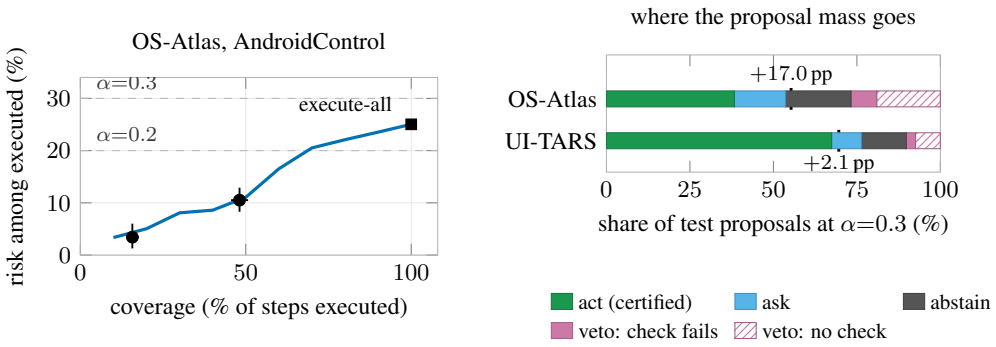


Figure 1: *Left:* OS-Atlas risk–coverage on AndroidControl, with its two certified operating points and episode-cluster bootstrap 95% intervals. *Right:* proposal mass at α=0.3 per backbone; the veto withholds two populations, the larger the proposals no applicable check covers; the rule marks the coverage withholding only the checker-failing slice reaches under the same guarantee.

VERIACT: THE COVERAGE A CERTIFICATE VETO COSTS FALLS ON UNVERIFIABLE ACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

A computer-use agent that must act at every step inherits the full error rate of its model. We study a training-free gate between a frozen GUI agent's policy and the environment that decides, for each proposed action, whether to ACT, ABSTAIN, or ASK a human. The gate fuses three uncertainty signals (self-consistency over equivalence clusters of the sampled actions, a frozen pre-operative critic, and model confidence), applies a machine-checkable static executability certificate over the accessibility tree or DOM as a veto, and turns the fused score into an ACT threshold whose executed-step error is bounded by simultaneous Bentkus bounds over a fixed threshold family, fitted on independent episode anchors. On AndroidControl and Mind2Web's three splits with two frozen 7B backbones the gate attains an 8.9% mean AURC reduction over the best of its own component signals used alone, with no external calibrated gate compared, and the pooled certificate keeps executed risk at or below the target at every non-vacuous setting. The main finding concerns the veto. 72.9% / 66.1% of what it withholds on AndroidControl and on Mind2Web carries no applicable check at all rather than a check that fails. What the veto buys is that no proposal violating the static checker's preconditions executes, and withholding only that checker-failing slice keeps the certificate and recovers certified coverage on both benchmarks, most of it on AndroidControl. The certified level does not transfer across splits although the ordering of policies does, and all claims are offline and step-level, with no live execution.

1 INTRODUCTION

Computer-use agents, systems that operate phones, browsers and desktops through the screens and inputs a human would use, have improved rapidly with native end-to-end GUI models such as UI-TARS (Qin et al., 2025), grounding models such as SeeClick (Cheng et al., 2024), and step-level

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

benchmarks (Li et al., 2024; Deng et al., 2023). Evaluation and deployment share an assumption: at every step the agent *must* emit an action, which is executed. The system inherits the model's per-step error rate, and one wrong step can fail the episode or commit something irreversible: a message sent, a file deleted, a payment submitted. The operational question is not average accuracy but whether *this particular action* should be executed, deferred or escalated (Figure 1).

Our main measured result: a hard executability veto over such an agent loses its coverage on the actions its checker cannot evaluate, not on those its checker rejects, and on OS-Atlas a second time through the calibration anchors withheld with them. Withholding only the checker-failing slice keeps the same guarantee at $+17.0$ (OS-Atlas) / $+2.1$ pp (UI-TARS) more certified coverage, nearly what no veto recovers; what the veto still buys is that the 4 of 208 / 13 of 75 proposals no veto executes against a failed check are not executed.

Selective prediction formalizes part of this choice: accept the output or abstain under a bound on accepted-output error. It needs a signal separating correct from incorrect actions and a gate turning it into a decision with a citable guarantee; for GUI agents both exist only in fragments. Uncertainty quantification has been benchmarked post hoc (Kumar et al., 2026b) and trained pre-operative critics flag bad actions before execution (Wanyan et al., 2025), but each is a score, not a calibrated decision layer. Conformal risk control (Angelopoulos et al., 2024) has been instantiated for mobile GUI automation in CORA (Feng et al., 2026), whose gate targets *harm* at trajectory level with a learned risk model, and agentic abstention framed as stop-vs-act without calibration (Luo et al., 2026). Missing is a training-free, step-level gate over an unmodified frozen agent that (i) scores each proposal with execution-grounded, machine-checkable evidence, (ii) chooses among ACT, ABSTAIN and ASK, not binary accept, and (iii) carries a distribution-free certificate on the error of executed steps.

We build that gate: a post-policy, pre-action layer between a frozen GUI agent and the environment. At every step it draws $k+1$ candidates from the frozen policy (k samples plus a greedy decode) and fuses four training-free signals: S_{static}, a static executability certificate on the accessibility tree or DOM; S_{sc}, self-consistency over equivalence clusters of the sampled actions; S_{critic}, a frozen pre-operative GUI critic (Wanyan et al., 2025) used as a signal rather than a decision; and S_{conf}, model confidence from generation logits. A finite-sample PAC calibration, simultaneous Bentkus upper bounds over a fixed threshold family fitted on one independent anchor action per held-out episode, maps the fused score to an ACT threshold with a certified bound on executed-step error, and the static certificate acts as a hard veto: a proposal executes only if a satisfied certificate covers it, so neither a checker-inapplicable action nor one the checker cannot evaluate is executed, a commitment section 4.4 prices. An irreversibility taxonomy (send / delete / pay / submit / overwrite) routes uncertain *irreversible* steps to ASK.

Evaluation is strictly offline and step-level, matching the benchmarks' protocol: the official test split of AndroidControl (Li et al., 2024) and the cross-task, cross-website and cross-domain test splits of the multimodal Mind2Web release (Deng et al., 2023), over frozen open-weight 7B GUI backbones. **RQ1**: where does the gate sit on the risk–coverage frontier (AURC) against training-free gating policies with a faithful step-level instantiation: post-hoc UQ (Kumar et al., 2026b), a frozen-critic gate (Wanyan et al., 2025), the static certificate alone, and random/execute-all references? **RQ2**: does the PAC certificate hold empirically across significance levels, under the discrete, tie-heavy scores GUI actions induce? **RQ3**: how sensitive is residual risk to the ASK budget? **RQ4**: does a gate calibrated on one split generalize across the cross-task/website/domain shift? Under them is the question every step-level gate faces: what to do with proposals no check covers.

Contributions. (i) A training-free, execution-grounded ACT/ABSTAIN/ASK gate for frozen GUI agents fusing a static executability certificate, self-consistency, a frozen critic and confidence (section 3). (ii) An episode-grouped finite-sample PAC certificate on executed-step error, simultaneous Bentkus bounds over a fixed threshold family, with a hard static-certificate veto and irreversibility-routed ASK (section 3.3). (iii) A step-level risk–coverage study on AndroidControl and three Mind2Web test splits with two frozen backbones, pricing the veto against a re-calibrated alternative that separates checker-failing from unverifiable proposals (Table 3, section 4.4).

2 Related Work

Step-level GUI and web agent benchmarks. Computer-use agents are measured by offline step-level action prediction (Rawles et al., 2023; Li et al., 2024; Deng et al., 2023; Lu et al., 2024) or online task success (Zhou et al., 2024; Rawles et al., 2024; Xie et al., 2024); both demand an action at every step, so *when to refuse* is not measured.

Foundation GUI agents and grounding. Grounding models (Cheng et al., 2024), native end-to-end agents (Qin et al., 2025) and step-adaptive compute (Wei et al., 2026) supply the actors a gate wraps, not the gate.

Trained verifiers, critics, and process rewards. Many methods learn an auxiliary judge: pre-operative critics such as GUI-Critic-R1 (Wanyan et al., 2025), process-reward and step-verification models (Chen et al., 2025; Hu et al., 2025), history-aware and continuous-alignment critics (Lee et al., 2026; Sun et al., 2026), verifier-construction and web-task judging (Cui et al., 2026; Rosset et al., 2026), and post-hoc action-effect verification (Zhang et al., 2026b); anomaly monitors over sensor streams set thresholds by search, not calibration (Xiao et al., 2025). Each emits an uncalibrated score with no guarantee on executed-step error and no three-way decision. Checkers whose rejection is built in, such as type- and effect-safe synthesis (Xiao et al., 2026a), are sound where they apply and silent elsewhere, which is why our certificate is a veto, not a ranking signal.

Uncertainty quantification and calibration. A complementary route quantifies uncertainty: grounding-level calibration (Wang et al., 2026), the cross-model UQ benchmark of Kumar et al. (2026b), dual-level uncertainty for web planning (Zhang et al., 2026a), component-level uncertainty triggering human refinement (Hao et al., 2025), and the design space of combining heterogeneous evidence (Xu et al., 2026): step-level signals exist but stop at ranking or region sizing, short of a calibrated act/abstain/ask rule.

Conformal risk control and certified agents. Conformal risk control (Angelopoulos et al., 2024) bounds the expectation of a monotone loss and has risk-controlled LLM annotation by calibrated gating (Xiao et al., 2026b), accepted trajectories under tool-use drift (Opoku & Banahene, 2026), and, closest to us, CORA and CURA (Feng et al., 2026; Kumar et al., 2026a); these certify at trajectory or safety-interruption level, not an act/abstain/ask policy over a frozen agent's steps.

Abstention, ask-for-help, and human-in-the-loop gates. Finally, agentic abstention (Luo et al., 2026) improves timely stopping at task level without a click-level certificate, action-trust scoring (Mei et al., 2025) gives a two-way act/ask decision without a conformal guarantee, query-driven and logic-based interaction (Wu et al., 2025; Lee et al., 2025) and help-seeking benchmarks (Trinh et al., 2026) formalize human handoff; Shi et al. (2025) survey the space. Across these the deferral decision is binary, task-level, learned or uncertified. Conformal ask-for-help planners (Ren et al., 2023) certify task-level help requests for LLM planners, and Learn-then-Test (Angelopoulos et al., 2022) supplies the fixed-family, union-bound construction our certificate uses; neither certifies, in finite samples over a frozen GUI agent, the error of the steps it executes.

3 VeriAct: A Training-Free Selective Execution Gate

3.1 Setting and Protocol

At each step t a frozen GUI agent π proposes an action a_t from the task goal, step instruction and observation (screenshot plus accessibility structure). The protocol is offline and step-level: gold history is replayed identically for all methods; no live device, emulator or browser runs. The gate is post-policy and pre-action: it sees the proposals and observation and emits $d_t \in \{\text{ACT}, \text{ABSTAIN}, \text{ASK}\}$. Nothing is trained: every component is a frozen open-weight model or a deterministic checker.

Candidate generation. At each step we draw $k{+}1$ candidates from the frozen policy: k temperature samples plus one greedy decode, with per-token logprobs recorded for the confidence signal. Backbones are the frozen open-weight 7B agents UI-TARS-1.5-7B (Qin et al., 2025) and OS-Atlas-Pro-7B, each served behind a smoke-tested inference stack (Appendix A).

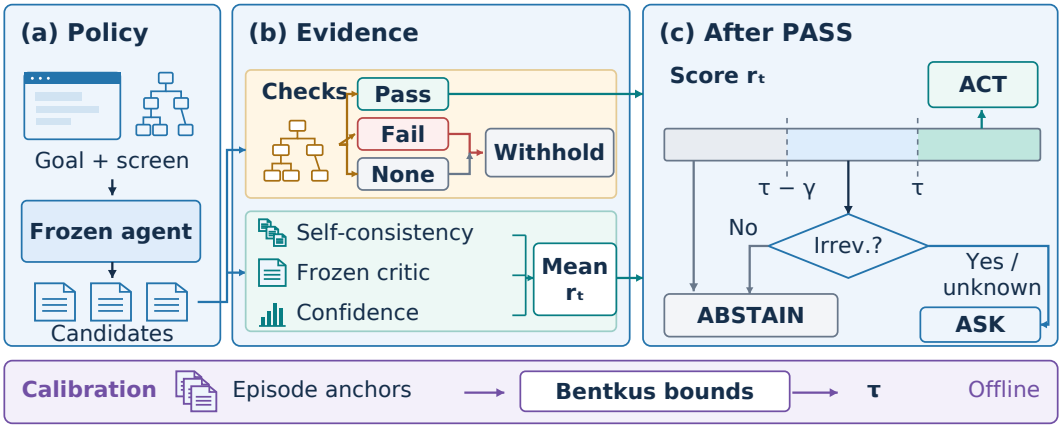


Figure 2: The gate in one picture. The static certificate checks that the target exists, that the action type applies to it and that the coordinates land inside it; a failed check and a proposal no check covers (*None*) are different populations, both withheld, and neither reaches ASK. A proposal that passes reaches ACT at a fused score $r_t \geq \tau$, ASK in the band $[\tau - \gamma, \tau)$ when the action is irreversible or of unknown irreversibility, and ABSTAIN otherwise; τ comes from the PAC calibration on one anchor per held-out episode and γ=0.10 is fixed.

3.2 Four Training-Free Signals

Static executability certificate (S_{static}). A deterministic checker evaluates machine-checkable preconditions on the step's accessibility structure (the Android accessibility forest, or the cleaned DOM with backend-node ids on the web): the target element *exists*, the action type is *applicable* to it, and coordinates land *inside* its bounds. Checks are binary and compose: an action passes every applicable check, fails one, or has none that applies (Figure 2, static channel).

Sampling self-consistency (S_{sc}). We cluster the k+1 candidates by action equivalence (same action type and resolved target/text, to coordinate tolerance) and score each by its cluster's mass.

Frozen pre-operative critic (S_{critic}). A frozen GUI critic (Wanyan et al., 2025), prompted with its authors' official template, scores the proposed action against the observation and instruction.

Model confidence (S_{conf}). Length-normalized generation logprobability of the candidate action, recorded at decode time at no extra cost.

3.3 The Three-Way Gate

Fused risk score. Signals are valued in $[0, 1]$ and combine with fixed equal weights, renormalized over the signals present, into a fused score r_t per proposal. On every proposal that can reach ACT, S_{static} is constant, so *as a ranking signal* the deployed score is an equal-weight mean of the three uncertainty signals, S_{static} contributing only the veto below. Section 4.4 measures both.

Finite-sample PAC risk calibration (validity). The ACT threshold τ is calibrated on held-out episodes, one label-independent hash-selected anchor each, so the certificate's independent unit is the episode rather than correlated within-trajectory steps. Over a fixed grid of K=101 thresholds, each acceptance set (certificate-passing anchors at or above it) gets a one-sided Bentkus-inverted binomial upper bound (Bentkus, 2004) on its error rate at failure budget δ/K, δ=0.1; the union bound makes all K bounds simultaneous, so taking as τ the maximum-coverage threshold whose bound is at most α preserves confidence $1 - \delta$, as in distribution-free risk control (Bates et al., 2021; Angelopoulos et al., 2024). When no threshold is certifiable the gate is vacuous: it accepts nothing and says so (RQ2). Calibration pools the four datasets per backbone, so the certified quantity is executed-anchor error over that mixture: a marginal guarantee, not a per-split one. The fixed family assumes no score continuity; GUI action signals are discrete and tie-heavy.

Certificate veto. A proposal reaches ACT only if a *satisfied, non-vacuous* certificate covers it; two disjoint populations fail it. An action failing an applicable check is never executed, never routed to ASK: a checker-inapplicable action is a replan case rather than an uncertainty one. "Failing" is the checker's static reading of the recorded accessibility tree: an action the parser rejects, a click on a node that is not interactive or is disabled, text entry with no focused editable field, or a click point under no visible node. It is not a proof that the live environment would reject the action; on AndroidControl 29 of 223 such proposals are graded correct against gold, and Appendix A.2 classifies each one on its tree (none is a checker error on the tree; the gold action fails the same check in 10/9 of them). Steps exposing no applicable check carry none and are withheld too: nothing was proven against them, and nothing was proven for them.

Irreversibility-aware ask. A v0 irreversibility taxonomy maps action types and UI context to five classes: *send, delete, pay, submit, overwrite*. A step scoring in the uncertainty band $[\tau-\gamma, \tau)$, with $\gamma=0.10$ throughout, is routed to ASK when classified irreversible or of unknown irreversibility (the conservative reading), and to ABSTAIN when known reversible: retrying a reversible action is cheaper than a user interruption. RQ3 adds a second width s above the threshold that demotes borderline acts; every other number in the paper is at $s=0$, where the ask rate is this band's alone.

Evaluation axes. The primary axis is the step-level *risk–coverage frontier*: sweeping the threshold trades coverage (fraction of steps executed) against risk (error among them), summarized by the area under that curve (AURC), with certified precision at matched coverage as operating point. Measured baselines are training-free gates with a faithful step-level instantiation (post-hoc UQ, a frozen-critic gate, the static certificate alone) plus execute-all and random-abstention references; the task- and trajectory-level guards of section 2 admit no such instantiation and are compared there.

4 EXPERIMENTS

Setup. Benchmarks are AndroidControl's official test split and Mind2Web's three multimodal test splits, as a seeded paired step subset shared by both backbones (seed 20260720; pools and exclusions in Appendix A): 2,500 of the 8,444 AndroidControl test steps covering all 1,543 test episodes, and 417 / 318 / 1,265 Mind2Web cross-task / cross-website / cross-domain steps. Each step contributes $k+1$ candidates per backbone (81,000) feeding self-consistency; decisions are graded on its greedy proposal (9,000 records with all four signals). The static checker certifies 72.5% of AndroidControl and 92.3% of Mind2Web proposals non-vacuously; another 20.1% / 5.0% expose no applicable check (coordinate-free actions like opening an app) and carry none. These rates cover all 9,000 analyzed proposals; section 4.4 uses the 5,404 test records. Failures are retained as veto mass (section 3.3). The taxonomy routing ASK covers the two groups unevenly, which shapes RQ3: on AndroidControl 1,091 of 2,964 (36.8%) records carry an *unknown* class and 158 of 2,964 (5.3%) classified irreversible, whereas on Mind2Web 2,321 of 2,440 (95.1%) are *unknown* and 0 of 2,440 classified irreversible.

4.1 RQ1: RISK–COVERAGE FRONTIER

The headline is the fused gate's frontier position among these own-signal ablations: an 8.9% mean AURC reduction over the strongest training-free baseline (range 1.9–23.8% across 8 backbone–split pairs). The curves behind Table 1 are Figure 6 (Appendix A). The baselines are not uniformly informative: 8 of 40 normalized cells exceed 1.0 (worse than random on that leg) — the static-only baseline on 5 legs (max 1.141) and the frozen critic on 3 legs (max 1.070). Each gain is tested as an episode-paired difference, a 95% interval on $\mathrm{AURC}(\text{gate}) - \mathrm{AURC}(\text{row})$ over resampled test episodes, the same resample for every policy on a leg (Table 5): it excludes zero against execute-all, random, the static certificate and the critic on 8 of 8 legs, and against the UQ row on 5 of 8 legs. Two further OS-Atlas draws set a run-to-run floor through self-consistency alone (Appendix A, Table 4): 7 of 8 legs clear it, the weakest (UI-TARS on M2W cross-task, $0.9\times$) does not and is not read as a gain, and 6 of 8 legs clear the absolute AURC floor of 0.0144.

Fusing four training-free signals beats the strongest baseline row on all 8 legs, by an 8.9% mean AURC reduction over the strongest training-free baseline (range 1.9–23.8% across 8 backbone–split pairs); on every leg but one that row is one of the gate's own uncertainty sig-

Table 1: Step-level selective execution on the 5,404 held-out test records of AndroidControl and Mind2Web (three splits). AURC (lower is better) is the area under the risk–coverage curve, averaged over legs (backbone × split); execute-all is its full-coverage point, random the flat random-ranking curve. cert-prec @ cov pools records across legs: each policy's precision at the coverage our α=0.3 gate attains, where ours adds the ASK route to the ranking the AURC column uses. "UQ signals only" is the per-run best of self-consistency and confidence from Kumar et al. (2026b), two of our own signals; "Frozen critic gate" is the gate's own critic signal thresholded alone. The four rows are own-signal ablations rather than external systems.

Gating policy	AndroidControl		Mind2Web (3 splits)	
	AURC ↓	cert-prec @ cov ↑	AURC ↓	cert-prec @ cov ↑
Execute-all (no gate)	0.277	—	0.690	—
Random abstention	0.277	71.4%	0.688	40.5%
Static certificate only	0.232	76.8%	0.699	40.0%
UQ signals only (Kumar et al., 2026b)	0.169	85.3%	0.603	48.7%
Frozen critic gate (Wanyan et al., 2025)	0.212	80.1%	0.682	41.4%
VeriAct (ours)	**0.137**	**87.9%**	**0.565**	**51.2%**

Table 2: Every (backbone, α) setting, gate as configured (γ=0.10, s=0). τ is the calibrated threshold, U the simultaneous Bentkus bound at confidence $1-\delta$ from the e_{cal} errors among the $n_{\text{cal}}^{\text{acc}}$ anchors τ accepts; accepting none certifies nothing. Coverage and executed risk are step-weighted over the four test sets; anchor risk takes one anchor per test episode, the unit the bound is stated on.

Backbone	α	τ	$e_{\text{cal}}/n_{\text{cal}}^{\text{acc}}$	bound U	coverage	executed risk	anchor risk
OS-Atlas	0.05	1.00	0/0	*vacuous*	0.0000	—	—
	0.10	1.00	0/0	*vacuous*	0.0000	—	—
	0.20	0.99	2/84	0.1391	0.0870	0.0383	0.0299
	0.30	0.94	81/380	0.2911	0.3830	0.2657	0.2458
UI-TARS	0.05	1.00	0/0	*vacuous*	0.0000	—	—
	0.10	1.00	0/0	*vacuous*	0.0000	—	—
	0.20	1.00	0/0	*vacuous*	0.0000	—	—
	0.30	0.83	153/646	0.2972	0.6743	0.2854	0.2585

nals, so the gain is fusion over its own strongest part. The episode-paired interval excludes zero on 5 of 8 legs against that row and on 8 of 8 legs against every other; the exceptions are all UI-TARS on Mind2Web, where the run-to-run floor is also not cleared.

4.2 RQ2: PAC Certificate Validity

The certificate, calibrated per backbone on one label-independent anchor per calibration episode, promises executed-*anchor* error over the pooled mixture at confidence $1-\delta$; Table 2 evaluates every setting on that unit and on steps and passes on both: 0 non-vacuous settings violate their certificate, step-weighted error staying at or below α at every non-vacuous setting (worst margin -1.5 pp), Wilson 95% [26.5, 30.7] at UI-TARS at $\alpha = 0.3$, -4.15 pp on the anchor unit. Re-drawing the calibration/test split 20 times (Appendix A.1) violates the certificate in 2 of the 63 non-vacuous (backbone, α, split) cells, 2 of 20 on UI-TARS at $\alpha = 0.3$, worst margin -0.52 pp, inside the δ=0.1 the bound states. Per-split conditional slices fall outside the pooled guarantee; 7 exceed α, all on Mind2Web: by $+12.4$ to $+40.5$ pp at α=0.3, plus 1 single-action slice at α=0.2.

The certificate holds on its own unit and on steps at all 3 of 8 non-vacuous settings, its thinnest margin -4.15 pp on the certified unit, and 5 of the 8 (backbone, α) settings certify nothing: at 952 anchors a fixed-family bound supports only the loosest levels.

RQ3: ask-budget sensitivity. Sweeping the ask budget s traces executed risk against ask rate (Figure 3). On top of the fixed band $[\tau-\gamma, \tau)$ below the threshold, an act scoring in $[\tau, \tau+s)$ is demoted to ASK when irreversible or unclassified; at s=0 the ask rate is the fixed band's alone, and only the demotion can lower executed risk. The knee sits at an ask rate of 38% (OS-Atlas) / 44% (UI-TARS) at α=0.3, where +22/+35 pp of asks buy 19/11 pp of executed-risk reduction (tails in Appendix A). Asked steps leave the executed set as if resolved by a human, with no model of

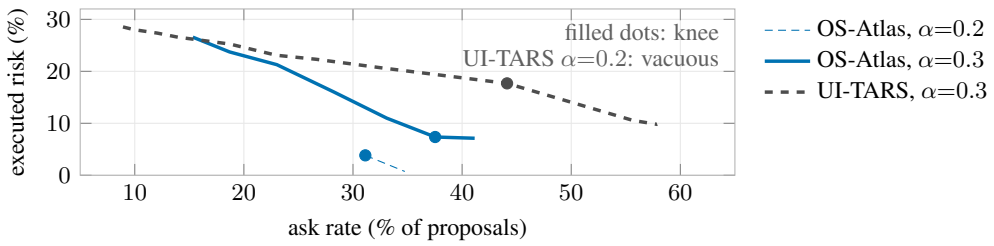


Figure 3: RQ3. Executed risk against ask rate as the demotion band s widens from 0 to 0.5; filled dots mark the knee. The UI-TARS α=0.2 gate is vacuous at every s.

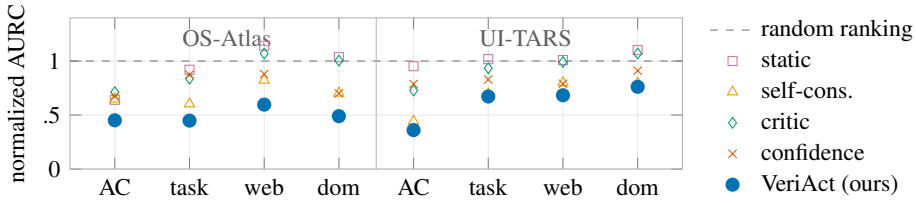


Figure 4: RQ4. Each leg's AURC on its difficulty band, $(\mathrm{AURC} - \mathrm{AURC}_{\mathrm{oracle}})/(\mathrm{AURC}_{\mathrm{random}} - \mathrm{AURC}_{\mathrm{oracle}})$: increasing and affine within a leg, so legs become comparable and no ranking changes. The gate's *level* tracks the leg, its *rank* does not.

human error, so the curves measure intervention cost rather than end-task success. On Mind2Web the routing is the unclassified default, not the taxonomy (Setup). Whether the routed asks are well chosen is a separate question, and at the gate's own budget the answer is no. At α=0.3 the gate asks 72 of 1,482 (4.9%) (OS-Atlas) / 80 of 1,482 (5.4%) (UI-TARS) of the AndroidControl test steps; two controls ask the same number and abstain from nothing, the k lowest-scoring steps or k at random (Table 6). The gate's executed risk is below both at -47.0 pp (95 % CI $[-49.7, -44.3]$) / -32.7 pp (95 % CI $[-35.1, -30.3]$) less coverage: that gap is the ABSTAIN arm. On the asked steps alone the band errs at 43.1% / 53.8%, the score rule at 100.0% / 86.2%, random at 23.6% / 23.8%: above chance, below the score alone.

Past the knee one backbone saturates and the other does not, so an ask budget set on one frozen agent does not transfer; and at the gate's own budget a score-only rule concentrates asks on errors better than the band, so the taxonomy's value is routing irreversible steps, which the offline protocol does not measure.

4.3 RQ4: Cross-Split Generalization

Recalibrating on any single Mind2Web split leaves the gate vacuous in 24/24 split–α combinations: one split's calibration mass cannot certify a threshold under the simultaneous bound, so pooled calibration is required and the per-split excesses of RQ2 are its cost. The fused gate's raw AURC spread over the three splits (0.052 (UI-TARS) / 0.133 (OS-Atlas)) conflates split difficulty with ranking quality; normalizing each leg by its own difficulty band separates the two (Figure 4). On that scale the *level* is not robust: the per-model spread over the three Mind2Web splits is 0.090 (UI-TARS) / 0.148 (OS-Atlas). The *ordering* is stable: the fused gate attains the best AURC of every training-free policy on all 8 legs, raw and normalized.

The certified level does not transfer across splits (24/24 single-split gates are vacuous and 7 conditional slices exceed α), while the policy ordering transfers on all 8 legs.

4.4 Analysis: Where the Coverage Goes

Fusion-weight sensitivity. Scaling any one fusion weight by 0.5–1.5× (16 configurations) keeps the fused gate below the strongest training-free baseline on all 8 legs each time, every recalibrated α=0.3 gate at or below α; its largest group-mean AURC shift (0.0020 / 0.0005) is under the run-to-run floor 0.0144 (Figure 5). Removing one is not local: dropping self-consistency costs $+0.0414$ / $+0.0236$ AURC and makes the UI-TARS α=0.3 gate vacuous, dropping confidence the OS-Atlas

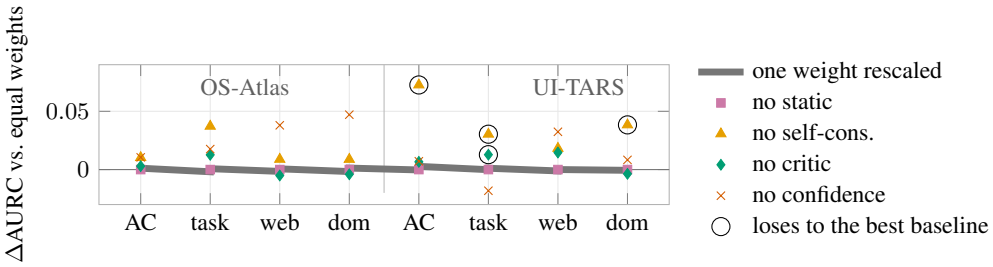


Figure 5: Rescaling a fusion weight barely moves the frontier; removing one does. Change in AURC per leg against the deployed gate: band = the rescalings, markers = the leave-one-out gates.

Table 3: *Top:* the withheld mass is two populations. Shares are of all test proposals in the group, pooled over backbones, intervals Wilson 95%; only the first row is what the *checker-inapplicable* guarantee covers. *Bottom:* three veto policies re-calibrated at α=0.3 on the same anchors, grid and δ, each carrying its own certificate. Policy B withholds only the checker-failing slice.

	AndroidControl		Mind2Web (3 splits)	
Withheld sub-population	n (% of proposals)	error rate [95 % CI]	n (%)	error rate [95 % CI]
An applicable check *fails*	223 (7.5%)	87.0% [81.9, 90.8]	60 (2.5%)	100.0% [94.0, 100.0]
No applicable check	599 (20.2%)	23.7% [20.5, 27.3]	117 (4.8%)	100.0% [96.8, 100.0]
All withheld (as reported)	822 (27.7%)	40.9%	177 (7.3%)	100.0%

Backbone	Veto policy	τ	$n_{\text{cal}}^{\text{acc}}$	coverage	executed risk
OS-Atlas	A: withhold every non-certified proposal *(deployed)*	0.94	380	0.3830	0.2657
	B: withhold only provably-failing proposals	0.91	565	0.5529	0.2845
	C: no veto	0.91	566	0.5544	0.2844
UI-TARS	A: withhold every non-certified proposal *(deployed)*	0.83	646	0.6743	0.2854
	B: withhold only provably-failing proposals	0.83	671	0.6958	0.2777
	C: no veto	0.83	677	0.7006	0.2779

gate, and removal pushes the gate *above* that baseline on 3 of 8 legs without self-consistency (all UI-TARS) and 1 of 8 without the critic removed, a loss the AURC deltas hide. Dropping the static *score* changes no ranking (it is identically 1.0 on certificate-passing proposals); its contribution is the veto alone (next).

Certificate-veto coverage cost. The gate withholds 27.7% of AndroidControl and 7.3% of Mind2Web test proposals from the ACT arm, and that mass is two populations, not one (section 3.3): proposals whose applicable check *fails* (7.5% / 2.5%) and those for which *no* check applies (20.2% / 4.8%), the latter larger at 72.9% / 66.1% of the withheld mass. Only the first is what the guarantee is about, so Table 3 (top) reports them apart. On AndroidControl they err at 87.0% (194/223, 95 % CI [81.9, 90.8]) and 23.7% (142/599, [20.5, 27.3]) respectively, so the per-backbone figure 49.5%/56.2% (OS-Atlas/UI-TARS) is a mixture that, read as one population, overstates the error rate of the larger slice roughly twofold. The 29 of 223 checker-failing proposals graded correct are audited on their recorded trees (Appendix A.2): the gold action fails the same check (10/9, OS-Atlas/UI-TARS), the tree holds no node for the gold target (2/4), or the proposed point lies on no interactive node but inside the grader's click radius (3/1); the checker misread the tree in 0/0. Grading is more permissive than the tree; the static reading is not contradicted. On Mind2Web both are wholly necessarily wrong under the element-match grading (Appendix A), so the overstatement is an AndroidControl finding; the two-population split itself holds on both.

Counterfactually lifting the veto at α=0.3 (same thresholds, scores and ask rule) admits proposals erring at 7.4% (18/244, 95 % CI [4.7, 11.4]) / 8.5% (6/71, [3.9, 17.2]), 244 of 244 (OS-Atlas) and 58 of 71 (UI-TARS) from the unverifiable slice, and adds +9.0/+2.6 pp of pooled coverage, almost all on AndroidControl (+16.3 pp on AndroidControl vs. +0.2 on Mind2Web for OS-Atlas; +4.8 vs. +0.0 for UI-TARS). It improves *both* axes at the reference operating point, executed risk falling from 26.6% (95 % CI [24.0, 29.3]) / 28.5% ([26.5, 30.7]) to 22.9% ([20.7, 25.3]) / 27.8% ([25.8,

29.8]) (-3.66/-0.75 pp), for the reason the decomposition gives: the unverifiable slice is mostly correct, so withholding it raises the error rate of the rest. That veto-off risk is an empirical test-set rate, not a certified one: the bound is stated on $\{\text{certificate holds}\} \cap \{r_t \geq \tau\}$, and lifting it executes a superset the calibration never covered. The veto also costs $+0.041$ (OS-Atlas) / $+0.015$ (UI-TARS) AURC on AndroidControl (episode-paired 95% intervals $[0.028, 0.053]$ / $[0.011, 0.019]$) and moves it by -0.002 $[-0.005, 0.000]$ on Mind2Web.

The certificate-carrying comparison needs recalibration; Table 3 (bottom) gives it. Policy B, which withholds only the checker-failing slice with its own certificate at the same α, δ and grid, recalibrates non-vacuously and recovers $+17.0$ (OS-Atlas) / $+2.1$ pp (UI-TARS) of certified coverage at an executed risk of 0.284 / 0.278 (certified at α=0.3). On this draw it is nearly indistinguishable from no veto (Policy C, $+17.1$ / $+2.6$ pp): the pooled coverage gap is -0.15 pp (95 % CI $[-0.33, -0.03]$) / -0.48 pp (95 % CI $[-0.81, -0.22]$), the risk gap $+0.01$ pp (95 % CI $[-0.13, 0.11]$) / -0.02 pp (95 % CI $[-0.22, 0.15]$), both zero on 6 of the 8 legs (every Mind2Web leg). What separates B from C is what executes: C executes 4 of 208 / 13 of 75 checker-failing proposals, erring at 25.0% (95 % CI $[4.5, 70.0]$) / 30.8% (95 % CI $[12.6, 57.7]$), no worse than the rest it executes; B executes 0 / 0; the near-identity is a property of the draw (Appendix A.1). The recovery has two sources: proposals about which nothing was proven, released directly (262 of 459 (OS-Atlas) and 58 of 58 (UI-TARS) of what B executes and the deployed gate does not); and on OS-Atlas, the other 197 already satisfied the certificate: the veto shrinks the calibration set from 565 accepted anchors to 380, lifting τ from 0.91 to 0.94. The deployed policy stays on a design commitment, that unverified is not verified-safe, and it is the commitment rather than the guarantee that costs the coverage.

Calibration mass predicts vacuity. From the published anchor pool's acceptance and error profile alone, the Bentkus bound at δ/K gives the smallest calibration mass $n^*(\alpha)$ that can certify a threshold (533 anchors for OS-Atlas at α=0.2, 337 anchors for OS-Atlas at α=0.3, and 298 anchors for UI-TARS at α=0.3); subsampling the anchors (Appendix A.1, Figure 7) confirms the registered prediction in 7 of 8 (backbone, α) settings and misses on OS-Atlas at α=0.3, where the predictor is too conservative: it puts the crossing at n^*=337 and the draws put it at 300, and the single-split masses fall below n^* in 24 of 24 split–α–backbone combinations, the 24/24 RQ4 measures.

The veto's coverage cost falls on unverifiable proposals rather than checker-failing ones (262 of 459 (OS-Atlas) and 58 of 58 (UI-TARS) of what the re-calibrated alternative recovers), and again through the calibration mass withheld, which is what makes the certificate vacuous, predicted in 7 of 8 settings.

5 Discussion and Conclusion

VeriAct gates a frozen GUI agent at every step under an episode-grouped PAC certificate on executed-step error and a hard veto. It is scoped to *correctness-risk gating quality* on offline step-level benchmarks, not end-task success or the harm-focused guards such as CORA (Feng et al., 2026).

Limitations. The certificate is marginal: stated at the episode-anchor unit over the pooled calibration mixture, it does not control per-split conditional risk (RQ2). The static certificate covers machine-checkable preconditions, not semantic appropriateness; the taxonomy's uneven coverage makes RQ3's web side measure the conservative default, and at the gate's own ask budget a score-only rule finds errors better than the band (RQ3). The veto costs $+9.0$/$+2.6$ pp of pooled coverage; withholding only the checker-failing slice returns $+17.0$ (OS-Atlas) / $+2.1$ pp (UI-TARS). All numbers come from one seeded paired subset and one test pass; the α and weight sweeps are sensitivity analyses rather than variance estimates; of 20 split re-draws (Appendix A.1) the published one sits at the high-coverage end. Both backbones are 7B models of one family, so the coverage-price result is about GUI-trained agents.

Reproducibility statement

All signals, labels, and gate outcomes derive from a fully offline step-level pipeline over public benchmarks and open-weight frozen agents; no live device execution is performed anywhere in this study. Each step draws k temperature samples alongside one greedy decode (§3), and those

temperature draws are reproducible: the serving engine is constructed without an explicit seed and therefore runs under its default fixed seed, so a re-run of the same cell reproduces the same draws. We verified this by replaying a slice of the OS-Atlas cell: of the nine candidates that overlap the released pool, the eight *sampled* ones came back byte-identical, and their mean log-probabilities agreed to the last bit on all but one, which differed only at the level of floating-point reduction order under a different batch composition, without changing the sampled action. Setting an explicit engine seed instead moves the draws (ten of forty sampled candidates change) while all five greedy candidates in the same slice are unchanged, which is what confirms the seed acts on the sampling path and nothing else. This makes the released pool a convenience rather than a necessity, and it also means a plain re-run measures no run-to-run spread; the spread we do report comes from a second draw under a different engine seed (Appendix A). Everything downstream of the pool needs no seed to reproduce: anchor selection and the calibration/test split are hash-based over the record id, not draws from an RNG whose state depends on call order, and every interval uses a fixed generator. The released pool therefore reproduces every number in this paper exactly. We release it in the supplementary zip as `archives/scored_pool.tar.gz` (`README_supp.md` gives the extraction command), with the gate stack and per-stage artifacts. Every experimental numeral is generated from the released result file by a macro generator, with no hand-transcribed numbers; any macro without a landed artifact renders as a visible `[UNFILLED-MACRO]` marker rather than a value. Appendix A details prompts, decoding parameters, and compute.

AI USE STATEMENT

In this work we disclose generative AI use under the categories of the ICLR 2027 AI policy. Within the required-disclosure categories, we used generative AI tools to help develop the conceptual framework, propose and refine hypotheses, formulate the paper's formal claims and statistical analyses, give feedback on research methodology and experiment design, implement methods (the experiment, analysis and figure-generating code released with this work), clean and reformat datasets, and interpret results. We have *not* used generative AI tools to generate synthetic data sets, assist with translation, support qualitative or thematic data analysis, or write proofs (this paper states no formal proof); the remaining required-disclosure categories are not applicable to this work, which has no human-subject component and no data collection beyond public benchmarks and the outputs of open-weight models. Under the recommended categories, generative AI was additionally used for software code, literature search and analysis, brainstorming, figure preparation, and editing the text of this paper. No experimental result was produced, selected or filtered by a generative AI tool: every number in this paper is emitted by a committed single-writer artifact and reaches the text through an auto-generated macro bridge that admits no hand-typed value. The research questions, the claims and the conclusions are the authors'. We have reviewed all AI-assisted work and take responsibility for the final content, including every claim, number and artifact.

ETHICS STATEMENT

This work gates a GUI agent's actions, so the question it raises is what the gate licenses. Three scoping facts belong here. The study is offline: every signal, label and gate outcome comes from a step-level replay over public benchmarks and frozen open-weight agents, and no live device, account or web session is acted on anywhere in it; a gate evaluated by executing its decisions on real interfaces would be a different study with different obligations. The gate measures correctness-risk rather than harm: §5 says so, and a deployment that read a satisfied certificate as a safety guarantee would be reading it for a property it was never calibrated against; harm-focused guards are complementary and cited there. And the certificate is marginal, stated at the episode-anchor unit over the pooled calibration mixture, so per-split conditional risk is not controlled; RQ2 reports the splits that exceed α rather than leaving the reader to assume they do not. The benchmarks are public and contain no human subject; we release no captured pages, credentials or sessions.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. arXiv:2110.01052, 2022.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations (ICLR)*, 2024.

Stephen Bates, Anastasios N. Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):43:1–43:34, 2021.

Vidmantas Bentkus. On Hoeffding's inequalities. *The Annals of Probability*, 32(2):1650–1673, 2004.

Cong Chen, Kaixiang Ji, Hao Zhong, et al. Gui-shepherd: Reliable process reward and verification for long-sequence gui tasks. arXiv:2509.23738, 2025.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Chaoqun Cui, Jing Huang, Shijing Wang, et al. Agentic reward modeling: Verifying gui agent via progressive trajectory-grounded interaction. arXiv:2602.00575, 2026.

Xiang Deng, Yu Gu, Boyuan Zheng, et al. Mind2web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Yushi Feng, Junye Du, Qifan Wang, et al. Cora: Conformal risk-controlled agents for safeguarded mobile gui automation. arXiv:2604.09155, 2026.

Chao Hao, Shuai Wang, and Kaiwen Zhou. Uncertainty-aware gui agent: Adaptive perception through component recommendation and human-in-the-loop refinement. arXiv:2508.04025, 2025.

Zhiyuan Hu, Shiyun Xiong, Yifan Zhang, et al. Guiding vlm agents with process rewards at inference time for gui navigation. arXiv:2504.16073, 2025.

Divake Kumar, Sina Tayebati, Devashri Naik, Amanda Sofie Rios, Nilesh Ahuja, Omesh Tickoo, Ranganath Krishnan, and Amit Ranjan Trivedi. Cura: Certified runtime alarms for computer-use agents. arXiv:2608.27808, 2026a.

Divake Kumar, Sina Tayebati, Devashri Naik, et al. Uncertainty quantification for computer-use agents: A benchmark across vision-language models and gui grounding datasets. arXiv:2606.25760, 2026b.

Jaewoo Lee, Zaid Khan, Archiki Prasad, et al. A history-aware visually grounded critic for computer use agents. arXiv:2606.11078, 2026.

Jungjae Lee, Dongjae Lee, Chihun Choi, et al. Verisafe agent: Safeguarding mobile gui agent via logic-based action verification. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 2025. doi: 10.1145/3680207.3765248.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on ui control agents. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Quanfeng Lu, Wenqi Shao, Zitao Liu, et al. Gui odyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. arXiv:2406.08451, 2024.

Han Luo, Bingbing Wen, and Lucy Lu Wang. Agentic abstention: Do agents know when to stop instead of act? arXiv:2606.28733, 2026.

Haiyang Mei, Difei Gao, Xiaopeng Wei, Xin Yang, and Mike Zheng Shou. Can i trust you? advancing gui task automation with action trust score. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2025. doi: 10.1145/3746027.3755618.

Jeffery Opoku and David Banahene. Toolchain-crc: Conformal risk control for agentic ai under retrieval and tool-use drift. arXiv:2606.18467, 2026.

Yujia Qin, Yining Ye, Junjie Fang, et al. Ui-tars: Pioneering automated gui interaction with native agents. arXiv:2501.12326, 2025.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, et al. Android in the wild: A large-scale dataset for android device control. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. arXiv:2405.14573, 2024.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Conference on Robot Learning (CoRL)*, 2023.

Corby Rosset, Pratyusha Sharma, Andrew Zhao, Miguel Gonzalez-Fernandez, and Ahmed Awadallah. The art of building verifiers for computer use agents. arXiv:2604.06240, 2026.

Yucheng Shi, Wenhao Yu, Jingyuan Huang, et al. Towards trustworthy gui agents: A survey. arXiv:2503.23434, 2025.

Yuchen Sun, Pei Fu, Shaojie Zhang, et al. Beyond binary: Reframing gui critique as continuous semantic alignment. arXiv:2605.14311, 2026.

Tu Trinh, Mohamed Elfeki, Guangze Luo, et al. Hil-bench (human-in-loop benchmark): Do agents know when to ask for help? arXiv:2604.09408, 2026.

Qingni Wang, Yue Fan, and Xin Eric Wang. Safeground: Know when to trust gui grounding models via uncertainty calibration. arXiv:2602.02419, 2026.

Yuyang Wanyan, Xi Zhang, Haiyang Xu, et al. Look before you leap: A gui-critic-r1 model for pre-operative error diagnosis in gui automation. arXiv:2506.04614, 2025.

Jinbiao Wei, Kangqi Ni, Yilun Zhao, Guo Gan, and Arman Cohan. Step-level optimization for efficient computer-use agents. arXiv:2604.27151, 2026.

Zheng Wu, Heyuan Huang, Xingyu Lou, et al. Verios: Query-driven proactive human-agent-gui interaction for trustworthy os agents. arXiv:2509.07553, 2025.

Yifan Xiao, Zhongjun Wang, Tiantian Yan, Weiwei Li, Dazhong Wu, and Jie Ma. Time series anomaly detection: A hybrid deep learning framework with adaptive thresholding for electricity industry data. In *IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)*, 2025. doi: 10.1109/CSCLOUD66326.2025.00046.

Yifan Xiao, Shijie Li, and Yuhao Ge. RunbookFX: Type- and effect-safe LLM synthesis for executable incident diagnosis and mitigation. *Proceedings of the ACM on Programming Languages (ICFP)*, 2026a. doi: 10.1145/3828675.

Yifan Xiao, Shijie Li, and Yu Huang. CALM-TS: Risk-controlled LLM labeling for time-series via calibrated selective gating. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2026b. doi: 10.1145/3770855.3817859.

Tianbao Xie, Danyang Zhang, Jixuan Chen, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Dexuan Xu, Yanyuan Chen, Zhongyan Chai, Yifan Xiao, Yandong Yan, Weiping Ding, Hanpin Wang, Zhi Jin, Wenpin Jiao, Weihua Yue, Hang Li, and Yu Huang. Knowledge fusion in deep learning-based medical vision-language models: A review. *Information Fusion*, 125:103455, 2026. doi: 10.1016/j.inffus.2025.103455.

Lingfeng Zhang, Yongan Sun, Jinpeng Hu, et al. Webuncertainty: Dual-level uncertainty driven planning and reasoning for autonomous web agent. arXiv:2604.17821, 2026a.

Yuzhe Zhang, Xianwei Xue, Xingyong Wu, et al. Don't act blindly: Robust gui automation via action-effect verification and self-correction. In *Annual Meeting of the Association for Computational Linguistics (ACL), Main Conference*, 2026b.

Shuyan Zhou, Frank F. Xu, Hao Zhu, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2307.13854.

A Reproducibility Details

Models. Frozen backbones: UI-TARS-1.5-7B (Qin et al., 2025) and OS-Atlas-Pro-7B (primary pair, smoke-gated before any batch), with a prompted Qwen2.5-VL-7B agent as the documented fallback; the critic signal uses the official GUI-Critic-R1 release (Wanyan et al., 2025). The fallback was run through the same 20-step smoke gate (pre-registered as a third backbone) and failed it: 20/20 steps completed, but only 89 of 180 candidates parsed as an action (greedy 14/20); the failures are a click without a target point (39), an action keyword outside the OS-Atlas grammar (38) and a type action without text (14). The full run was not started; the candidates and the gate verdict are in the released records. Exact Hugging Face revisions are pinned in the released config.

Decoding. $k = 8$ temperature samples plus one greedy decode per step, temperature 0.7, top-p 0.95, max 256 new tokens, per-token top-logprobs recorded for the confidence signal.

Datasets. AndroidControl official test split (1,543 episodes of the 15,283-episode corpus (Li et al., 2024)) and the test_task / test_website / test_domain splits of the multimodal Mind2Web release (Deng et al., 2023); dataset revisions are pinned in the released `config.yaml`, and the schema probes run in the released `prep_data.py`. The analyzed step set is a seeded paired subset (seed 20260720), drawn once and shared verbatim by both backbones and every gating policy: 2,500 of 8,444 AndroidControl test steps covering all 1,543 test episodes, and 417 / 318 / 1,265 Mind2Web steps from eligible pools of 1,257 / 975 / 3,828. Eligibility excludes rows whose gold target is unresolvable in the provided DOM or whose screenshot is missing (358 Mind2Web rows in total); these exclusions are prep-time findings, logged and counted in the released subset manifest rather than silently discarded.

Compute. Inference-only. A hard budget cap of 160 GPU·h is enforced by a launch guard, and the guard and this paragraph read the same configuration key, so the stated cap is the enforced one. The cap was raised twice during the study, each time on a decision recorded before any run under the raised limit: from 56 to 72 GPU·h for an additional engine seed, and from 72 to 160 for the UI-TARS second draw (Appendix A.1), which ran on a different machine whose throughput is about a third of the main one. Every run except that draw was launched under the 72 GPU·h cap. Per-stage seconds-per-item are measured rather than estimated and recorded in a GPU ledger. Total ledgered spend on the main machine is 63.7 GPU·h (run veriact-20260718): 41.4 for the original programme, 10.0 for the second draw described next and 12.3 for the third. The UI-TARS second draw ran on the other machine, outside this ledger; its chain logs still on that machine record 7 launches spanning 42.9 wall hours (81.2 card·h on one or two cards), a lower bound because earlier launches were overwritten by their relaunches. The gate's own cost is 3.5 s per step for the nine OS-Atlas decodes over 4,500 completed steps and 14.6 s over 3,763 for UI-TARS, plus 3.8 s per scored candidate for the critic (8,986 candidates), one critic call per step; the static checker runs on CPU. The UI-TARS AndroidControl cell was generated partly inside a batch the ledger records only in aggregate, so its per-step rate is reported over the completed cells alone rather than imputed to the rest.

Second and third draws of the OS-Atlas pool. Because the serving engine is seeded, a same-configuration re-run replays the released pool (Reproducibility Statement). To measure how much of the certificate is one draw, we re-generated the entire OS-Atlas cell (all four splits, same configuration, model revision and step subsets) with the engine seed set to 1 and, in a second series, to 2, re-ran the critic stage, and re-aggregated each against the *published* UI-TARS cells, which enter that aggregate as hard links of the released files and came back bit-identical, so any movement on the OS-Atlas side is attributable to the seed. The read-out was fixed before the run (the

registration ships with the code): does the pooled executed-anchor risk stay at or below α? It does, at both non-vacuous settings. At α=0.2 the risk moves from 3.83% (9/235) to 4.66% (11/236), +0.83 pp, leaving a margin of -15.34 pp; at α=0.3 from 26.57% (275/1035) to 27.03% (280/1036), +0.46 pp, margin -2.97 pp. 0 pooled settings exceed α, and α=0.05 and 0.1 remain vacuous in both draws. The third draw lands between them: 4.20% (10/238, +0.37 pp, margin -15.80 pp) at α=0.2 and 27.77% (291/1048, +1.20 pp, margin -2.23 pp) at α=0.3, with 0 pooled exceedances, α=0.05 and 0.1 vacuous, slice violations 3, and its largest per-split shift +4.9 pp on test_task at α=0.3 (n_{act} 57→59). Per-split conditional slices move more, as their sizes predict: the largest shift is +7.1 pp on test_website at α=0.3 (n_{act} 44→49), and the OS-Atlas slice-violation count goes 4→3. These are two resamples of one backbone on the same machine; UI-TARS has one further draw on a different machine (Appendix A.1), and three draws do not turn a single-draw certificate into a distribution over draws.

These 3 draws are the paper's only measurement of run-to-run drift, and they vary exactly one thing. Greedy decoding does not consult the engine seed and every decision is graded on the greedy proposal, so across the three pairs 0 proposed actions and 0 correctness labels move across the three pairs, and 3 of the four signals (the static certificate, the critic, and confidence) are identical; the drift reaches the fused score through self-consistency alone, whose value moves on 1,088 of 2,702 test records on the worst pair. Temperature sampling is this pipeline's only source of run-to-run randomness, so this is the complete drift for a fixed configuration rather than a floor on it, and it is drift through one signal of four. On the RQ1 axis the four-leg mean gain is 11.5%, 10.8% and 11.3% (largest pairwise 0.66 pp), with a largest single-leg shift of 1.98 pp (M2W task, draw 2 vs draw 3), the eight-leg headline mean gain (8.9%) exceeding that single-leg floor by $4.4\times$, a largest AURC shift of 0.0144 and largest AUROC shifts of 0.0542 (self-consistency) and 0.0358 (fused). At α=0.3, 2,367 of 2,702 gate decisions and 1,144 of 1,428 episodes are unchanged on the worst of the three pairs (draw 2 vs draw 3). Both floors are measured on the four OS-Atlas legs; the UI-TARS draw of Appendix A.1 changes the machine as well as the seed, so it does not enter the floor, and its per-leg AURC shift is not measured; the floor is an OS-Atlas floor. The two-draw figures are kept for the record: mean gain 11.5% to 10.8% (-0.66 pp), largest leg shift 1.05 pp, AURC shift 0.0080, 1,088 of 2,702 records moved, 2,376 of 2,702 gate decisions and 1,151 of 1,428 episodes are unchanged; the eight-leg headline exceeded that floor by $8.4\times$ and the weakest leg by $1.8\times$, the ratio the three-draw floor withdraws (0 proposed actions and 0 correctness labels move, and 3 of the four signals (the static certificate, the critic, and confidence) are identical).

Calibration mass and the predicted n^*. The subsampling study behind Figure 7 draws, for each mass n and repetition r, the n anchors with the smallest $\text{sha256}(\texttt{record_id}\,|\,r)$, so it uses no random-number state, and calibrates each draw with the same calibrator the deployed gate uses. The predictor takes the full pool's acceptance rate and error rate at every threshold of the fixed grid, floors the acceptance at mass n, rounds the implied error count, and asks whether any threshold's Bentkus bound at δ/K falls at or below α; n^* is the smallest mass from which that holds for every larger mass. Both the predictor and the verdict were registered before the draws were taken, and the registration is released with the code.

Ask-budget tail. Beyond its knee the OS-Atlas curve saturates (+4 pp more asks buy only 0.2 pp); the UI-TARS curve does not — a further +14 pp of asks buys 7.9 pp more, reaching 9.8% residual risk at a 58% ask rate before flattening (Figure 3; knee in section 4.2).

Fusion-weight scale sweep. Over the 16 configurations of section 4.4, scores are renormalized, so a perturbed signal's effective weight moves between 0.14 and 0.33. The largest group-mean AURC move is 0.0020 on AndroidControl and 0.0005 on Mind2Web (1.4% / 0.1% of the equal-weight value), and the largest certified coverage move is 4.7 pp (OS-Atlas) / 2.5 pp (UI-TARS) (12.4% / 3.7% relative). These are measured maxima at this section's precision rather than bounds. 4 of the configurations are provably inert: they rescale a score that is constant on every proposal reaching the ACT arm, so they map the fused score affinely and can move no ranking.

Irreversibility taxonomy v0. Rule classes: *send* (message/mail composition targets), *delete* (removal affordances), *pay* (payment confirmation), *submit* (form/order submission), *overwrite* (destructive edits); the per-dataset action-type mapping is released with the gate code.

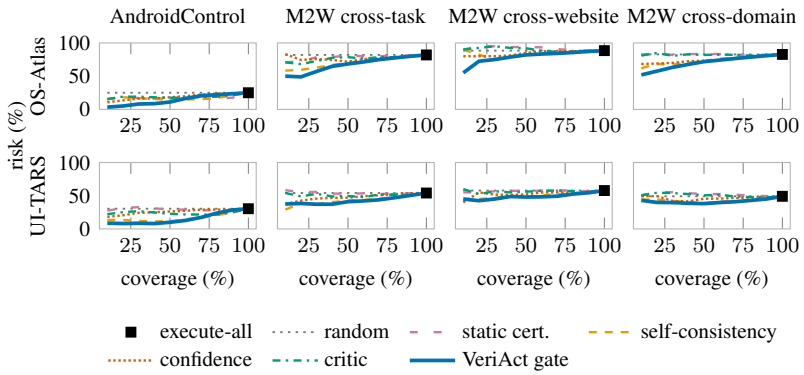


Figure 6: Step-level risk–coverage frontiers underlying Table 1, per backbone (rows) and dataset (columns), from the same per-run artifacts. Black square: execute-all. Gray dotted: random-ranking reference. Lower dominates; the fused gate (solid) sits at or below the strongest per-point training-free baseline on 70 of 80 coverage grid points (worst exceedance $+0.083$ risk).

Table 4: Run-to-run drift, per leg: the 3 draws of the OS-Atlas cell. Draw 1 is the published pool; draws 2 and 3 set engine seed 1 and 2. "gain vs best TF" is the RQ1 gain on that leg; the headline 8.9% averages it over the eight legs of both backbones. AUROC is over all analyzed test records of the leg, positive class = correct.

OS-Atlas leg	draw	AURC (ours)	gain vs best TF (%)	AUROC S_{sc}	AUROC fused
AndroidControl	1	0.1317	23.8	0.6943	0.8428
	2	0.1335	22.7	0.7041	0.8424
	3	0.1326	23.2	0.6988	0.8447
M2W task	1	0.6478	6.8	0.7479	0.8210
	2	0.6398	6.0	0.7774	0.8388
	3	0.6487	8.0	0.7279	0.8098
M2W website	1	0.7811	6.6	0.7139	0.7604
	2	0.7875	6.3	0.6923	0.7423
	3	0.7731	6.5	0.7465	0.7782
M2W domain	1	0.6721	8.7	0.7216	0.7828
	2	0.6760	8.1	0.7253	0.7834
	3	0.6803	7.6	0.7024	0.7695

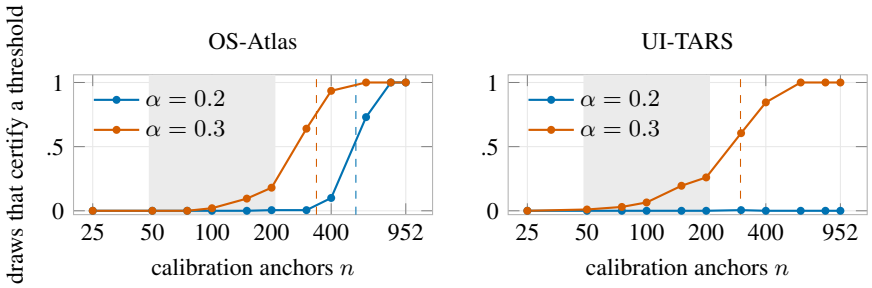


Figure 7: Vacuity is a calibration-mass effect the bound predicts. Each point: the fraction of 200 hash-selected subsamples of the 952 anchors certifying a threshold; dashed rules n^*, from the full pool alone, registered before these draws. The band spans RQ4's single-split masses (209, 70, 48).

Reporting discipline. Every numeral in the paper is produced by a macro generator from the single released result file; tables and figures are regenerated by the scripts listed in the supplementary `README_supp.md`, and certificate violations over re-drawn calibration/test splits are counted by `split_resampling.py`.

A.1 SPLIT RESAMPLING, THE UI-TARS SECOND DRAW, AND WHAT THE VETO WITHHOLDS

Re-drawing the calibration/test split. The published split assigns episodes to calibration or test by a seeded hash. We re-drew that hash 20 times on CPU, re-ran the calibration and the veto ablation on each draw, and read the same quantities as Table 2; the published seed re-run reproduces the published files field for field. Vacuity is stable: OS-Atlas certifies nothing in 20, 17, 0, and 0 of 20 at $\alpha = 0.05, 0.1, 0.2,$ and 0.3, UI-TARS in 20, 20, 20, and 0 of 20 at $\alpha = 0.05, 0.1, 0.2,$ and 0.3, and at α=0.3 every draw certifies (0 of 20 / 0 of 20). At that level the certified threshold spans 0.94–0.97 / 0.80–0.89 (OS-Atlas / UI-TARS), certified coverage 0.206–0.394 / 0.488–0.739, and executed risk 0.145–0.264 / 0.221–0.306. The certificate is violated in 2 of the 63 non-vacuous (backbone, α, split) cells, worst margin -0.52 pp at UI-TARS at $\alpha = 0.3$; the published draw lies at the high-coverage end of the OS-Atlas range. Across draws the re-calibrated policies recover 12.4–30.2 / 0.4–15.0 pp (Policy B) and 12.4–30.2 / 0.4–13.6 pp (no veto) of certified coverage at α=0.3; on two draws B's coverage exceeds C's, because B's calibration set differs, so "B $\approx$ C" is a property of a draw and not an identity. On the published draw the per-leg B$-$C gap (episode-cluster bootstrap, 2000 resamples) is largest at -0.88 pp (95 % CI $[-1.43, -0.40]$), UI-TARS on AndroidControl in coverage and -0.24 pp (95 % CI $[-0.64, 0.09]$), UI-TARS on AndroidControl in risk.

The UI-TARS second draw. UI-TARS was resampled once with engine seed 1 on a different machine, so this draw changes the sampling seed and the hardware together and is read as a second draw rather than as a seed effect. At α=0.3 its executed risk is 26.40% (434/1644) against 28.54% (520/1822) on the published draw, a shift of -2.14 pp that widens the margin from -1.46 to -3.60 pp; 0 pooled settings violate, α=0.05 and 0.1 and 0.2 stay vacuous, and the Mind2Web slice-violation count goes 3$\rightarrow$3. The largest per-split shift is -3.2 pp on test_domain at α=0.3 (n_{act} 594$\rightarrow$533). Re-running the veto ablation on this draw gives coverage A 0.6743, B 0.6958, C 0.7006 (published draw); A 0.6084, B 0.6247, C 0.6791 (second draw) and executed risk A 0.2854, B 0.2777, C 0.2779 (published draw); A 0.2640, B 0.2589, C 0.2676 (second draw); Policy B recovers $+2.15$ pp (published draw) and $+1.63$ pp (second draw), no veto $+2.63$ pp (published draw) and $+7.07$ pp (second draw), and B's recovery comes 58 of 58 by direct release and 0 through the re-calibrated threshold (published draw); 44 of 44 and 0 (second draw). The OS-Atlas rows of the same ablation are identical to the published ones, as they must be: that backbone was not re-drawn here.

What the veto withholds. The withheld proposals of Table 3 are mostly actions that name no element, so the checker has nothing to verify: scroll, wait, open-app, back, home, terminate, or an action the parser rejected. 750 of 822 (91.2%) withheld on AndroidControl and 149 of 177 (84.2%) on Mind2Web are of this kind. The element-bearing remainder is the part a check could in principle have covered; it errs at 66.7% ($48/72$, 95 % CI $[55.1, 76.5]$) on AndroidControl and 100.0% ($28/28$, 95 % CI $[87.9, 100.0]$) on Mind2Web, against execute-all error rates of 27.7% and 67.5%. On AndroidControl the element-free slice is graded by action type and direction, so it is not necessarily wrong, as it is on Mind2Web, where every withheld proposal is wrong.

A.2 REVIEW-ROUND CONTROLS: PAIRED INTERVALS, THE VETO AUDIT, THE MATCHED-BUDGET ASK

Episode-paired AURC intervals. Table 5 gives, per leg, the difference $\mathrm{AURC}(\text{gate}) - \mathrm{AURC}(\text{row})$ with a 95% percentile interval from 5,000 resamples of test episodes; every policy on a leg is scored on the same resample, and the two backbones share the resample sequence, so the intervals are paired across policies and across backbones. Point estimates agree with Table 1 to numerical precision. The *veto* column is $\mathrm{AURC}(\text{gate}) - \mathrm{AURC}(\text{fused score without the veto})$: AURC has no threshold, so the re-calibrated no-veto policy and the plain fused ranking coincide. The interval excludes zero against execute-all, random, the static certificate and the critic on 8 of 8 legs; against the UQ row on 5 of 8 legs (including it on UI-TARS on M2W cross-task, UI-TARS on M2W cross-website, and UI-TARS on M2W cross-domain); for the veto on 2 of 8 legs, both AndroidControl; it includes zero on OS-Atlas on M2W cross-task, UI-TARS on M2W cross-task, OS-Atlas on M2W cross-website, UI-TARS on M2W cross-website, OS-Atlas on M2W cross-domain, and UI-TARS on M2W cross-domain, where the veto's group mean is $[-0.005, 0.000]$, so on Mind2Web the veto does not move the ranking.

Table 5: Episode-paired AURC differences, gate minus row (negative: gate lower), with 95% intervals; the veto column is with minus without the veto (positive: the veto costs AURC). Group rows pool the legs of a dataset group over both backbones.

Leg	Exec.-all	Random	Static	UQ	Critic	Veto
OS-Atlas, AC	-0.119 $[-0.135, -0.103]$	-0.118 $[-0.134, -0.103]$	-0.041 $[-0.065, -0.018]$	-0.042 $[-0.066, -0.018]$	-0.056 $[-0.080, -0.034]$	$+0.041$ $[0.028, 0.053]$
OS-Atlas, M2W task	-0.172 $[-0.219, -0.112]$	-0.168 $[-0.216, -0.109]$	-0.144 $[-0.210, -0.067]$	-0.047 $[-0.089, -0.004]$	-0.119 $[-0.180, -0.047]$	0.000 $[-0.001, 0.000]$
OS-Atlas, M2W web.	-0.103 $[-0.168, -0.047]$	-0.099 $[-0.164, -0.042]$	-0.133 $[-0.220, -0.058]$	-0.055 $[-0.110, -0.007]$	-0.116 $[-0.197, -0.047]$	-0.007 $[-0.024, 0.000]$
OS-Atlas, M2W domain	-0.155 $[-0.189, -0.119]$	-0.154 $[-0.188, -0.118]$	-0.165 $[-0.207, -0.120]$	-0.064 $[-0.086, -0.040]$	-0.155 $[-0.197, -0.112]$	-0.005 $[-0.014, 0.000]$
UI-TARS, AC	-0.161 $[-0.179, -0.143]$	-0.161 $[-0.179, -0.143]$	-0.149 $[-0.179, -0.120]$	-0.021 $[-0.040, -0.002]$	-0.092 $[-0.118, -0.067]$	$+0.015$ $[0.011, 0.019]$
UI-TARS, M2W task	-0.119 $[-0.174, -0.065]$	-0.117 $[-0.172, -0.062]$	-0.123 $[-0.206, -0.040]$	-0.008 $[-0.069, 0.051]$	-0.093 $[-0.174, -0.016]$	0.000 $[-0.001, 0.000]$
UI-TARS, M2W web.	-0.117 $[-0.170, -0.060]$	-0.114 $[-0.167, -0.057]$	-0.118 $[-0.201, -0.026]$	-0.039 $[-0.084, 0.006]$	-0.111 $[-0.200, -0.017]$	0.000 $[-0.001, 0.000]$
UI-TARS, M2W domain	-0.082 $[-0.117, -0.046]$	-0.082 $[-0.116, -0.045]$	-0.117 $[-0.165, -0.068]$	-0.012 $[-0.047, 0.024]$	-0.106 $[-0.154, -0.057]$	0.000 $[-0.001, 0.000]$
AndroidControl (pooled)	-0.140 $[-0.152, -0.127]$	-0.140 $[-0.152, -0.127]$	-0.095 $[-0.117, -0.074]$	-0.032 $[-0.047, -0.016]$	-0.074 $[-0.095, -0.054]$	$+0.028$ $[0.021, 0.035]$
Mind2Web (pooled)	-0.125 $[-0.146, -0.102]$	-0.122 $[-0.144, -0.100]$	-0.133 $[-0.165, -0.100]$	-0.037 $[-0.055, -0.019]$	-0.117 $[-0.146, -0.085]$	-0.002 $[-0.005, 0.000]$

The checker-failing proposals graded correct. Of the AndroidControl test proposals that fail an applicable check, the checker's reasons are unparseable output 133, a click on a non-interactive node 62, text entry with no focused editable field 18, a disabled target 7, and no visible node at the point 3 (of 223). 29 of 223 of them are graded correct against gold: 15 for OS-Atlas (2.6% of that backbone's withheld mass) and 14 for UI-TARS (5.7%), together 3.5% of everything withheld on the split. We fetched the recorded accessibility tree of each, re-ran the checker on it, and required the recorded report to be reproduced field for field before classifying; all were. Four classes were fixed in advance. (1) The checker misread a premise of the tree: 0/0 (OS-Atlas/UI-TARS). (2) The gold action fails the same check on the same tree: 10/9. (3) The tree holds no node for the gold target: 2/4, one a click with no node under the gold point and five text entries into a tree with no editable node. (4) Other: 3/1, all of one kind: gold passes every check, the proposed point falls outside the gold element's box and on no interactive node, and the grader accepts it because it lies within its click radius of 0.14 of the screen, at normalized distances of 12.2 to 31.4 against a radius of 140. In 21 of 29 the failing check is a click on a non-interactive node. None of the 29 of 223 is evidence against the static reading; each is a step where the element-match grading accepts what the tree does not support.

The matched-budget ask controls. Table 6 compares the deployed gate at α=0.3 with two controls that ask exactly as many AndroidControl test steps as the gate does, 72 of 1,482 (4.9%) for OS-Atlas and 80 of 1,482 (5.4%) for UI-TARS, and execute every other step: asking the k lowest fused scores, and asking k steps drawn uniformly (seed 0). Both controls abstain from nothing, so their coverage is not matched to the gate's; the risk differences (-10.7 pp (95 % CI $[-12.9, -8.5]$) / -14.0 pp (95 % CI $[-16.1, -11.8]$) against the score rule, -14.6 pp (95 % CI $[-17.0, -12.3]$) / -17.5 pp (95 % CI $[-19.8, -15.2]$) against random, episode-paired intervals) are the price of the ABSTAIN arm, and the coverage differences (-47.0 pp (95 % CI $[-49.7, -44.3]$) / -32.7 pp (95 % CI $[-35.1, -30.3]$) against either control) are its cost. The column that isolates the ASK arm is the error rate among asked steps. The match holds at α=0.3 only, since τ and hence k change with α.

Table 6: Three policies asking the same number of AndroidControl test steps at α=0.3. Coverage and executed risk are over all steps; the last column is the error rate of the asked steps, the quantity the ASK arm is judged on.

Backbone	Policy	Coverage	Executed risk	Error among asked
OS-Atlas	Gate (band + taxonomy)	48.1%	10.5%	43.1%
	Ask k lowest scores	95.1%	21.2%	100.0%
	Ask k at random	95.1%	25.1%	23.6%
UI-TARS	Gate (band + taxonomy)	61.9%	13.3%	53.8%
	Ask k lowest scores	94.6%	27.2%	86.2%
	Ask k at random	94.6%	30.8%	23.8%