPM-S9 SetScanRate; SPM-S10 SetSetPoint; TEM-T7 Magnification and brightness; TEM-T8 Stage position centering; TEM-T9 Objective lens focus; XRD-S4 SetAngles; XRD-S5 SetStepSize; XRD-S6 SetScanRate; LFM-L10 WhiteBalance; LFM-L11 ExposureAdjust; FIB-F6 WD7mm; FIB-F7 Tilt10deg; FIB-F13 BeamCurrent10pA; FIB-F16 IonSnapshot5000x; APT-S3 SelectTemperature; APT-S4 SelectDetectionRate; APT-S5 SelectPulseFreq; APT-S6 SelectPulseEnergy; EDS-S1 SelectPointMode.
Experimental Execution	Operations that start, monitor, or complete the main scientific action after the setup is ready.	SEM-S10 StartScan; SPM-S13 Scan; TEM-T10 Camera insert and acquire; XRD-S7 RunScan; LFM-L12 BFCapture; FIB-F10 FirstRectStart; FIB-F11 DeletePattern; FIB-F12 SecondRectStart; FIB-F14 PtNeedleIn; FIB-F15 PtDepositionStart; FIB-F17 CrossSectionCutStart; FIB-F18 CleaningCrossSectionStart; APT-S7 StartExperiment; APT-S8 WaitCompletion.
Post-processing & Completion	Operations that inspect, save, export, reconstruct, or finalize experimental outputs.	SEM-S11 SaveImage; SEM-S12 MosaicRoiRegionSave; SPM-S14 Save; XRD-S8 SaveResult; FIB-F19 Tilt0deg; FIB-F20 TaskComplete; APT-S9 Reconstruct; APT-S10 SetICF; APT-S11 SetKFactor; APT-S12 Finish; EDS-S3 OpenLabel; EDS-S4 OpenSemiQuant; EDS-S5 OpenMaps; EDS-S6 ToggleSiMapping; EDS-S7 OpenComposite; EDS-S8 ReturnToMain.

Table 12: Main fields in the exported evaluation logs.

Level	Field	Description
Global	`num_runs_per_subtask`	Number of repeated runs used for each subtask.
Global	`subtasks`	List of subtask-level evaluation records.
Global	`overall_metrics_summary`	Aggregated diagnostic metrics across all subtasks in the simulator.
Subtask	`subtask_id`	Identifier of the target subtask, such as `S1`.
Subtask	`subtask_name`	Name of the target subtask.
Subtask	`num_runs`	Number of runs available for the subtask.
Subtask	`success_count`	Number of successful runs for the subtask.
Subtask	`success_rate`	Success rate of the subtask over repeated runs.
Subtask	`metrics_summary`	Averaged diagnostic metrics over runs, such as actual steps, grounding accuracy, and target-subtask attempts.
Subtask	`runs`	Detailed records for individual runs.
Run	`run`	Index of the repeated run.
Run	`success`	Whether the run is judged successful by the evaluation script.
Run	`episode_dir`	Directory storing the episode trace and screenshots. Paths are anonymized in released logs.
Run	`metrics`	Run-level diagnostic metrics.
Metric	`actual_steps`	Number of interaction steps used by the agent.
Metric	`target_subtask_attempts`	Number of detected attempts related to the target subtask.
Metric	`widget_grounding_accuracy`	Whether the agent correctly grounds the target GUI widget when applicable.
Metric	`text_grounding_accuracy`	Whether the agent correctly grounds the relevant text or label when applicable.
Metric	`state_grounding_accuracy`	Whether the agent correctly reaches or recognizes the required simulator state when applicable.
Metric	`target_subtask_success`	Whether the target-subtask event is detected in the in-page log.

Table 13: Subtask-level success rates on SEM.

ID	Subtask	Qwen3VL-32B	EvoCUA	Sonnet-4.5	Kimi	Seed	GPT-5.5	Opus-4.5	UI-TARS	GUI-Owl	GTA1	VLAA	Hippo
S1	VentChamber	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S2	OpenChamber	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S3	CloseChamber	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00
S4	EvacuateChamber	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S5	SelectSample	0.50	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.50	0.50	0.00	0.50
S6	TurnOnHT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00
S7	SetAccVoltage	1.00	1.00	1.00	0.50	1.00	1.00	1.00	0.00	1.00	1.00	0.00	1.00
S8	SetContrast	0.50	1.00	1.00	0.00	0.00	1.00	1.00	0.00	0.50	1.00	0.00	0.50
S9	AdjustClarity	1.00	1.00	1.00	0.50	0.50	1.00	1.00	0.00	1.00	1.00	0.00	1.00
S10	StartScan	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00
S11	SaveImage	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00	0.00	1.00
S12	MosaicRoiRegionSave	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 14: Subtask-level success rates on SPM.

ID	Subtask	Qwen3VL-32B	EvoCUA	Sonnet-4.5	Kimi	Seed	GPT-5.5	Opus-4.5	UI-TARS	GUI-Owl	GTA1	VLAA	Hippo
S1	SelectTappingMode	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S2	LaserAlignment	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00
S3	PhotodiodeAlignment	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
S4	SetTargetAmplitude	0.50	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00
S5	SetFrequency	0.50	0.00	1.00	0.00	1.00	1.00	1.00	1.00	0.50	1.00	0.00	0.50
S6	AutoTune	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S7	SetScanSize	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00
S8	SetIntegralGain	0.50	1.00	0.00	1.00	1.00	1.00	0.00	1.00	0.50	1.00	0.00	0.50
S9	SetScanRate	1.00	0.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S10	SetSetPoint	0.50	1.00	0.00	0.00	0.00	1.00	0.00	0.50	0.50	0.50	0.00	0.50
S11	MotorApproach	0.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.50	0.50	0.00	0.50
S12	Engage	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S13	Scan	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S14	Save	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 15: Subtask-level success rates on TEM.

ID	Subtask	Qwen3VL-32B	EvoCUA	Sonnet-4.5	Kimi	Seed	GPT-5.5	Opus-4.5	UI-TARS	GUI-Owl	GTA1	VLAA	Hippo
T1	Select sample	0.50	1.00	0.00	1.00	1.00	0.00	0.00	1.00	0.50	1.00	1.00	0.50
T2	Remove empty holder	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
T3	Airlock pump	0.50	0.00	0.00	0.00	1.00	1.00	0.00	1.00	0.50	1.00	0.00	0.50
T4	Insert specimen	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
T5	Select acceleration voltage	0.50	1.00	1.00	1.00	1.00	0.00	1.00	1.00	0.50	1.00	0.00	0.50
T6	Beam on and filament current	0.00	0.00	0.00	1.00	0.50	1.00	0.00	0.00	0.00	0.50	0.00	0.00
T7	Magnification and brightness	1.00	1.00	1.00	0.00	0.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00
T8	Stage position centering	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00
T9	Objective lens focus	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00
T10	Camera insert and acquire	0.50	0.00	0.00	1.00	1.00	0.00	0.00	1.00	0.50	1.00	0.00	0.50

Table 16: Subtask-level success rates on XRD.

ID	Subtask	Qwen3VL-32B	EvoCUA	Sonnet-4.5	Kimi	Seed	GPT-5.5	Opus-4.5	UI-TARS	GUI-Owl	GTA1	VLAA	Hippo
S1	SelectSpecimen	0.50	1.00	1.00	1.00	1.00	0.00	1.00	1.00	0.50	1.00	1.00	0.50
S2	Doors	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S3	PowerUp	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00
S4	SetAngles	0.50	0.00	0.00	1.00	1.00	0.00	0.00	1.00	0.50	0.50	0.00	0.50
S5	SetStepSize	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00
S6	SetScanRate	0.50	1.00	1.00	0.00	0.00	1.00	1.00	0.00	0.50	1.00	1.00	0.50
S7	RunScan	0.50	0.00	0.00	1.00	1.00	0.00	0.00	1.00	0.50	0.50	0.00	0.50
S8	SaveResult	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 17: Subtask-level success rates on LFM.

ID	Subtask	Qwen3VL-32B	EvoCUA	Sonnet-4.5	Kimi	Seed	GPT-5.5	Opus-4.5	UI-TARS	GUI-Owl	GTA1	VLAA	Hippo
L1	DragSample	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00
L2	SelectBrightfield	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00
L3	SelectHalogen	0.00	0.00	0.00	1.00	1.00	0.00	0.00	0.50	0.00	0.50	0.00	0.50
L4	Select10xFocus	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00
L5	FieldDiaphragmClose	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00	0.00	1.00
L6	FieldDiaphragmFocus	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L7	FieldDiaphragmCenter	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00
L8	FieldDiaphragmOpen	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00	0.00	1.00
L9	ApertureDiaphragm	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00
L10	WhiteBalance	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.50
L11	ExposureAdjust	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
L12	BFCapture	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 18: Subtask-level success rates on FIB.

ID	Subtask	Qwen3VL-32B	EvoCUA	Sonnet-4.5	Kimi	Seed	GPT-5.5	Opus-4.5	UI-TARS	GUI-Owl	GTA1	VLAA	Hippo
F1	VentChamber	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F2	PumpDown	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F3	SelectSample	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F4	EbeamOn	0.50	0.00	1.00	0.50	1.00	1.00	1.00	1.00	0.50	1.00	0.00	0.50
F5	EbeamLiveFocus	0.00	0.00	0.00	0.50	0.00	1.00	0.00	0.00	0.00	0.50	0.00	0.00
F6	WD7mm	0.50	0.00	1.00	0.50	0.50	1.00	1.00	1.00	0.50	1.00	0.00	0.50
F7	Tilt10deg	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F8	StageZCenter	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
F9	IonBeamLiveCenter	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00
F10	FirstRectStart	0.00	0.00	0.00	1.00	1.00	0.00	0.00	1.00	0.00	0.50	0.00	0.00
F11	DeletePattern	0.00	0.00	0.00	0.50	0.50	1.00	0.00	1.00	0.50	0.50	0.00	0.00
F12	SecondRectStart	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00
F13	BeamCurrent10pA	0.50	1.00	0.00	0.50	1.00	1.00	0.00	1.00	0.50	1.00	0.00	0.50
F14	PtNeedleIn	0.50	1.00	0.00	1.00	1.00	1.00	0.00	1.00	0.50	1.00	0.00	0.50
F15	PtDepositionStart	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.50	0.00	0.00
F16	IonSnapshot5000x	0.50	1.00	0.00	1.00	1.00	1.00	0.00	1.00	0.50	1.00	0.00	0.50
F17	CrossSectionCutStart	0.00	0.00	0.00	0.50	1.00	1.00	0.00	0.00	0.00	0.50	0.00	0.00
F18	CleaningCrossSectionStart	0.50	1.00	0.00	1.00	1.00	1.00	0.00	1.00	0.50	1.00	0.00	0.50
F19	Tilt0deg	0.00	0.00	0.00	0.50	1.00	0.00	0.00	1.00	0.00	0.50	0.00	0.00
F20	TaskComplete	0.00	0.00	0.00	0.50	1.00	0.00	0.00	1.00	0.00	0.50	0.00	0.00

Table 19: Subtask-level success rates on APT.

ID	Subtask	Qwen3VL-32B	EvoCUA	Sonnet-4.5	Kimi	Seed	GPT-5.5	Opus-4.5	UI-TARS	GUI-Owl	GTA1	VLAA	Hippo
S1	SelectSample	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S2	AlignSample	0.50	1.00	0.00	0.00	0.50	1.00	0.00	0.00	0.50	0.50	0.00	0.50
S3	SelectTemperature	0.50	1.00	0.00	1.00	1.00	1.00	0.00	1.00	0.50	1.00	0.00	0.50
S4	SelectDetectionRate	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S5	SelectPulseFreq	0.50	0.50	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00	0.00	0.50
S6	SelectPulseEnergy	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S7	StartExperiment	0.50	1.00	1.00	0.00	1.00	1.00	1.00	0.00	0.50	1.00	1.00	0.50
S8	StopExperiment	1.00	1.00	1.00	0.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00	0.50
S9	Reconstruct	0.50	1.00	1.00	0.00	0.00	1.00	0.00	0.00	0.50	0.50	0.00	0.50
S10	SetICF	0.50	0.50	0.00	1.00	1.00	1.00	0.00	1.00	0.50	0.50	0.00	0.50
S11	SetKFactor	0.50	0.00	0.00	1.00	1.00	0.50	0.00	1.00	0.50	0.50	0.00	0.00
S12	Finish	0.50	0.00	1.00	1.00	1.00	0.50	1.00	1.00	0.50	1.00	1.00	0.50

Table 20: Subtask-level success rates on EDS.

ID	Subtask	Qwen3VL-32B	EvoCUA	Sonnet-4.5	Kimi	Seed	GPT-5.5	Opus-4.5	UI-TARS	GUI-Owl	GTA1	VLAA	Hippo
S1	SelectPointMode	0.50	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S2	PickSamplePoint	0.50	1.00	1.00	1.00	0.50	1.00	1.00	0.50	1.00	0.50	0.00	1.00
S3	OpenLabel	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S4	OpenSemiQuant	0.50	0.50	1.00	0.00	0.50	1.00	1.00	1.00	1.00	0.50	1.00	1.00
S5	OpenMaps	1.00	0.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00
S6	ToggleSiMapping	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S7	OpenComposite	1.00	0.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S8	ReturnToMain	1.00	0.00	1.00	0.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00