

Authentic but Not Neutral: Selective Evidence in Agent Leaderboards

Xiaoxuan Li Bangbang Liu Ying Zhao

Shanghai Artificial Intelligence Laboratory

{lixiaoxuan, liubangbang, zhaoying}@pjlab.org.cn

Abstract

Large language models are increasingly being used as agents that interact with tools and environments to carry out complex tasks. This development has prompted a growing range of benchmarks and leaderboards for comparing agent performance. Yet the evidence behind these comparisons is not always fully disclosed. Even when every disclosed record is authentic and correctly scored, a reporter with access to the complete ledger can choose a favorable subset. We study whether rank claims remain informative under such adaptive release. An authenticated disclosure frame fixes systems, denominators, bundle bounds, and the canonical claim universe before selection. Worst-case completion keeps every reported edge sound, but a budgeted reporter can still choose which comparisons become certifiable. This separates the soundness of reported comparisons from neutrality in what gets reported. Across four public agent-evaluation ledgers, target-directed release produces positive attention gaps for most eligible targets at intermediate budgets; the effect depends on budget and bundle layout and disappears under an equal-per-system quota in one ledger. We derive an exact polynomial algorithm for the unit-cost target problem and prove weak NP-completeness under heterogeneous costs. These experiments are structural stress tests, not evidence that benchmark maintainers select evidence strategically. They show that certificate soundness does not imply disclosure neutrality.

1 Introduction

Large language models are increasingly being used as agents that interact with tools and environments to carry out complex tasks, from software engineering to desktop computer use (Zan et al., 2025; Xie et al., 2024). This development has prompted a growing range of benchmarks and leaderboards for comparing agent performance.

Yet the evidence behind these comparisons cannot always be released in full: hidden tasks protect benchmark integrity, enterprise environments contain confidential state, and long-horizon logs are expensive to store (Patel et al., 2026b). These are legitimate constraints, not defects.

But they create a scientific question. Suppose every disclosed record is authentic and correctly scored. Is a truthful partial release sufficient to justify the rank claims in a report? Truthfulness constrains what released evidence says, not which evidence is released. A sender who sees a complete private ledger may disclose bundles that support favorable comparisons and leave adverse bundles unopened.

Prior work addresses related parts of evaluation reporting. Benchmark audits test tasks, environments, and graders (Wang et al., 2026); outcome-evidence audits ask whether retained records determine a benchmark’s native success claim (Gao and Zhou, 2026); and rollout-reporting standards preserve records and document omissions for reanalysis (Masters et al., 2026). We take the bounded contribution of each evidence bundle as given and study the choice of which truthful bundles to release.

The central distinction is between *certificate soundness* and *disclosure neutrality*. We define an authenticated disclosure frame that is fixed before evidence selection. It commits the relevant denominators, an exhaustive universe of candidate rank claims, and an outer bound on each unopened bundle’s contribution. A checker reports a directional claim only if it holds for every admissible completion. This rule remains sound even when the sender sees the full ledger and chooses a subset adaptively. Yet soundness does not force the supported subset to be neutral: the sender can still decide which valid comparisons become resolvable under a limited disclosure budget.

We measure this discretion by comparing the

target’s supported wins and losses under target-directed and target-blind release policies.

Four public ledgers illustrate the effect without assuming strategic behavior by maintainers. On Terminal-Bench (16 systems, 80 run-shard bundles), strategic release achieves a median attention gap of 0.400 over target-blind coverage at budget 40; 14 of 15 targets benefit (Merrill et al., 2026). Multi-SWE shows a contrasting boundary: all 13 promotable targets benefit at five bundles, but gaps vanish once global coverage saturates the 45 supportable claims (Zan et al., 2025). The effect is budget- and frame-dependent.

Two non-coding replications vary domain and bundle granularity. OSWorld-Verified provides ten desktop-agent systems on a 357-task common frame; all ten targets have positive gaps at a 40-domain-bundle cap on both step-budget surfaces (Xie et al., 2024). An archived Assistant-Bench/HAL release contains all 15 active Browser-Use leaderboard entries, giving 495 task-level bundles over a common 33-task set, 105 all-pair and 14 adjacent strict claims (Yoran et al., 2024; Princeton Language and Intelligence, 2026). At a 50% cap, 14 of 15 targets show positive all-pair gaps (median 0.286, maximum 1.000); an equal-per-system quota eliminates all gaps (Section 5).

We formalize when rank claims remain sound after adaptive selection and show why a finite bound on omitted evidence is necessary. Across four benchmark families, we measure how selection changes the comparisons that receive attention. We also give an exact polynomial algorithm for the unit-cost target problem and show weak NP-completeness under heterogeneous costs. Authentic evidence and correct scoring are therefore insufficient to make a partial release neutral.

2 Related Work

Agent evaluation validity and evidence. Agent benchmarks now span multi-environment interaction (Liu et al., 2024), desktop control (Xie et al., 2024), application and API interaction (Yao et al., 2025), and software engineering (Jimenez et al., 2024; Zan et al., 2025). These settings differ in task state, execution harnesses, graders, and the public records retained after evaluation. We study a reporting property shared across these benchmark families rather than introduce a new task metric.

Calls for rigorous agent evaluation emphasize

standardized protocols, costs, holdouts, and reproducibility rather than accuracy alone (Kapoor et al., 2025). ABA audits task specifications, environments, grading logic, and ground truth, then validates detected defects through expert review and upstream reports (Wang et al., 2026). Gao and Zhou study a closer but different object: whether stored records decide the benchmark’s native outcome claim. Their Pass/Fail/Unknown ledger induces fixed-record score bounds and rank summaries under missing outcomes (Gao and Zhou, 2026). Our analysis begins after each bundle has an authenticated bounded contribution. It neither decides Unknown outcomes nor attributes failures; instead, it makes the evidence-selection rule adversarial.

Rollout Cards preserve rollout records, reporting rules, and documented omissions (Masters et al., 2026); Evaluation Cards document additional details about how an evaluation was produced (Ghosh et al., 2026). Both can record the disclosure frame proposed here.

Rank uncertainty and partial identification.

Worst-case bounds for bounded missing outcomes are classical partial identification (Manski, 2003). Gao and Zhou apply endpoint separation to outcome-evidence Unknowns. Statistical rank intervals instead quantify task-sampling or hierarchical uncertainty when observations exist (Neuhof and Benjamini, 2026). The certificate here is finite-ledger and frame-relative, supplying neither confidence intervals nor repeated-run or out-of-sample guarantees. Classical interval arithmetic is applied to an authenticated completion set whose opened bundles are chosen strategically after the ledger is observed.

Strategic and verifiable disclosure. Classical voluntary-disclosure models analyze how withholding verifiable information changes receiver inference (Milgrom, 1981; Dye, 1985). Bayesian persuasion studies sender-designed information structures (Kamenica and Gentzkow, 2011); verifiable evidence models study what a sender can disclose and a receiver can check (Glazer and Rubinstein, 2004; Bull and Watson, 2004). We borrow the adversarial sender/receiver lens without priors or equilibrium behavior; the receiver uses a worst-case finite-ledger certificate, and the leaderboard structure yields a tractable unit-cost subcase.

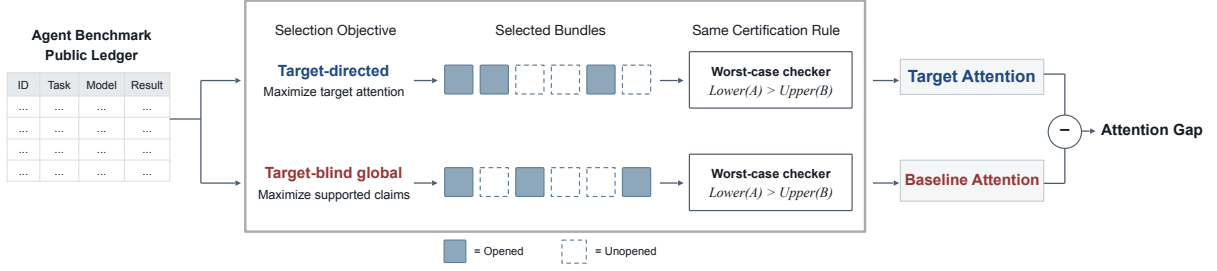


Figure 1: Different disclosure objectives can certify different comparisons. Both policies use the same complete ledger, fixed frame, and budget cap. Each release is checked separately against all admissible completions of unopened evidence. The attention gap is target-directed minus global-policy attention (Equation 3). Filled and dashed bundles denote opened and unopened evidence; the selected subsets are schematic.

3 Certification Before Selection

3.1 Authenticated disclosure frame

Let a finite evaluation ledger be partitioned into evidence bundles $\mathcal{E} = \{e_1, \dots, e_m\}$. Each bundle belongs to one system and has a bounded contribution to that system’s numerator. A sender observes the realized ledger and selects $\mathcal{S} \subseteq \mathcal{E}$ for disclosure, possibly as an arbitrary function of all bundle contents. Opened bundles reveal exact contributions; unopened bundles retain authenticated outer bounds.

The *disclosure frame* is fixed before selection: it contains (i) system denominators N_j , (ii) bundle identities and contribution bounds, and (iii) a canonical claim universe $\mathcal{P}$. An independent record must establish these fields before the sender sees the outcomes. The checker then matches each opened value to that record and retains the stated bound for every unopened bundle. The frame is a trust premise; evidence registered only after selection cannot establish which trials were eligible to remain hidden.

The benchmark operator or an independent evaluator may define the bundles, but must do so before outcomes are available. Our results are conditional on that choice: they protect claims after the frame is fixed, not against a frame chosen to favor a system.

Table 1 makes the contract explicit. The checker needs authenticated metadata that fixes the adverse completion, not raw trajectories. These fields do not replay a benchmark’s scoring implementation.

The distinction matters operationally. A certificate can disclose an exact aggregate contribution for an opened bundle while retaining a committed range for an unopened one; it need not expose the underlying trajectory. Conversely, an authentic

Committed field	Checker requirement and failure if absent
System denominator	Normalize every completion; without it the omitted population can grow.
Bundle IDs and owners	Fix selectable units; without them evidence can be added after outcomes are known.
Contribution bounds	Bound unopened numerators; without them the adverse completion is unbounded.
Claim universe and direction	One status per committed claim; without it unfavorable claims can disappear.
Budget and bundle costs	Common resource scale; without it coverage levels are incomparable.

Table 1: Authenticated disclosure-frame fields. Opened values are additionally checked against the committed bundle identity. Scorer conformance is a separate prerequisite, not an output of this certificate.

score file without a committed denominator, bundle universe, and adverse bound cannot establish what evidence was eligible to remain hidden. Certification therefore concerns a partial report relative to its frame. It does not check the scorer, benchmark tasks, or execution harness.

For system j , let $r_j(\mathcal{S})$ denote the exact numerator contribution of opened bundles. Let the total unopened contribution lie in $[L_j(\mathcal{S}), U_j(\mathcal{S})]$. Its admissible score interval is

$$\Theta_j(\mathcal{S}) = \left[\frac{r_j(\mathcal{S}) + L_j(\mathcal{S})}{N_j}, \frac{r_j(\mathcal{S}) + U_j(\mathcal{S})}{N_j} \right]. \quad (1)$$

A directional claim $A > B$ is *frame-supported* when it holds under every admissible completion of unopened evidence—an identification statement about the committed finite ledger, not a hypothesis test.

Worked binary example. Suppose A and B each have $N = 10$ authenticated binary trials. Revealing seven A successes and four B failures gives $\Theta_A = [.7, 1]$ and $\Theta_B = [0, .6]$, so $A > B$ is supported ($7 + 4 \geq 11$). With only three B failures opened, $\Theta_B = [0, .7]$ and the claim is unresolved. Both verdicts remain sound even if the sender selected bundles after seeing all 20 outcomes; authentication prevents silently changing the denominator or the selectable universe.

For every $p \in \mathcal{P}$, the certificate contains the selected bundles, recomputed endpoints, and exactly one status: SUPPORTED or UNRESOLVED. A checker rejects unregistered bundles, changed bounds, over-budget releases, and missing claim rows.

3.2 Tight support and adaptive soundness

Theorem 1 (Tight rectangular support). *Suppose the admissible completion set is rectangular across systems. Then $A > B$ is frame-supported at $\mathcal{S}$ if and only if*

$$\frac{r_A(\mathcal{S}) + L_A(\mathcal{S})}{N_A} > \frac{r_B(\mathcal{S}) + U_B(\mathcal{S})}{N_B}. \quad (2)$$

The left side is A 's lowest admissible score, the right side B 's highest; necessity follows because rectangularity makes the joint endpoint completion admissible. With coupled completions, Equation 2 remains sufficient but may not be necessary.

The endpoint-separation argument is standard partial identification (Manski, 2003; Gao and Zhou, 2026); here, bounded contributions are fixed before the sender adaptively chooses which bundles remain unopened.

Corollary 1 (Adaptive-selection soundness). *If the frame is authenticated before selection and opened contributions are correct, a checker applying Equation 2 is sound for every disclosure rule, including a sender that observes the complete ledger and selects $\mathcal{S}$ adaptively.*

The theorem holds pointwise for every subset; opening a bundle replaces its bound by an exact contribution, so score intervals shrink and a supported claim remains supported after further disclosure. Added evidence cannot invalidate an already supported edge, but the sender still chooses the order in which different edges first become supported.

Rectangularity fits a ledger in which each system's remaining bounded contribution can vary independently, as in the binary constructions below.

If benchmark rules couple systems, tasks, or unopened bundles, the endpoint comparison remains conservative but can leave some truly identified claims unresolved. A deployment can use a tighter completion set when it can authenticate those couplings; it must then expose enough structure for the checker to validate the corresponding bound.

3.3 Why authentication is necessary

Truthful rows alone do not authenticate the complement of a release.

Theorem 2 (No bound, no nontrivial rank certificate). *Suppose a verifier sees only truthful disclosed binary outcomes and has no externally authenticated finite bound on evidence that may remain omitted. No disclosure-only verifier can soundly support a directional mean claim for every compatible ledger in general.*

One compatible ledger can append undisclosed successes to A and failures to B ; another can append the reverse. The verifier sees the same disclosed view, so any directional verdict is unsound in one world. The result does not require public raw traces; it requires some trusted mechanism that bounds the adverse completion.

The checker requires exactly one status per committed $p \in \mathcal{P}$, preventing deletion of unresolved comparisons. Section 4 models the remaining discretion over which resolved edges receive attention.

4 Residual Attention Discretion

Sound certificates prevent unsupported rank edges. They do not determine which supported edges appear under a limited release. We now formalize this remaining choice.

4.1 A declared attention model

Let the canonical claim direction follow a fixed strict public ordering. For a target system j , let D_j and H_j be the lower- and higher-ranked systems in $\mathcal{P}$, and let $d_j = |D_j| + |H_j|$. Given disclosure set $\mathcal{S}$, define

$$I_j(\mathcal{S}) = \frac{W_j(\mathcal{S}) - L_j(\mathcal{S})}{d_j}, \quad (3)$$

where W_j counts supported claims $j > k$ for $k \in D_j$, and L_j counts supported claims $h > j$ for $h \in H_j$. Unresolved comparisons contribute zero. Thus I_j is net certified dominance normalized by the target's committed incident comparisons.

Equation 3 defines a reader model in which resolved comparisons receive attention and unresolved ones do not. It does not claim to estimate latent rank, user belief, or causal persuasion. A win-only alternative W_j/d_j is also reported. A sender seeking to highlight j chooses at most B truthful bundles to maximize I_j ; because the sender observes the complete ledger, this offline optimum upper-bounds any sequential policy with the same frame and budget.

The target’s public score and the claim directions are held fixed throughout. The sender cannot reverse an ordering, alter a disclosed value, or make an unresolved edge count as a win; it can only allocate a finite disclosure budget across bundles. Consequently, a positive strategic-minus-baseline gap does not mean that the target’s performance changed. It means that the same complete ledger admits two truthful partial certificates that make different subsets of the already committed comparisons salient.

4.2 Fixed-denominator binary structure

For the main structural result, assume that every system has the same authenticated denominator N ; opened outcomes are exact and binary; each bundle belongs to one system; each system’s bundles partition its N rows without overlap or gaps; every bundle costs one; and at most B bundles may be opened. For claim $a > b$, let $s_a(\mathcal{S})$ be disclosed successes for a and $f_b(\mathcal{S})$ disclosed failures for b . Worst-case completion support reduces to

$$s_a(\mathcal{S}) + f_b(\mathcal{S}) \geq N + 1. \quad (4)$$

Theorem 3 (Exact target-attention algorithm). *Under the assumptions above, a disclosure maximizing I_j can be found in polynomial time by: (i) enumerating the number q of target bundles opened; (ii) taking the q target bundles with most successes; (iii) for each $k \in D_j$, computing the shortest descending-failure prefix of k ’s bundles that satisfies Equation 4; and (iv) buying the cheapest such opponent prefixes within the remaining budget. The best solution over q is globally optimal.*

No bundle from a higher system can help an outgoing edge, and without higher-system bundles no incoming loss is certified. At fixed q , replacing a target bundle by one with more successes weakly helps every outgoing edge; each lower opponent

controls one independent edge with a minimum-cost prefix. Proofs appear in Appendix A.

The only shared decision is therefore how many target bundles to open. Once q is fixed, the target contribution is common to every outgoing claim, and each opponent contributes to one independent threshold. Sorting turns each opponent’s choice into a shortest feasible prefix, after which the remaining budget is spent on the cheapest prefixes. This separation is why the unit-cost case is tractable; heterogeneous bundle costs restore a knapsack choice even when only one directional claim is considered.

If M is the number of bundles and $d = |D_j|$, sorting takes $O(M \log M)$ and enumerating target counts takes $O(m_j d \log d)$ after prefix sums. Tractability here is specific to the leaderboard structure; general disclosure allocation need not be easy.

Theorem 4 (Heterogeneous-cost boundary). *With positive integer costs and compact binary-encoded bundle counts, deciding whether one target win can be supported within budget is weakly NP-complete, even for two systems, binary outcomes, a common denominator, and one system per bundle.*

The reduction is from 0–1 knapsack: item values and costs map to opponent bundle failures and costs, a fixed target bundle supplies $N + 1 - V$ successes, and one edge is supported exactly when selected items reach value V within budget C . Thus the hardness boundary is caused by unequal disclosure costs, not by interval arithmetic alone. The qualifier *weakly* matters: a pseudo-polynomial dynamic program exists when counts are small.

4.3 Policies compared

Figure 1 summarizes the primary comparison. The strategic policy maximizes Equation 3. Baselines are uniform random (B bundles without replacement), target-blind global (maximizes supported claims), utility-matched target-blind (maximizes $\sum_j W_j/d_j$), and quota-balanced ($\lfloor B/J \rfloor$ or $\lceil B/J \rceil$ bundles per system). A fixed rule resolves ties for every optimized policy. The policies use the same frame, costs, budget, and checker; only their selection objective differs. The global baseline measures target-blind certificate coverage. The utility-matched baseline tests whether the gap is explained by optimizing a different aggregate utility, and the quota control tests whether partial

B	Global edges	Positive	Gap Q1/med./Q3	Max
5	0	0/15	.000/.000/.000	.000
10	1	13/15	.067/.067/.071	.071
20	6	14/15	.133/.214/.214	.286
40	26	14/15	.233/.400/.483	.533

Table 2: Terminal-Bench strategic-minus-global attention gaps over all eligible targets. Global edges are total claims supported by the target-blind policy.

release alone creates a gap. Random disclosure provides a descriptive reference. Since B is an upper bound, a policy may leave capacity unused; a 100% cap permits full disclosure but does not require it.

5 Evidence

Selective release is simulated from complete public records; every partial certificate is tested against the full bounded ledger. No benchmark maintainer is assumed to have selected evidence strategically.

5.1 Terminal-Bench: natural run shards

Terminal-Bench Core 0.1.1 provides 16 systems with 400 fixed-frame binary trials each, split into five 80-row run shards (80 unit-cost bundles) (Merrill et al., 2026; Terminal-Bench Team, 2026). Fifteen systems have at least one outgoing claim and are eligible as targets. The canonical claim universe contains 117 strict pairwise directions; the complete ledger supports 87. At each budget $B \in \{5, 10, 20, 40\}$, the target-specific algorithm and a general integer program agree in all cases. Random baselines use 500 draws per target and budget.

Table 2 and Figure 2a show a budget-dependent effect. At 40 bundles, the median gap is 0.400 and 14/15 targets are positive; the largest-gap system reaches $I_j=0.533$ (eight wins, no losses) versus no supported incident edge under target-blind coverage.

The low-budget rows make the comparison informative rather than tautological. At $B=5$, neither policy certifies an edge and every gap is zero. At $B=10$, target-blind coverage reaches only one of 117 canonical directions, yet 13 of 15 targets can obtain a positive gap under target-directed release. The gap then grows with budget while the global policy also resolves more edges. This pattern separates a regime in which no certificate is informative from one in which the same limited bud-

get can be allocated toward different rank claims.

Against the matched-utility baseline, budget-40 win-only gaps remain positive for 11 of 15 targets. The quota-balanced policy supports no strict edge through $B=40$ (at most three of five shards per system) and first reaches all 87 supportable edges at $B=80$. Zero soundness violations are observed across all subsets.

5.2 Multi-SWE: a saturation boundary

Multi-SWE provides one result file per system per language surface (Zan et al., 2025; Multi-SWE-bench Team, 2026). Four verified surfaces (C++, Java, JavaScript, TypeScript) with 27 systems each yield 108 bundles and 1,336 within-surface claims; the complete ledger supports only 45.

Only 13 of the 108 surface-system targets are *promotable* (at least one outgoing edge supported by the complete ledger); non-promotable targets can gain a positive gap merely by staying silent while the global policy reveals incoming losses, so they are excluded. At five bundles, all 13 promotable targets have positive gaps (median 0.080). At 20 bundles the global policy saturates all 45 supportable claims and the median gap becomes zero. The effect is therefore not automatic: it depends on frame, layout, and budget.

The exclusion removes a trivial advantage in which a target receives a better net attention score only because it has no full-ledger-supported win to promote and avoids opening evidence for an incoming loss. The remaining 13 targets each have at least one claim that could be certified in their favor under complete disclosure. Their common low-budget gap therefore reflects selective resolution of genuine outgoing comparisons, not silence about a target that cannot be supported at all.

Figure 3 shows that Multi-SWE’s global policy saturates all 45 supportable claims after 20 of 108 bundles, whereas Terminal-Bench resolves only 26 of 87 at budget 40 and AssistantBench reaches all 105 only at full disclosure.

5.3 Non-coding replications

OSWorld-Verified provides ten desktop-agent systems under 15-step and 100-step configurations with 357 common tasks in ten application domains, giving 100 bundles and 45 strict claims per surface (Xie et al., 2024). At cap 40, all ten targets have positive gaps (medians 0.333 and 0.389; Table 3).

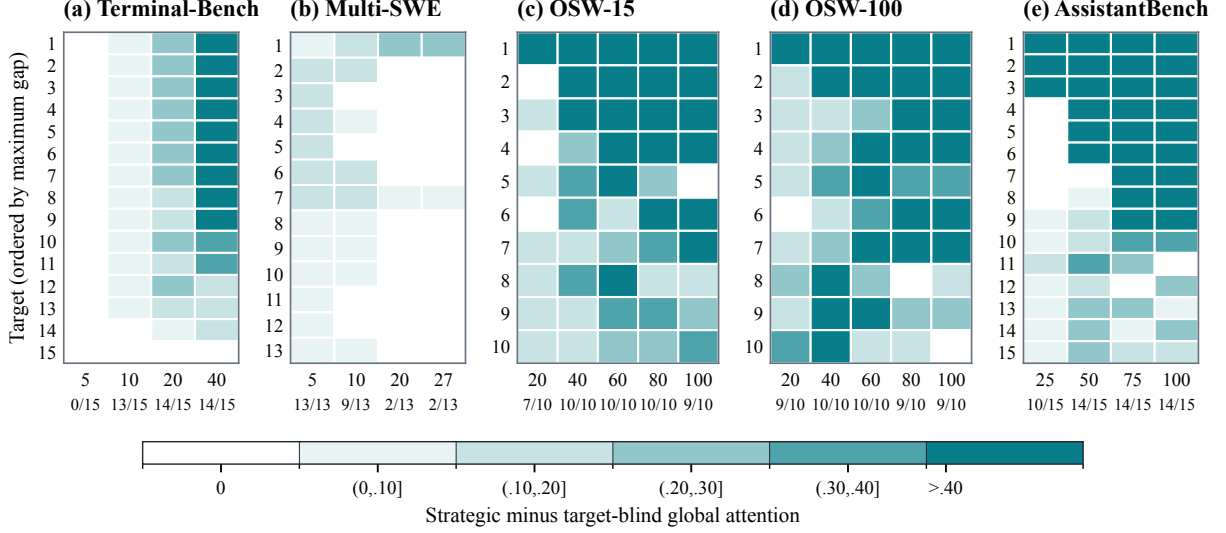


Figure 2: Target-level strategic-minus-global attention gaps. Panels show Terminal-Bench (15 targets, 80 run shards), Multi-SWE (13 promotable targets, 108 files), OSW-15/100 (ten targets and 100 domain bundles each), and AssistantBench (15 active entries, 495 score-record bundles, 105 all-pair claims). Rows are ordered by maximum gap; counts give positive gaps. B is a cap and may not be exhausted; AssistantBench labels report fractions of its 495 bundles, whereas the other panels report absolute caps. All panels share the same scale.

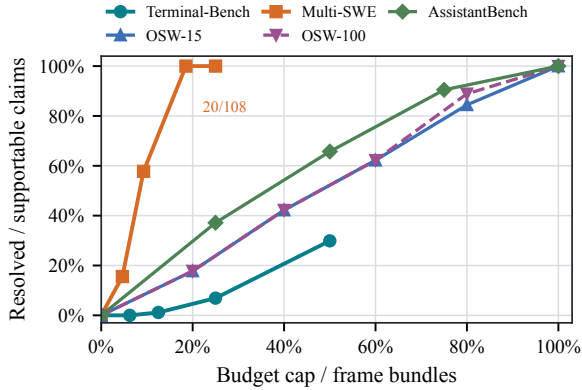


Figure 3: Target-blind coverage versus budget cap, normalized within each frame; bundle fractions are not comparable cross-frame costs. Full-ledger supportable/canonical claim counts are 87/117 (Terminal-Bench), 45/1,336 (Multi-SWE), 45/45 per OSWorld surface, and 105/105 (AssistantBench).

The two OSWorld surfaces keep the common task frame and domain-level bundle layout while changing the step budget. Positive gaps on both surfaces show that the phenomenon is not tied to a binary coding benchmark or to one particular failure representation. They do not establish an ordering between the two surfaces: bundle costs and attention values are interpreted only within the authenticated frame used for that replication.

An archived AssistantBench/HAL release contains all 15 active Browser-Use entries (Yoran et al., 2024; Princeton Language and Intelligence,

Frame	Cap	Global	Positive	Med./max
OSW-15	40	19/45	10/10	.333/.889
OSW-100	40	19/45	10/10	.389/1.000
AssistantBench	247	69/105	14/15	.286/1.000

Table 3: Non-coding replications. “Global” is target-blind certificate coverage; positive counts targets with a strategic-minus-global gap above zero.

2026). Each run contributes 33 ordered score records in $[0, 1]$; one record is one bundle, giving 495 bundles. HAL publishes the score sequence separately from task identifiers, so we keep the public order and do not infer a score-to-task mapping. Recomputed means match the published values for all 15 systems. The systems induce 105 all-pair and 14 adjacent strict claims. At a 50% cap ($B=247$), 14 of 15 targets show positive all-pair gaps (median 0.286, max 1.000); 13 of 15 are positive on adjacent claims (median 0.500, max 1.000). A balanced-quota control certifies no claim at the same cap, eliminating all gaps—confirming that the effect is not automatic. Zero soundness violations are observed.

The quota result also exposes a coverage cost. At the 50% cap, balanced exposure leaves every pair of score intervals overlapping, so it removes the attention gap by resolving no comparison. The control therefore limits selection discretion at the price of an uninformative certificate at this budget.

The four frames differ in selectable unit—run shard, result file, application-domain block, and ordered score record—yet each permits a target-directed release to redirect certified attention under some intermediate budget. This is a structural result, not evidence that maintainers have behaved strategically. Multi-SWE loses the gap once target-blind coverage saturates its supportable claims, and AssistantBench’s equal-per-system quota produces no certified claim at the same 50% cap. The gap therefore depends on the frame, bundle layout, budget, and release policy; partial evidence by itself does not produce it. Nor do the results imply that finer bundles create more discretion: Multi-SWE saturates with coarse result-file bundles, while AssistantBench has no gap under a balanced quota at the stated cap.

A reader can verify the claims in a partial certificate but cannot infer from their truth alone whether the supported subset represents all claims permitted by the frame. Reporting every claim’s status and overall certificate coverage makes that selection visible. These experiments do not estimate how often real reports face a binding disclosure budget or whether any maintainer selected evidence strategically.

5.4 Reproducibility

The released materials record the source data, bundle definitions, costs, denominators, claims, and tie rules. All 22 automated tests pass; Appendix D summarizes the checks.

6 Discussion

What should a report authenticate? The theorem requires a bound on omitted evidence, not public raw traces. A benchmark can register the trial population, denominators, and score ranges before results are available. A record created only after evidence has been selected does not establish the population from which that selection was made. Rollout Cards provide one place to report these fields.

Fixing the frame does not make the opened bundles representative of task difficulty or of the complete ledger. The certificate instead shows exactly which declared comparisons remain resolved under adverse completions. It can be checked from bounded bundle contributions, so raw trajectories may remain private; the additional reporting burden is the frame and the status of each declared

comparison.

Interpreting attention. Equation 3 assumes unresolved comparisons receive no attention and weighs incident claims equally. The win-only sensitivity preserves the main Terminal-Bench pattern, but alternative utilities (top- k , adjacent ranks) can be inserted into the general integer program, though Theorem 3 may no longer apply.

7 Conclusion

Truthful partial evidence need not be neutrally selected. An authenticated frame keeps certificates sound under adaptive selection while leaving the sender free to determine which valid comparisons resolve. Across four public ledgers, target-directed release reallocates support; Multi-SWE also shows that this discretion contracts after global coverage saturates. Exact unit-cost optimization and heterogeneous-cost hardness characterize the computation. Partial releases should report the frame and certificate coverage.

8 Limitations

Certificates are conditional on an authenticated frame and do not verify task validity, scorer conformance, or ledger completeness. The bound applies to a fixed finite ledger; sampling uncertainty, repeated-run variation, and future model versions fall outside its scope. None of the four experiments estimates prevalence, alleges strategic behavior, or claims statistical significance. The attention score is a declared reporting utility, not a model of reader psychology, and the unit-cost experiments abstract from real disclosure costs.

9 Ethical Considerations

All data come from released benchmark records; no private trajectories are exposed and no strategic behavior is attributed to any benchmark maintainer. The adaptive sender models a worst-case reporting choice rather than observed practice. Publishing the status of every declared comparison can expose selective presentation without revealing sensitive trajectory contents. An unresolved claim should not be presented as a tie or a loss: it means only that the disclosed evidence cannot certify the declared direction. Reports should identify who fixed the frame and when, and distinguish unresolved claims from contradicted ones. These labels reduce the risk that selective coverage is mistaken for a complete ranking.

References

- Jesse Bull and Joel Watson. 2004. [Evidence disclosure and verifiability](#). *Journal of Economic Theory*, 118(1):1–31.
- Ronald A. Dye. 1985. [Disclosure of nonproprietary information](#). *Journal of Accounting Research*, 23(1):123–145.
- Shanshan Gao and Liyi Zhou. 2026. [Can agent benchmarks support their scores? evidence-supported bounds for interactive-agent evaluation](#). *Preprint*, arXiv:2605.10448.
- Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.
- Avijit Ghosh et al. 2026. [Evaluation cards: An interpretive layer for AI evaluation reporting](#). *Preprint*, arXiv:2606.09809.
- Jacob Glazer and Ariel Rubinstein. 2004. [On optimal rules of persuasion](#). *Econometrica*, 72(6):1715–1736.
- Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. [SWE-bench: Can language models resolve real-world GitHub issues?](#) In *International Conference on Learning Representations*, pages 54107–54157.
- Emir Kamenica and Matthew Gentzkow. 2011. [Bayesian persuasion](#). *American Economic Review*, 101(6):2590–2615.
- Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. 2025. [AI agents that matter](#). *Transactions on Machine Learning Research*.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2024. [AgentBench: Evaluating LLMs as agents](#). In *International Conference on Learning Representations*, pages 52989–53046.
- Charles F. Manski. 2003. *Partial Identification of Probability Distributions*. Springer.
- Charlie Masters, Ziyuan Liu, and Stefano V. Albrecht. 2026. [Rollout cards: A reproducibility standard for agent research](#). *Preprint*, arXiv:2605.12131.
- Mike A. Merrill et al. 2026. [Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces](#). *Preprint*, arXiv:2601.11868.
- Paul R. Milgrom. 1981. [Good news and bad news: Representation theorems and applications](#). *The Bell Journal of Economics*, 12(2):380–391.
- Multi-SWE-bench Team. 2026. [Multi-SWE-bench Experiments: Predictions, logs, trajectories, and results](#). Official artifact repository.
- Bitya Neuhof and Yuval Benjamini. 2026. [Rank intervals for leaderboards: A hierarchical framework for model evaluation](#). *Preprint*, arXiv:2606.08679.
- Dhaval Patel, Chathurangi Shyalika, Suryanarayana Reddy Yarrabothula, Ling Yue, Shuxin Lin, Nianjun Zhou, and James Rayfield. 2026a. [Results and retrospective analysis of the CODS 2025 AssetOpsBench challenge](#). *Preprint*, arXiv:2605.08518.
- Dhaval Patel, Nianjun Zhou, Shuxin Lin, James Rayfield, Chathurangi Shyalika, and Suryanarayana Reddy Yarrabothula. 2026b. [AssetOpsBench-Live: Privacy-aware online evaluation of multi-agent performance in industrial operations](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(48):41658–41660.
- Princeton Language and Intelligence. 2026. [HAL AssistantBench leaderboard and public traces](#). Holistic Agent Leaderboard. Accessed 2026-08-05.
- Terminal-Bench Team. 2026. [Terminal-Bench Core 0.1.1 registry and leaderboard](#). Official benchmark registry.
- Junlin Wang, Federico Bianchi, Shang Zhu, Fan Nie, Yongchan Kwon, Bhuwan Dhingra, and James Zou. 2026. [Automated benchmark auditing for AI agents and large language models](#). *Preprint*, arXiv:2605.26079.
- Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. [OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 52040–52094.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2025. [\$\tau\$ -bench: A benchmark for tool-agent-user interaction in real-world domains](#). In *International Conference on Learning Representations*, pages 9965–10017.
- Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. 2024. [AssistantBench: Can web agents solve realistic and time-consuming tasks?](#) *Preprint*, arXiv:2407.15711.
- Daoguang Zan et al. 2025. [Multi-SWE-bench: A multilingual benchmark for issue resolving](#). *Preprint*, arXiv:2504.02605.

A Complete Proofs

A.1 Tight support and monotonicity

Proof of Theorem 1. For every admissible completion,

$$\theta_A \geq \frac{r_A + L_A}{N_A} \quad \text{and} \quad \theta_B \leq \frac{r_B + U_B}{N_B}.$$

If Equation 2 holds, transitivity gives $\theta_A > \theta_B$ for every completion. Conversely, if the inequality does not hold, rectangularity makes the joint completion $x_A = L_A$, $x_B = U_B$ admissible. Under that completion $\theta_A \leq \theta_B$, so the strict claim fails. $\square$

Proof of Corollary 1. Theorem 1 holds for each set $\mathcal{S}$ separately and makes no assumption about the distribution or rule that generated it. Hence every edge reported after adaptive selection still holds under all completions admitted by the pre-authenticated frame. $\square$

If $\mathcal{S} \subseteq \mathcal{T}$ and opening a bundle only replaces a bound by its exact value, the lower endpoint of A cannot decrease and the upper endpoint of B cannot increase. Endpoint separation at $\mathcal{S}$ therefore implies separation at $\mathcal{T}$.

A.2 Impossibility without a finite adverse bound

Proof of Theorem 2. Fix a finite disclosed view for A and B . For arbitrary M , construct two compatible complete ledgers. In W^+ append M undisclosed successes to A and M failures to B . In W^- append M failures to A and M successes to B . The disclosed view is identical. As M grows, the means approach one and zero respectively in W^+ , with the limits reversed in W^- . For sufficiently large finite M , $A > B$ holds in W^+ and fails in W^- . A verifier supporting either direction on the shared view is unsound in one compatible ledger. $\square$

A.3 Exact unit-cost target optimization

We prove Theorem 3 through four exchange arguments.

Lemma 1. *There is an optimum containing no bundle from any $h \in H_j$.*

Proof. A bundle from h cannot contribute to an outgoing target edge $j > k$. Its successes can only help the incoming edge $h > j$. Removing every selected higher-system bundle preserves target wins,

can only remove target losses, and weakly reduces cost. $\square$

Lemma 2. *An optimum selected as above has no supported target loss.*

Proof. No higher-system bundle is open, so disclosed successes for every $h \in H_j$ are zero. Since j has exactly N binary rows, disclosed target failures from any subset are at most N . Thus $s_h + f_j \leq N < N + 1$. $\square$

Lemma 3. *At a fixed number q of target bundles, the q bundles with largest success counts weakly dominate every other target subset.*

Proof. After the previous lemma, target failures cannot create a supported loss. Target bundles affect each outgoing edge only through total target successes. Exchanging a selected bundle for an unselected bundle with more successes weakly raises the left side of Equation 4 for every lower opponent. $\square$

Lemma 4. *After fixing target bundles, each lower opponent has an independent minimum unit prefix cost.*

Proof. Opponent k affects target-incident claims only through $j > k$. Among r bundles from k , the maximum failure sum comes from the r largest failure counts. The first sorted prefix reaching the residual threshold is therefore minimum. Different opponents use disjoint bundles and do not alter one another's thresholds. $\square$

For fixed q , every supportable outgoing edge now has an independent integer cost and equal benefit one. Taking finite costs in ascending order maximizes the number of wins within $B - q$. Enumerating every feasible q includes the target-bundle count of a global optimum, proving Theorem 3.

A.4 Heterogeneous-cost reduction

Proof of Theorem 4. Membership in NP is immediate. Reduce from decision 0–1 knapsack (Garey and Johnson, 1979). Let positive item values and costs be v_i, c_i , with target value V , budget C , and $N = \sum_i v_i$. Create target system j and lower system k . A cost-one target bundle contains $N+1-V$ successes. A second target bundle costs $C+2$ and contains the remaining $V-1$ failures, so j has exactly N rows. Each item creates an opponent

bundle with v_i failures and cost c_i ; these also total N rows. Set disclosure budget $C + 1$.

The expensive target bundle is unavailable. Without the cheap target bundle, opponent failures are at most N , below the strict threshold. With it, the edge is supported exactly when

$$(N + 1 - V) + \sum_{i \in Q} v_i \geq N + 1,$$

or $\sum_{i \in Q} v_i \geq V$. The total disclosure cost is at most $C + 1$ exactly when $\sum_{i \in Q} c_i \leq C$. Counts and costs are binary-encoded, so the construction has polynomial encoding length. The inherited knapsack hardness is weak, consistent with a pseudo-polynomial dynamic program. $\square$

B Checking a Partial Release

The theory does not require a particular storage or signing system. It requires an independent record, created before outcomes are available, that fixes the systems, denominators, bundle boundaries, contribution bounds, costs, and claim universe. Opened bundles must be matched to that record. The checker rejects unregistered or repeated bundles, changed bounds or costs, and releases that exceed the budget.

For each system, the checker replaces the bound of an opened bundle with its observed contribution and retains the original bound for every unopened bundle. It then recomputes the score intervals and assigns SUPPORTED or UNRESOLVED to every declared comparison. Requiring a status for every comparison prevents a sender from hiding an inconvenient unresolved row.

This procedure assumes that the registered population is complete. It does not verify the benchmark tasks, rerun the scorer, or rule out executions omitted before registration. Those questions require separate evidence from the benchmark operator. An unresolved comparison remains a valid output of the checker rather than a failed certificate.

C Extended Empirical Results

C.1 Terminal-Bench baselines

Table 4 places the three target-blind baselines on a common budget grid. “Utility” maximizes summed normalized certified wins; its comparison uses the same win-only target objective. All optimization statuses are optimal.

Metric	$B = 5$	$B = 10$	$B = 20$	$B = 40$
Global edges	0	1	6	26
Utility edges	0	1	6	26
Quota edges	0	0	0	0
Positive / global	0/15	13/15	14/15	14/15
Median gap	.000	.067	.214	.400
Positive / utility	0/15	12/15	12/15	11/15
Matched median	.000	.067	.200	.133

Table 4: Terminal-Bench baseline summary.

B	Per-system quota	Supported edges
5	0–1	0
10	0–1	0
20	1–2	0
40	2–3	0
48	3	0
56	3–4	7
64	4	7
72	4–5	50
80	5	87

Table 5: Quota-balanced coverage sweep. The policy maximizes total supported claims within the indicated floor/ceiling quota.

The random baseline uses seed 20260803 plus the budget and 500 independent uniform subsets per target-budget cell. The structured algorithm and generic integer program agree in all 45 target cases at budgets 10, 20, and 40. Every selected-subset supported edge is also supported by the complete bounded ledger, yielding zero soundness violations across all 60 target-budget runs.

C.2 Terminal-Bench targets at $B = 40$

Tables 6 and 7 jointly report every eligible target. Compact IDs keep the metric table readable; the released row-level table retains the full system names.

ID	Date	System / model
T1	1013	Ante / Claude 4.5 Sonnet
T2	0923	Droid / Claude 4.1 Opus
T3	1001	Droid / Claude 4.5 Sonnet
T4	0926	Ante / Claude 4 Sonnet
T5	0924	Droid / GPT-5
T6	1010	Chaterm / Claude 4.5 Sonnet
T7	0923	Droid / Claude 4 Sonnet
T8	1017	DeepAgent Desktop / Claude 4.5
T9	0911	Chaterm / Claude 4 Sonnet
T10	1016	Apex Agent / GPT-5
T11	0711	OpenHands / Claude 4 Sonnet
T12	1012	Alpha / Claude 4.5 Sonnet
T13	0902	Orchestrator / Claude 4 Sonnet
T14	0811	Cursor CLI / Claude 4 Sonnet
T15	0902	Orchestrator / Qwen-3-Coder-480B

Table 6: Compact target IDs. Dates are 2025 MMDD.

ID	d_j	Target	Global	Random	Gap	Win gap
T1	15	.533	.400	.007	.133	.067
T2	14	.533	.400	.006	.133	.533
T3	13	.533	.000	.006	.533	.533
T4	12	.533	.200	.003	.333	.133
T5	10	.500	.000	.003	.500	.500
T6	10	.500	.000	.002	.500	.500
T7	8	.500	.071	.003	.429	.143
T8	8	.500	.000	.002	.500	.500
T9	6	.357	.000	.001	.357	.357
T10	6	.357	-.071	.000	.429	.071
T11	5	.133	-.267	-.001	.400	.000
T12	4	.133	.000	-.002	.133	.133
T13	3	.133	-.267	-.001	.400	.000
T14	2	.000	-.467	-.011	.467	.000
T15	1	.000	.000	-.015	.000	.000

Table 7: All eligible Terminal-Bench targets at $B = 40$. Target and Global are I_j values; Random is the mean over 500 subsets; Win gap uses the matched target-blind win utility.

C.3 Multi-SWE sensitivity

The four included surfaces each contain 27 system files with a common declared denominator. C, Go, and Rust are excluded before optimization because their public files do not share a constant declared denominator across systems. The claim universe retains all 1,336 strict directions within included surfaces. JavaScript and TypeScript contain no promotable target under the complete bounded ledger. Across all 108 targets, positive gaps at larger budgets can also arise from strategic silence avoiding an incoming loss; the main paper excludes these silence-only cases from the promotion interpretation.

B	Global	Positive	Gap Q1/med./Q3	Win-only
5	7	13/13	.077/.080/.154	12/13
10	26	9/13	.000/.077/.115	7/13
20	45	2/13	.000/.000/.000	0/13
27	45	2/13	.000/.000/.000	0/13

Table 8: Multi-SWE results over the 13 promotable targets.

Surface	N	Prom.	$B = 5$	$B = 10$	$B = 20$	$B = 27$
C++	129	2	2/.077	2/.077	0/.000	0/.000
Java	128	11	11/.080	6/.077	2/.000	2/.000
JavaScript	356	0	—	—	—	—
TypeScript	224	0	—	—	—	—

Table 9: Per-surface sensitivity. Budget cells give positive targets/median gap; the promotable denominator appears in the third column.

C.4 OSWorld-Verified replication

We use the archived OSWorld-Verified source revision. Ten systems have public task-level ledgers under both 15-step and 100-step configurations. Their global intersection contains 357 task identities in ten native application domains: Chrome, GIMP, LibreOffice Calc, Impress, and Writer, multi-application tasks, operating-system tasks, Thunderbird, VLC, and VS Code. Each system-domain pair is one unit-cost bundle. Step configurations remain separate surfaces, each with 100 bundles and 45 exhaustive strict pair claims.

Positive gaps are not solely silence-only effects. At cap 40, one 15-step target certifies one favorable incident comparison and no loss, with a 0.667 gap over global coverage. On the 100-step surface, one target certifies seven favorable comparisons and no loss, with a 0.556 gap. Every partial-certificate edge is supported by the corresponding complete 357-task ledger.

C.5 AssistantBench/HAL replication

We use an archived public HAL Browser-Use release containing 15 active leaderboard entries (Table 11); historical and withdrawn rows are excluded before selection. Each run has 33 ordered scores in $[0, 1]$, giving $15 \times 33 = 495$ bundles. Every run covers the same 33 tasks, but HAL publishes the score sequence separately from task identifiers. We therefore preserve the public score order without assigning scores to individual tasks. Recomputed means match the published value for every system. The 15 systems induce 105 all-pair and 14 adjacent strict claims, all supported by the complete bounded ledger.

Surface	Cap	Global	Positive	Median	Max
100-step	20	8/45	9/10	.111	.889
100-step	40	19/45	10/10	.389	1.000
100-step	60	28/45	10/10	.500	1.000
100-step	80	40/45	9/10	.500	1.000
100-step	100	45/45	9/10	.500	1.000
15-step	20	8/45	7/10	.111	.889
15-step	40	19/45	10/10	.333	.889
15-step	60	28/45	10/10	.444	1.000
15-step	80	38/45	10/10	.389	1.000
15-step	100	45/45	9/10	.500	1.000

Table 10: OSWorld-Verified application-domain disclosure sweep. The cap is an upper bound: selective omission remains permitted when capacity is unused.

Rank	System ID	Model / configuration
1	o3-medium	o3 / medium
2	gpt-5-medium	gpt-5 / medium
3	o4-mini-low	o4-mini / low
4	o4-mini-high	o4-mini / high
5	gpt-4.1	gpt-4.1 / default
6	claude-3.7-sonnet	claude-3.7-sonnet / default
7	claude-opus-4.1-high	claude-opus-4.1 / high
8	claude-3.7-sonnet-high	claude-3.7-sonnet / high
9	claude-sonnet-4.5-high	claude-sonnet-4.5 / high
10	deepseek-r1-may	deepseek-r1 / may-2025
11	claude-opus-4.1	claude-opus-4.1 / default
12	claude-sonnet-4.5	claude-sonnet-4.5 / default
13	gemini-2.0-flash	gemini-2.0-flash / default
14	deepseek-v3	deepseek-v3 / default
15	deepseek-r1-jan	deepseek-r1 / january-2025

Table 11: The 15 active Browser-Use entries in the archived HAL release. Ranks follow the published order, which equals the recomputed score order.

At the 50% cap, the target-blind global policy resolves 69/105 all-pair claims; 14 of 15 targets receive a positive target-directed gap over that baseline (median 0.286, maximum 1.000). On adjacent claims, 13 of 15 targets are positive (median 0.500, maximum 1.000) against global coverage of 7/14. A balanced-quota control exposes the boundary (Table 13): under an exact budget with per-system counts differing by at most one, the same 50% cap certifies no all-pair or adjacent claim and no target has a positive gap. Bundle-access constraints can therefore eliminate measured discretion entirely, so the headline gaps are not an automatic consequence of partial release. The result applies to these 15 active runs and does not describe the full history of the live HAL leaderboard.

Fraction	Cap	Scope	Global	Positive	Median	Max
25%	123	all-pair	39/105	10/15	.071	1.000
50%	247	all-pair	69/105	14/15	.286	1.000
75%	371	all-pair	95/105	14/15	.500	1.000
100%	495	all-pair	105/105	14/15	.500	1.000
25%	123	adjacent	3/14	13/15	.500	1.000
50%	247	adjacent	7/14	13/15	.500	1.000
75%	371	adjacent	11/14	14/15	.500	1.000
100%	495	adjacent	14/14	14/15	.500	1.000

Table 12: AssistantBench disclosure sweep on the active-15 frame. “Cap” is the bundle count at the listed fraction of 495; it is an upper bound, not a quota. “Global” is target-blind coverage; positive counts targets with a positive strategic gap.

Fraction	Cap	Scope	Global	Positive	Median	Max
50%	247	all-pair	0/105	0/15	.000	.000
50%	247	adjacent	0/14	0/15	.000	.000

Table 13: Equal-quota sensitivity at the 50% cap. Per-system counts differ by at most one; no claim resolves and no target has a positive gap.

D Reproducibility

The released materials contain source tables, bundle definitions, row-level results, and analysis code. Twenty-two tests check optimization, score reconstruction, benchmark-specific claims, and every reported supported edge.

E Applying the Analysis to Other Benchmarks

The same analysis applies beyond the four benchmarks only when the population, bundle boundaries, score bounds, and comparison set are fixed before outcomes are inspected.

E.1 Population and bundle construction

The frame identifies the benchmark version, systems, trials, and denominator. Exclusion rules precede evidence selection, and repeated runs remain separate trials. Bundles follow outcome-independent release units, such as a run shard, task group, or per-system file. Mixed-system data require a prior partition or coupled bounds. Our four frames use run shards, per-system files, application domains, and ordered score records, respectively.

E.2 Scores, repetitions, and claim universes

Binary success is only a special case. For a bounded trial score $y_i \in [a_i, b_i]$, a bundle commits the sum of its lower and upper trial bounds; opening the bundle replaces this outer interval with

its authenticated exact contribution. Partial-credit benchmarks can therefore use the same checker when the metric range and aggregation rule are fixed. Unbounded metrics require a scientifically justified finite envelope or fall under Theorem 2.

The claim universe follows a rule declared before viewing the ledger, such as all ordered pairs, all strict directions in a published leaderboard, or a fixed set of comparisons with a baseline. The certificate concerns this finite ledger only. Repeated-run variability, task-sampling uncertainty, and generalization to future tasks require separate statistical analysis.

E.3 What to report

A partial release should identify the evaluated systems and trials, the bundle boundaries and score ranges, the disclosure budget, the selected bundles, and the status of every declared comparison. This information makes omissions and claim coverage visible without revealing unopened trajectories.

E.4 Variation in trace size

AssetOpsBench-Live reports 2,196 successful execution traces (Patel et al., 2026a). Their sent-token counts vary widely (Table 14), motivating heterogeneous disclosure costs. The release does not connect each trace to the outcomes needed for a rank analysis.

Statistic	Sent tokens
Total	243.1 million
Median per trace	53,906.5
95th percentile	426,347
Maximum	1,189,583

Table 14: Sent-token counts in the AssetOpsBench-Live summary.