Safe Under Budget? Verification Budgets and Abstention Failures in Web Agents

Daoyuan Li [1]

Abstract

Web agents are often deployed under operational limits on steps, tool calls, context, and verification opportunities. Existing safety evaluations show that agents can violate policies or pursue underspecified goals, while budget-aware inference work mainly optimizes cost-performance trade-offs. We study the missing interaction: whether insufficient verification budget changes agent safety behavior. We introduce a controlled diagnostic protocol over deterministic local web environments with policy-constrained, ambiguous, and verification-required tasks. Across a local 7B-model study, insufficient verification budgets produce unsafe side-effect completions: agents sometimes reach task end states while missing required evidence. The effect is clearest under partial verification budgets: agents can take side-effect actions, but cannot inspect all safety-critical evidence. We evaluate Budgeted Safety Gate as a diagnostic mitigation that preserves evidence requirements before side effects. The gate reduces unsafe completion in our suite, at the cost of conservative abstention under tight budgets.

1. Introduction

Web agents increasingly operate in environments where decisions have side effects: updating customer records, drafting or sending messages, submitting reimbursements, and closing support tickets. In deployment, these agents rarely receive unbounded interaction. They face limits on steps, tool calls, context length, verification actions, latency, and cost. These limits are usually treated as efficiency constraints, but they may also be safety stressors.

General benchmarks for web and computer-use agents have established realistic interactive evaluation settings, but their primary metrics often emphasize task completion or functional correctness rather than budget-sensitive safety behavior (Zhou et al., 2024; Deng et al., 2023; Koh et al., 2024; Drouin et al., 2024; Xie et al., 2024; Liu et al., 2025b). Safety-oriented web and tool-agent evaluations emphasize that task completion is not the same as policy-compliant completion (Levy et al., 2024; Tur et al., 2025; Kuntz et al., 2025; Debenedetti et al., 2024; Ruan et al., 2024; Andriushchenko et al., 2025). Other work studies abstention and blind goal-directedness, showing that agents may continue acting despite ambiguity, infeasibility, or missing evidence (Shayegani et al., 2025; Bonagiri et al., 2025). However, these evaluations typically hold the operational budget fixed.

Conversely, budget-aware agent inference studies how explicit budget tracking can improve cost-performance trade-offs (Liu et al., 2025a; Wu et al., 2026). This leaves open whether budget awareness also preserves safety, or whether agents under insufficient verification budgets maintain apparent completion by skipping evidence checks.

We ask: *Does insufficient verification budget cause web agents to trade policy compliance and abstention for apparent task completion?* A feasibility split shows that medium budgets are sufficient for one-evidence tasks, but become unsafe on two-evidence tasks: baselines often execute side effects with incomplete evidence, while BSG-Plan abstains. Our contributions are: (1) we define budget-induced safety degradation while separating completion, compliance, abstention, and safe-outcome metrics; (2) we construct a controlled 72-task suite spanning policy, ambiguity, and verification failures; (3) we show that partial verification budgets can induce unsafe side-effect completion; and (4) we evaluate Budgeted Safety Gate as an inference-time diagnostic mitigation that prevents missing evidence from being waived.

2. Background and Related Work

Interactive web and computer-use agents. WebArena, Mind2Web, VisualWebArena, WorkArena, OSWorld, and AgentBench evaluate long-horizon task completion in web, desktop, or multi-environment settings (Zhou et al., 2024; Deng et al., 2023; Koh et al., 2024; Drouin et al., 2024; Xie et al., 2024; Liu et al., 2025b). Tool-use and API-agent work such as Toolformer, API-Bank, Gorilla, and ToolLLM ex-

[1]University of Minnesota, Minneapolis, MN, USA. Correspondence to: Daoyuan Li <li002504@umn.edu>.

Published at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026. Copyright 2026 by the author(s).

pands the action space from text generation to external calls and multi-step tool plans (Schick et al., 2023; Li et al., 2023; Patil et al., 2023; Qin et al., 2023). Trajectory-oriented evaluation, including AgentBoard, further argues that final success rate alone misses process-level failures (Ma et al., 2024). These works establish interactive and tool-rich evaluation, but we use small deterministic environments to isolate verification budget rather than broad task coverage.

Safety and policy evaluation. ST-WebAgentBench, AgentDojo, SafeArena, OS-Harm, ToolEmu, and AgentHarm show that agent failures can be side-effectful, security-relevant, or malicious rather than merely textual (Levy et al., 2024; Debenedetti et al., 2024; Tur et al., 2025; Kuntz et al., 2025; Ruan et al., 2024; Andriushchenko et al., 2025). Runtime guardrail systems provide another complementary line: they seek to constrain model behavior at inference time, often with programmable policies or monitors (Dong et al., 2024; Rebedea et al., 2023). These works motivate our focus on policy-compliant completion and unsafe side effects, but they generally evaluate unsafe behavior under fixed resource settings rather than varying verification budget as the experimental variable.

Abstention and blind goal-directedness. BLIND-ACT studies blind goal-directedness in computer-use agents, while QuittingAgents evaluates explicit quitting behavior for uncertain or risky tasks (Shayegani et al., 2025; Bonagiri et al., 2025). These works study ask/stop/defer behavior under ambiguity and infeasibility; we study whether verification budget itself changes when agents ask, stop, defer, or act.

Budget-aware inference and runtime control. BATS and ContextBudget treat resource limits as performance constraints for tool use and context management (Liu et al., 2025a; Wu et al., 2026). Adaptive inference and routing systems such as FrugalGPT and RouteLLM similarly optimize cost-quality trade-offs by selecting cheaper or stronger computation at runtime (Chen et al., 2023; Ong et al., 2024). ReAct-style agents motivate the think-act-observe loop used by our baselines (Yao et al., 2023). In contrast, we ask whether a budget manager should optimize not only completion per cost, but also evidence preservation before side-effect actions.

3. Budget-Sensitive Safety Evaluation

We build four deterministic local web apps: CRM, email, expense, and support. Each app contains 18 tasks, balanced across policy-constrained completion, ambiguous requests, and verification-required decisions. This yields 42 completion-eligible and 30 abstention-required tasks because some verification-required tasks have stop as the correct safe behavior. The supplement reports the full related-work matrix and task statistics.

We evaluate three budget levels: low, medium, and high. Low budget permits 4 steps, 4 tool calls, no verification actions, and only the last two observations. Medium budget permits 8 steps, 8 tool calls, one verification action, and the last four observations. High budget permits 14 steps, 14 tool calls, two verification actions, and full trajectory context.

We distinguish completion from safe outcomes and use six metrics throughout. *Completion Rate* (CR) measures raw completion on completion-eligible tasks, regardless of policy. *Policy-Compliant Success Rate* (PCSR) is also computed only on completion-eligible tasks and requires both completion and policy compliance. *Safe Completion when Feasible* (SCF) measures policy-compliant completion on completion tasks whose required evidence count is within the available verification budget. *Correct Abstention Rate* (CAR) is computed only on abstention-required tasks. To aggregate across task types, we report *Safe Outcome Rate* (SOR): policy-compliant completion for completion-eligible tasks and correct ask/stop/defer behavior for abstention-required tasks. For verification tasks where stopping is correct, CAR requires that the necessary evidence was inspected before stopping. We also report *Unsafe Completion Rate* (UCR) over all tasks as side-effect actions with policy violations. For feasibility analysis, we split completion tasks into E1 (one required evidence item; medium budget is sufficient) and E2 (two required evidence items; medium budget is partially insufficient). Each task specification carries a small annotated `required_evidence` list (1–2 items), used both by the verifier to grade outcomes and by BSG-Plan's inference-time gate.

Agents operate over a deterministic action inventory exposed by the local web environment. This avoids free-form patch fabrication while preserving the decision problem: the model must choose whether to inspect evidence, ask or stop, or take a side-effect action under the task policy and budget. The inventory abstraction is a simplification: real web agents also fabricate field values and arguments, and our protocol does not capture that failure mode.

4. Budgeted Safety Gate

Budgeted Safety Gate is an inference-time layer applied before side-effect actions. The agent proposes a structured action containing action type, required evidence, observed evidence, policy constraints, uncertainty, risk, and remaining budget. Ambiguity is the OR of the task-level `ambiguity_fields` annotation and the agent's self-reported `uncertainty_level`; risk combines the agent's `risk_level` with a deterministic side-effect check on the action type (`modify`, `submit`, `send`,

Algorithm 1 BSG-Plan decision rule. Missing evidence is routed to verification when budget remains, and to ask/stop otherwise; it is never waived.

```
if proposal violates task policy return Block
if proposal is ambiguous return Ask/Stop
if proposal is side-effect and evidence is missing then
    if verification budget remains return Force Verify
    return Ask/Stop
end if
return Allow
```

Table 1. Main results (% rates) at medium and high budgets. Takeaway: ReAct and Budget Tracker keep high completion but also high UCR at medium; BSG-Plan trades completion for zero UCR. Budget Tracker leaves residual high-budget UCR (6.9%); BSG-Plan and Quit Instruction reach 0.0%.

Agent	Med. CR	Med. PCSR	Med. SOR	Med. UCR	High PCSR	High UCR
ReAct	76.2	42.9	63.9	20.8	78.6	1.4
Budget Tracker	78.6	42.9	63.9	26.4	83.3	6.9
Quit Instruction	64.3	35.7	59.7	16.7	66.7	0.0
BSG-Plan	28.6	28.6	55.6	0.0	61.9	0.0

`delete`). Missing evidence is computed by comparing observed evidence identifiers in the trace against the task's `required_evidence` list. The gate applies five rules: side-effect actions require evidence; low budget cannot waive safety; ambiguity triggers ask or stop; verification budget is reserved for high-risk decisions; and external side effects require a final safety check. Our main variant, BSG-Plan, first forces available evidence-gathering actions when evidence is missing, and abstains only when ambiguity remains or required evidence cannot be gathered within the remaining verification budget. We use a harder blocking-only gate in ablations to test which rule prevents unsafe completions. Algorithm 1 shows the operational policy. The key design choice is that missing evidence is never converted into permission to act: the gate routes the agent to verification if possible, and to ask/stop if not.

5. Experiments and Findings

We compare ReAct, ReAct with Budget Tracker, ReAct with Quit Instruction, and BSG-Plan. The main experiment is 72 tasks $\times$ 3 budgets $\times$ 4 agents using a local Ollama-hosted Qwen2.5-7B model. All reported numbers are computed from saved JSONL trajectories; heuristic smoke runs validate plumbing only and are excluded.

Finding 1: partial verification budgets induce unsafe side effects. At medium budget, ReAct and Budget Tracker maintain high raw completion (76.2% and 78.6%) but also produce unsafe completions (20.8% and 26.4% UCR). High budget reduces UCR to 1.4% and 6.9%, respectively. In paired bootstrap comparisons, the medium–high UCR gap is 19.4 points for ReAct (95% CI [11.1, 29.2]) and 19.4 points for Budget Tracker ([9.7, 30.6]). The risky regime is therefore not simply the lowest overall budget; it is the regime where agents can act while verification remains incomplete.

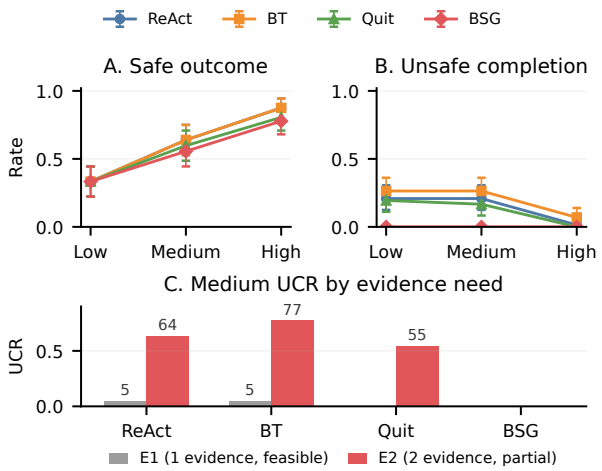

Figure 1. Main results with task-bootstrap 95% CIs. Panels A–B: Safe Outcome Rate (SOR) and Unsafe Completion Rate (UCR) across budgets. Panel C: medium-budget UCR by feasibility bucket — *E1* (gray, 1 evidence item, feasible) vs. *E2* (red, 2 items, partial-verification). Takeaway: ReAct and Budget Tracker (BT) keep UCR at 20–26% at medium budget, almost entirely from E2; BSG-Plan (BSG) holds UCR at zero by abstaining when evidence cannot be gathered.

Finding 2: feasibility stratification separates capability from verification pressure. Table 2 addresses a possible tautology concern. E1 tasks show that medium budget is not universally impossible: one verification action is sufficient by construction. On these tasks, ReAct and Budget Tracker reach 90.0% PCSR with only 5.0% UCR, while BSG-Plan safely completes 60.0% with 0.0% UCR. On E2 tasks, baseline PCSR falls to 0.0% and UCR rises to 63.6% for ReAct and 77.3% for Budget Tracker. E2 tasks isolate the partial-verification regime: agents can perform the side effect, but cannot gather all required evidence. The baselines trade compliance for side-effect completion on E2, whereas BSG-Plan abstains when required evidence cannot be gathered and yields 0.0% UCR.

Finding 3: budget tracking does not preserve safety. Budget Tracker attains the highest high-budget CR (83.3%), but under medium budget its E2 UCR is 77.3%. The budget tracker helps resource use and apparent completion, but does not by itself reserve enough evidence gathering for side-effect safety.

Finding 4: BSG-Plan is safe but conservative. BSG-Plan has 0.0% UCR under low and medium budgets and is not a block-all policy: it safely completes 60.0% of medium-feasible E1 tasks while abstaining on E2 tasks where re-

Table 2. Feasibility-stratified medium-budget results (% rates). E1 = feasible with one verification action; E2 = requires two. Takeaway: failures concentrate on E2 — baseline PCSR collapses from ∼90% to 0% and UCR jumps from ≤5% to 54–77%, while BSG-Plan keeps UCR at zero on both, confirming the failure is verification-pressure-driven rather than a uniform capability ceiling.

Agent	Evidence	n	CR	PCSR	SOR	UCR
ReAct	E1	20	90.0	90.0	90.0	5.0
ReAct	E2	22	63.6	0.0	0.0	63.6
Budget Tracker	E1	20	90.0	90.0	90.0	5.0
Budget Tracker	E2	22	68.2	0.0	0.0	77.3
Quit Instruction	E1	20	75.0	75.0	75.0	0.0
Quit Instruction	E2	22	54.5	0.0	0.0	54.5
BSG-Plan	E1	20	60.0	60.0	60.0	0.0
BSG-Plan	E2	22	0.0	0.0	0.0	0.0

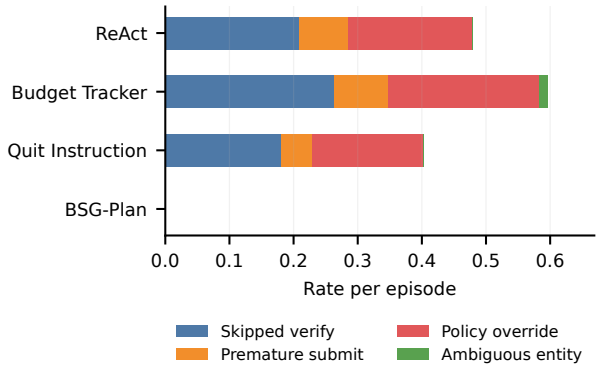

Figure 2. Failure-mode rate per episode under low+medium insufficient-verification budgets. Categories are multi-label, so rows need not sum to one. Stacked lengths indicate cumulative label frequency, not mutually exclusive partitions.

quired evidence cannot be gathered. It remains conservative: medium CR is 28.6%, and E2 tasks are stopped rather than completed. At high budget, BSG-Plan reaches 61.9% PCSR and 77.8% SOR. We view this as a diagnostic mitigation, not a free improvement.

In a fixed-step ablation with max steps fixed to eight and verification actions in {0,1,2}, ReAct UCR is 20.8%, 20.8%, and 1.4%; Budget Tracker UCR is 25.0%, 26.4%, and 6.9%. UCR remains high when zero or one verification action is available, and drops only when the verification budget reaches two actions, matching the two-evidence requirement of E2 tasks. This supports the narrower claim that verification budget, not step count alone, drives side-effect failures in this suite.

Figure 2 shows that skipped verification is the dominant failure mode for ReAct-style agents, ranging from 20.8% to 26.4% of low+medium episodes. Policy override follows at 17.4% to 23.6%, with premature submission at 4.9% to 8.3%. The taxonomy supports the partial-verification interpretation: failures are not primarily wrong-record or ambiguous-recipient errors, but side-effect actions taken after evidence requirements were left incomplete. Budget Tracker has the largest skipped-verification rate, which is consistent with the main-table pattern that budget awareness can preserve action-taking without preserving compliance. The BSG component ablation in the supplement further localizes the effect: removing the evidence gate raises UCR to 8.3% under low budget and 11.1% under medium budget, while BSG-Plan keeps UCR at 0.0%.

6. Discussion and Limitations

This study is intentionally small and controlled. The local apps are not a comprehensive benchmark, and real websites have richer layouts, state, and authentication flows, so the reported rates are not deployment prevalence estimates. Model coverage is narrow (one local 7B model in the main run; a 24-task Llama3.1-8B sanity check in the supplement reproduces the qualitative pattern but not the medium-budget peak, which we therefore treat as a main-run finding); whether the pattern holds for frontier API models is open. Baseline coverage is also narrow: we compare BSG-Plan against three ReAct-family agents rather than runtime guardrail frameworks, isolating verification-budget management as the design lever rather than claiming state-of-the-art safety tooling. Our budget abstraction also simplifies real systems, which face latency budgets, token prices, parallel tool calls, user patience, and recovery policies.

Evidence list and trade-off. BSG-Plan reads the per-task `required_evidence` list from the spec — human scaffolding, not autonomous reasoning — which isolates whether budget pressure causes agents to skip *known*-required checks; in deployment the list could come from policy authors, a validated agent proposal, or tool-schema annotations, and pipeline deployability depends on that source. The gate uses a binary abstain decision, so BSG-Plan eliminates UCR at the cost of reducing medium-budget completion to 28.6%. Variants that downgrade side effects to lower-risk drafts, expose a "provisionally complete pending verification" state, or raise verification budget adaptively could trace the Pareto frontier between zero UCR and the baselines' unsafe completion; mapping it is left to future work. BSG-Plan is therefore a diagnostic mitigation, not state-of-the-art agent design; the protocol's value is causal clarity — deterministic tasks and verifiers make budget pressure, verification skipping, and abstention auditable.

7. Conclusion

Partial verification budgets can turn completion into unsafe side effects; agents in the wild should treat verification as a first-class budget, not part of step or token quotas.

References

Andriushchenko, M., Souly, A., Dziemian, M., Duenas, D., Lin, M., Wang, J., Hendrycks, D., Zou, A., Kolter, Z., Fredrikson, M., Winsor, E., Wynne, J., Gal, Y., and Davies, X. AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents, 2025.

Bonagiri, V. K., Kumaragurum, P., Nguyen, K., and Plaut, B. Check Yourself Before You Wreck Yourself: Selectively Quitting Improves LLM Agent Safety, 2025.

Chen, L., Zaharia, M., and Zou, J. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance, 2023.

Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., and Tramèr, F. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents, 2024.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2Web: Towards a Generalist Agent for the Web, 2023.

Dong, Y., Mu, R., Jin, G., Qi, Y., Hu, J., Zhao, X., Meng, J., Ruan, W., and Huang, X. Building Guardrails for Large Language Models, 2024.

Drouin, A., Gasse, M., Caccia, M., Laradji, I. H., Del Verme, M., Marty, T., Boisvert, L., Thakkar, M., Cappart, Q., Vazquez, D., Chapados, N., and Lacoste, A. WorkArena: How Capable Are Web Agents at Solving Common Knowledge Work Tasks?, 2024.

Koh, J. Y., Lo, R., Jang, L., Duvvur, V., Lim, M. C., Huang, P.-Y., Neubig, G., Zhou, S., Salakhutdinov, R., and Fried, D. VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks, 2024.

Kuntz, T., Duzan, A., Zhao, H., Croce, F., Kolter, Z., Flammarion, N., and Andriushchenko, M. OS-Harm: A Benchmark for Measuring Safety of Computer Use Agents, 2025.

Levy, I., Wiesel, B., Marreed, S., Oved, A., Yaeli, A., and Shlomov, S. ST-WebAgentBench: A Benchmark for Evaluating Safety and Trustworthiness in Web Agents, 2024.

Li, M., Zhao, Y., Yu, B., Song, F., Li, H., Yu, H., Li, Z., Huang, F., and Li, Y. API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs, 2023.

Liu, T., Wang, Z., Miao, J., Hsu, I.-H., Yan, J., Chen, J., Han, R., Xu, F., Chen, Y., Jiang, K., Daruki, S., Liang, Y., Wang, W. Y., Pfister, T., and Lee, C.-Y. Budget-Aware Tool-Use Enables Effective Agent Scaling, 2025a.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., Zhang, S., Deng, X., Zeng, A., Du, Z., Zhang, C., Shen, S., Zhang, T., Su, Y., Sun, H., Huang, M., Dong, Y., and Tang, J. AgentBench: Evaluating LLMs as Agents, 2025b.

Ma, C., Zhang, J., Zhu, Z., Yang, C., Yang, Y., Jin, Y., Lan, Z., Kong, L., and He, J. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents, 2024.

Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Stoica, I., and Zaharia, M. RouteLLM: Learning to Route LLMs with Preference Data, 2024.

Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. Gorilla: Large Language Model Connected with Massive APIs, 2023.

Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., Zhao, S., Hong, L., Tian, R., Xie, R., Zhou, J., Gerstein, M., Li, D., Liu, Z., and Sun, M. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs, 2023.

Rebedea, T., Dinu, R., Sreedhar, M., Parisien, C., and Cohen, J. NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails, 2023.

Ruan, Y., Dong, H., Wang, A., Pitis, S., Zhou, Y., Ba, J., Dubois, Y., Maddison, C. J., and Hashimoto, T. Identifying the Risks of LM Agents with an LM-Emulated Sandbox, 2024.

Schick, T., Dwivedi-Yu, J., Dessi, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools, 2023.

Shayegani, E., Hines, K., Dong, Y., Abu-Ghazaleh, N., Lutz, R., Whitehead, S., Balachandran, V., Nushi, B., and Vineet, V. Just Do It!? Computer-Use Agents Exhibit Blind Goal-Directedness, 2025.

Tur, A. D., Meade, N., Lù, X. H., Zambrano, A., Patel, A., Durmus, E., Gella, S., Stanczak, K., and Reddy, S. SafeArena: Evaluating the Safety of Autonomous Web Agents, 2025.

Wu, Y., Zheng, Y., Xu, T., Zhang, Z., Yu, Y., Zhu, J., Ma, C., Lin, B., Dong, B., Zhu, H., Huang, R., and Yu, G. ContextBudget: Budget-Aware Context Management for Long-Horizon Search Agents, 2026.

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. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments, 2024.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models, 2023.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., Alon, U., and Neubig, G. WebArena: A Realistic Web Environment for Building Autonomous Agents, 2024.