AUDITING INFERRED TOOL-CALL DEPENDENCIES AS PROXIES FOR EXECUTION IMPACT

Anonymous authors
Paper under double-blind review

ABSTRACT

When dependency violations are used to score reordered agent trajectories, inferred edges serve as proxies for execution impact. We audit this assumption through *fixed-call replay*, swapping two recorded tool calls while preserving their identities and arguments and measuring changes in observations, final state, and reward. Across more than $40{,}000$ transpositions on τ-bench and BFCL, we compare token overlap with typed exact value matching between outputs and arguments and with a published typed-ID extractor. On the same eligible pairs, exploratory analyses show that typed matching improves prediction of observation changes relative to token overlap, with Matthews correlation coefficients of 0.13–0.29. The published extractor reaches 0.16–0.37. A predictor based on whether the originally later call changes state during baseline or prefix execution reaches 0.26–0.75, exceeding typed matching and the published extractor in every evaluated setting. This predictor requires environment replay and additional per-call state access, but no pairwise transposition sweep. The advantage extends to trajectory-level observation-change scoring, although gains over simple inversion counts remain uncertain in two of six arms when response positions are fixed. At a fixed inspection budget, it increases observation-change yield but lowers state-change yield relative to typed matching. Fixed-call replay measures execution with supplied arguments; it does not establish whether the agent could generate the reordered calls or identify direct causal dependencies. These findings motivate outcome-specific validation of dependency-based trajectory scores.

1 INTRODUCTION

A relation that describes how a tool call obtains its arguments need not predict the execution consequences of reordering recorded calls. Consider a user lookup that supplies an ID to an order lookup. The first call can be necessary to acquire the ID, yet replaying the second with that recorded ID already supplied can preserve its result when the calls are reordered. Conversely, knowing a reservation ID does not make the reservation exist before its creation. These examples distinguish information acquisition from execution with supplied arguments. Neither distinction invalidates a relation designed to represent value reuse.

The distinction matters when dependency violations are used as a proxy for execution impact. We ask how evaluated value-reuse extractors predict fixed-call observation changes, whether a predictor with state access adds information, and which inspection or evaluation decisions this supports. We compare token overlap and typed scalar equality with a frozen published typed-ID implementation and a schema-aware ID matcher. Write status records whether a call changes state during baseline or prefix execution. It uses additional execution information, without requiring a pairwise transposition sweep.

Our fixed-call replay procedure starts from the same initial configuration, retains recorded call identities and serialised arguments, and re-executes after changing their order. We distinguish observation changes (MECH), final state changes (STATE), and benchmark score changes (REWARD). The current pair comparison uses $41{,}473$ pairs from six τ-bench (Yao et al., 2024) and two BFCL arms (Patil et al., 2025) after excluding known input artefacts; historical estimates and adjudications retain their original populations. Fixed-call replay does not test whether an agent could generate the

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.

calls in the new order. A non-adjacent transposition also moves the endpoints across intermediate calls, so its outcome does not identify a direct causal edge.

Extractor choice explains part of the observation-prediction gap. On the same τ-bench pairs, published typed-ID improves MCC over typed scalar equality by 0.046–0.149, with positive adjusted paired intervals in $4/6$ arms. Write retains a 0.246–0.419 advantage over published typed-ID, with positive adjusted intervals in $6/6$ arms. At the trajectory level, response-fixed write scores improve observation-count ranking over total tool inversions in $4/6$ arms. At a fixed inspection budget, write increases observation-change yield over the typed-value baseline by 4.37 percentage points, while state-change yield is 0.98 points lower. These latter comparisons do not include the published typed-ID score. The separate evaluator-selection study does not establish a change in evaluator choice.

We contribute an outcome-specific validation procedure and an audit of its practical scope. The binding controls make the supplied-information intervention explicit; the matched-pair comparisons quantify the gains from refined extraction and additional state access; and the inspection analysis shows why the target outcome matters when choosing a score. The supported use is prioritising observation-change inspection under the reported conditions. Functional evaluation and information-flow provenance require their own validation.

2 Related work

2.1 Dependency structure over tool calls

Our scoped survey covers relations used in tool-call plans, workflows and trajectory evaluation. Of 34 works, 19 use only value/data provenance within the coded relation scope, 12 also use other relations, and three have no qualifying value relation. The 31 value-inclusive works use different extraction methods; they do not all implement one matching rule. We found no study evaluating an edge's prediction of replay changes after transposing two calls with identities and concrete arguments fixed. This arXiv-based, single-coder sample does not establish field-wide prevalence. Appendix H defines the coding scope, per-work evidence and sensitivity to counting graph topology alone.

Sun & Kazakov (2026) study representations of an oracle value-reuse DAG. Their typed-ID edges are a subset of their substring edges (precision 1.0, recall 0.754, treating substring edges as the reference); TaskBench supplies independent DAGs for a separate representation-probe replication. Our token-overlap and scalar-equality predictors differ from these oracles. The additional comparison applies the frozen public substring and typed-ID functions to our logs, while separately retaining the paper-literal substring definition (Section 3; Appendix J.2). It does not reproduce the representation-probing study.

GRADE (Zhao, 2026), outside the frozen survey frame, uses corpus-specific resource-access and identifier-consumption rules and evaluates failure prediction against run-size and degree-matched attachment controls. Our replay comparison evaluates fixed-call pair-transposition outcomes.

2.2 Controlled perturbation of agent trajectories

Trajectory perturbation studies include OTAP's graph-alignment evaluation of causal inversions and mixed corruptions (Barazandeh et al., 2026), Plan-RewardBench's text edits (Wang et al., 2026b), and Zhang et al. (2026)'s ten operational fault types over 275 traces. We transpose two recorded calls and measure the resulting execution changes by re-execution. Our question is which pairs predict execution changes and how edge-based severity scores rank reorderings; detecting disorder in a trajectory is a separate target (Appendix H). We apply published extractors and evaluation formats to fixed-call replay without reproducing the source studies' full tasks or results; Appendix Table 18 details the scope of each comparison.

2.3 Process quality, and the validation of instruments

Process failures can escape a benchmark's reward (Liu et al., 2026; Cao et al., 2026; Wang et al., 2026a; Sahoo et al., 2026). False Success (Advani, 2026) studies a different failure: an agent claims completion despite a failed programmatic evaluation. Our concern is how call order is measured.

Hu et al. (2026) formally define redundancy by success preservation after deletion, but construct labels through heuristic candidates and expert review. This formal criterion permits observation or state changes despite a preserved score; it does not imply that every label came from a deletion-and-reexecution test. The outcome distinction in Appendix C also matters for our transposition intervention.

Among studies of instruments rather than agents, AgentRewardBench (Lù et al., 2025) evaluates multiple judges and rule-based evaluators across web benchmarks against expert trajectory annotations. Fan et al. (2026b)'s step-level human annotations on BFCL and τ^2-bench provide the closest comparison to Section 4.3, with broader scope. At the benchmark level, Zhu et al. (2025b) audit τ-bench's construct validity and report that a do-nothing agent passes 38% of its airline tasks; we use that benchmark and inherit the caveat (Section 6).

3 A FIXED-CALL REPLAY TEST OF ORDER SENSITIVITY

We compare dependency proxies with execution changes under fixed-call replay, using recorded call pairs for the pair audit and complete reorderings for observation-change ranking and inspection.

3.1 EXPERIMENTAL UNITS AND REPLAY CONDITIONS

Let $a_1, \ldots, a_n$ be the recorded action sequence, with tool-call indices $T \subseteq \{1, \ldots, n\}$. Pair endpoints are tool calls, but observation comparison follows the owning runner over all recorded action identities. In τ-bench this includes `RESPOND` pseudo-actions (EL-1 in Appendix A). Write o_k for the observation returned for action a_k. The baseline and reordered runs start from the same initial environment configuration, and the target fixes call identities and serialised arguments at their recorded values (known argument-aliasing and BFCL string-rewriting exceptions are excluded from the current comparisons and documented in Appendix J.1). We exchange the positions of two selected calls and leave all other calls in place. Since the agent regenerates neither calls nor arguments, the intervention tests execution with supplied arguments. Whether the agent could have generated those arguments in the new order is a separate question that this intervention leaves open. As a controlled illustration, we execute 100 explicit-binding workflows using unchanged τ-bench tools. Reordering an information-only workflow fails in $20/20$ instances when the consumer must obtain the producer's return, but in $0/20$ when the recorded value is supplied. State prerequisites can cause failure under both policies. These constructed controls illustrate the intervention's scope; they do not estimate its prevalence in natural logs or the performance of adaptive agents (Appendix J.5).

A trajectory enters only after satisfying its corpus-specific admission and fidelity gate (Appendix G). Reproducing the recorded τ-bench reward, for example, does not require byte-identical observations in every corpus. Section 5.1 gives the environments, admitted populations and analysis-specific exclusions.

3.2 INTERVENTION AND OUTCOMES

For (i, j) with $i < j$ and $i, j \in T$, we transpose a_i and a_j, re-execute the whole trajectory, and compare each call's observation with its own baseline. Observations are matched by original action index, so a_k is compared with itself wherever it moves. Writing o'_k for action a_k's observation under transposition, we define

$$\mathrm{MECH}(i, j) \;=\; \mathbf{1}\big[\,\exists k \in \{1, \ldots, n\} : \; \mathcal{C}_{\mathrm{env}}(o'_k) \neq \mathcal{C}_{\mathrm{env}}(o_k)\,\big], \tag{1}$$

Here $\mathcal{C}_{\mathrm{env}}$ applies the original runner's equality rule: normalised observations in τ-bench and exact `str()` equality in BFCL. Appendix G specifies the comparison rules.

We record three distinct outcomes:

- **MECH**: any changed identity-matched observation.
- **STATE**: a changed final environment data hash.
- **REWARD**: a changed benchmark score.

After a_1, $x = 0$; a_2 leaves it unchanged. a_3 reads x and a_4 sets x to 1.

Baseline	**Transpose** $(2, 4)$
$a_1,\ a_2,\ a_3,\ a_4$	$a_1,\ a_4,\ a_3,\ a_2$
$o_3 = 0$	$o_3' = 1$

Figure 1: A conceptual shared-state effect under fixed-call replay. Moving a_4 ahead of a_3 changes the intermediate read, so $\mathrm{MECH}(2, 4) = 1$. This is a possible path to an observation change, not a sampled trajectory, a general later-write advantage, or a direct dependency between a_2 and a_4. The separate binding controls in Appendix J.5 distinguish supplied-information and state-prerequisite constraints.

These answer different questions about observations, state, and task outcome; their empirical relationship is reported in Appendix C.[1] For a non-adjacent pair, transposition also changes each endpoint's order relative to every intervening call, so MECH measures the outcome of this in-context intervention; a positive label does not identify a direct dependency between the endpoints. Figure 1 illustrates one possible shared-state path.

3.3 Predictors and information access

The current comparison distinguishes five predictors:

- **Token overlap (`rule`)**: a shared extracted token in a source observation and later argument text; field names can contribute tokens.
- **Typed scalar equality (`typed_all`)**: matching scalar representation type and canonical value, without schema-ID semantics.
- **Published typed-ID (`published_id`)**: the frozen public extractor accompanying Sun & Kazakov (2026), using its ID candidates and normalisation. Its public substring comparator is reported separately.
- **Schema-ID (`schema_id`)**: our fixed mapping of output and argument paths to entity-ID types, compared with the same candidates ignoring entity type. This matcher is defined for τ-bench only.
- **Write status (`a_write`/`b_write`)**: the originally earlier/later call changes the data hash during baseline or prefix execution.

The value matchers do not establish actual provenance. Published and new schema-ID matchers differ in candidate and normalisation rules, so their difference does not isolate semantic typing. Appendix J.2 specifies the frozen public code, schema rules, ablation and extraction fidelity; Appendix G preserves the original token/scalar definitions.

We compare extractors on the same available pairs. Value matching uses observation–argument logs; schema-ID also needs the fixed semantic mapping. Write additionally requires a re-runnable environment and per-call state access. In τ-bench it is collected during one baseline pass; BFCL uses fresh prefix replays with a quadratic number of primitive tool executions. It needs no pairwise transposition sweep, unlike MECH labels and the replay-informed reference. These access requirements and execution counts are distinct from wall-time or cost equivalence (Appendix G).

3.4 Metrics and analysis status

The pair audit asks whether a predictor identifies execution changes within an arm. We report lift alongside coverage and balanced metrics defined below. The pre-registered primary comparison is the paired lift difference between `b_write` and `rule` for the τ-bench and BFCL held-out audits; BFCL base is exploratory. MECH and STATE are evaluated separately wherever the protocol specifies both.

[1] Li et al. (2026) show that losses and compensations can cancel in a matched score and argue for reporting execution paths and final-state validators. Gao & Zhou (2026) bound scores when a surface check can pass without the intended state change. They study score and check validity; we ask what a reported insensitivity conceals across levels.

Balanced and extractor comparisons were planned with earlier outcomes available and are exploratory; they do not re-adjudicate the original gates. We use paired task-bootstrap inference, with Holm–Bonferroni adjustment for the original six-arm τ-bench family and Bonferroni intervals for the 24 new τ-bench MECH contrasts. BFCL and STATE in the newer extractor analysis are supplementary. Resampling, stability and support rules are in Appendices F and J.2.

We report $\mathrm{lift}(P) = \mathrm{precision}(P)/\mathrm{base}$ as positive-set enrichment, where $\mathrm{base} = \Pr[\mathrm{MECH} = 1]$ within an arm. Because coverage affects attainable lift, we interpret it alongside positive prediction rate q, recall, false-positive rate, informedness ($J = \mathrm{recall} - \mathrm{FPR}$), and MCC (Appendix Table 5). The pre-registered eligibility band excludes arms with base outside $[0.05, 0.60]$ from lift claims; Appendix B gives the bounds and their interpretation.

3.5 Trajectory-level ranking and inspection

For each eligible τ-bench trajectory with at least three tool calls, the response-fixed generator produces $K = 12$ fixed-seed draws by moving tool identities while retaining every `RESPOND` slot. Scores count violated token edges ($V_{\texttt{rule}}$), typed-value edges ($V_{\texttt{typed}}$), later-write inversions ($V_{\texttt{write}}$), or all tool inversions. $V_{\texttt{mech}}$ is a replay-informed reference without a guaranteed performance ceiling. BREAK counts identity-matched observations changed by the variant's actual execution.

For a permutation π, the score is $V_P(\pi) = \sum_{i<j,\, i,j \in T} P(i,j)\mathbf{1}[\mathrm{pos}_\pi(i) > \mathrm{pos}_\pi(j)]$, where P is the selected pair predictor or replay-informed MECH label.

Within-trajectory concordance ranks variants by BREAK count and is macro-averaged over trajectories with varying outcomes. Original draw weights are primary; unique-permutation weighting is a sensitivity. Appendix I.4 gives tie rules, exploratory paired task-bootstrap intervals and the distinct historical pooled AUROC analysis.

For inspection, we select the top $\lceil qn \rceil$ non-identity unique response-fixed variants per trajectory. Yield is the positive fraction among selected variants, not recall. Observation change, final-state change and success loss are separate endpoints; the last uses successful baselines only. Appendix J.3 gives task averaging, ties, uncertainty, budgets and support rules. Neither trajectory analysis includes a published typed-ID score or validates a complete external graph-alignment metric.

4 Checks on the replay measure

We examine replay stability and three further properties of the fixed-call measure: whether it misses known positive causal links in classical planning, how its labels compare with human judgments, and whether the principal positional confound accounts for the later-write association. The checks have different targets, and together they do not establish a universal dependency relation.

4.1 Same-order replay control

As an exploratory control, we repeated same-order runs of all 1,880 pair-contributing trajectories in the original primary populations, together with the two previously excluded BFCL-base trajectories. Under the original comparison rules, observations change only in the five BFCL artefact cases already known, and two of these also change final state. No write or available value-matching predictor label changes between repeats or disagrees with its stored baseline label, and no null change is observed in the tag-clean populations. Because the repeats are finite, the check cannot establish universal determinism, nor does it revalidate every transposed outcome (Appendix F gives the design and scope).

4.2 A one-sided check on known causal links

Across 113 planner-generated plans in 12 PDDL domains (Alkhazraji et al., 2020), the replay test misses none of the computed causal support links in the observed sample: $\Pr(\mathrm{MECH} = 1 \mid \text{computed causal link} = 1) = 1.000$ in every domain. This one-sided check establishes neither specificity nor a universal guarantee, direct-edge recovery, or validity on LLM trajectories (Appendix B).

4.3 HUMAN JUDGMENTS BEFORE AND AFTER OBSERVING EXECUTION

We compare human judgments of functional change from calls alone (J1) and after both observation sequences are revealed (J2). Author-rater A assessed 200 pairs; independent raters B and supplementary C assessed the same 100 stage-1 items. A's pre-registered point-estimate verdict remains PASS, although two of three intervals cross their pass thresholds. Calls-only agreement depends on the rater: C's agreement and positive judgments on MECH-negative items fall in the original failure bands on point estimates. These supplementary results do not re-adjudicate A's verdict. J2 reveals the observations that define MECH, so its agreement is not independent semantic validation; six MECH-positive items receive J2 No from all three raters. Observation change and functional endorsement remain distinct (Appendix D).

4.4 POSITION-CONFOUND CONTROL

In the original τ-bench populations, logistic regression adjusting for call position and distance gives positive 95% intervals for the later-write coefficient in all six arms; a model-free stratified comparison also gives positive risk differences throughout. The association thus survives the measured position controls, without establishing a causal effect or a general later-call mechanism: the directional hypothesis fails on BFCL. Design, reduced-model and argument-exclusion sensitivities are reported in Appendix I.2; the BFCL result is in Appendix B.

5 THE AUDIT

5.1 CORPORA AND ANALYSIS POPULATIONS

We executed $41{,}626$ transpositions across six τ-bench arms and BFCL multi-turn base and held-out. After excluding known harness artefacts, the current exploratory pair population comprises $41{,}473$ pairs; exclusion details and historical analysis populations are reported in Appendix J.1.

The τ-bench audit was pre-registered; BFCL base was exploratory, and held-out tasks were disjoint from base under a protocol committed before generation. SWE-bench (Jimenez et al., 2024) and planner-generated PDDL are supporting analyses outside these totals. Appendix B gives the original admission and arm configurations. Supporting position, effect-location, human and outcome-level analyses retain their stated historical populations.

5.2 PUBLISHED EXTRACTORS AND THE REMAINING PREDICTION GAP

Figure 2 compares predictors on the same available pairs after the known-artefact exclusions. The τ-bench common support is all $33{,}059$ pairs; BFCL common support is $2{,}373/2{,}484$ base and $5{,}692/5{,}930$ held-out pairs. Missing typed observations are unavailable, not negative labels. The new extractor analysis is exploratory: its plan and implementation were frozen after earlier outcomes were known.

The published typed-ID extractor improves τ-bench MECH MCC from the typed-value baseline's 0.188–0.292 to 0.281–0.370. Paired gains are 0.046–0.149, positive in point estimate for all six arms. Their intervals exclude zero in $5/6$ arms at marginal 95% and $4/6$ after adjustment for the 24 planned comparisons. Write status reaches MCC 0.571–0.750; its advantage over published typed-ID is 0.246–0.419, with positive adjusted intervals in all six arms (Appendix Table 35). Write uses additional per-call state information, so these are not comparisons under equal information access.

The schema-ID matcher does not improve on typed scalar equality. Its edges are identical to those of the same ID-candidate matcher with entity types ignored in all six arms. The data therefore provide no pair-level contrast for isolating the benefit of that semantic typing step. This result concerns the implemented ID-only matcher, not schema-aware provenance methods in general. Appendix J.2 reports the candidate rules, ablation, public-code provenance and all extractor results.

In BFCL, published typed-ID has MCC $0.171/0.161$ (base/held-out). Its gains over typed scalar equality have 95% intervals spanning zero, while its precision is about 0.76 and recall only $0.061/0.053$. Write also outperforms published typed-ID in both BFCL arms, with marginal 95% confidence intervals for the MCC differences entirely above zero. These supplementary compar-

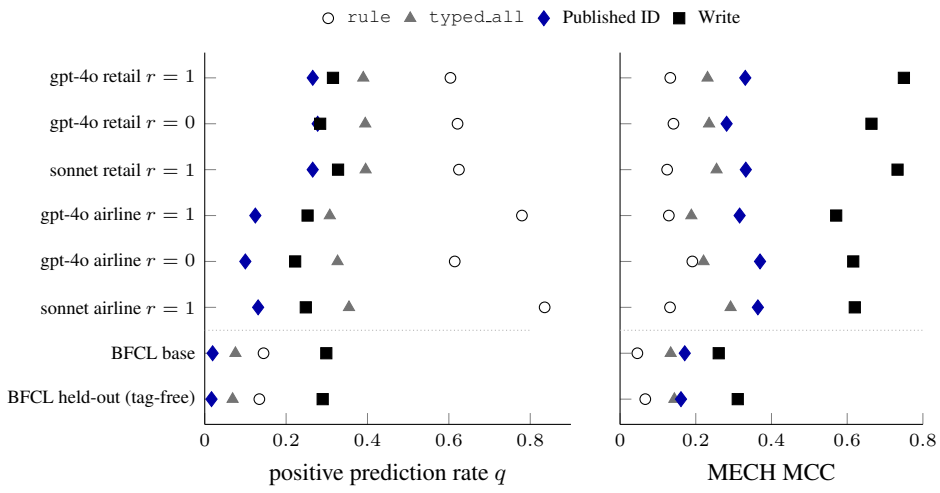


Figure 2: Coverage and MECH MCC on common available pairs after known-artefact exclusions. The published ID extractor improves on typed scalar equality in all six τ-bench point estimates, with positive adjusted intervals in four; write-minus-published-ID intervals are positive in all six. BFCL is supplementary and its ID-minus-typed intervals span zero. Write requires additional state access. Current support is 33,059 τ-bench pairs and 2,373/5,692 BFCL pairs (base/held-out). Schema-ID and the full paired intervals appear in Appendix J.2.

Table 1: Response-fixed reorderings after the argument-aliasing exclusion. I, T and W are total tool inversions, typed-value and later-write inversion scores. Eligible/all counts trajectories with varying BREAK counts. Paired 95% intervals are exploratory marginal task-bootstrap intervals, using the original draw weights; outcome ties are omitted.

Arm	Eligible/all	Concordance I/T/W	W−I (95%)	W−T (95%)
gpt-4o retail $r = 1$	211/247	.694/.828/.895	$+.201$ $[+.171, +.231]$	$+.068$ $[+.041, +.094]$
gpt-4o retail $r = 0$	134/158	.697/.795/.858	$+.161$ $[+.122, +.199]$	$+.063$ $[+.034, +.092]$
sonnet retail $r = 1$	475/560	.702/.811/.880	$+.178$ $[+.153, +.203]$	$+.069$ $[+.044, +.094]$
gpt-4o airline $r = 1$	24/47	.813/.812/.864	$+.051$ $[-.025, +.150]$	$+.053$ $[-.029, +.159]$
gpt-4o airline $r = 0$	64/76	.775/.784/.871	$+.096$ $[+.042, +.154]$	$+.086$ $[+.033, +.142]$
sonnet airline $r = 1$	48/144	.807/.817/.871	$+.063$ $[-.001, +.128]$	$+.054$ $[-.017, +.132]$

isons are outside the 24-comparison τ-bench family; exact estimates and intervals appear in Appendix J.2. Historical write–rule lift and write–typed comparisons remain in Appendix I. Neither the current MECH comparison nor the mixed STATE results establish all-outcome superiority.

An exploratory check on the original pair populations restricts positive outcomes to observation changes at a swapped endpoint or a later call. The write–rule MCC difference retains positive intervals in $5/6$ τ-bench arms and both BFCL arms, so the advantage is not confined to intermediate-only changes; this does not identify a direct causal edge (Appendix F).

5.3 Ranking observation changes under response-fixed reorderings

We keep response slots fixed and compare variants within each trajectory by the number of changed identity-matched observations, BREAK (Section 3.5). The current estimate excludes the known aliasing trajectory. Of 1,232 trajectories, 956 have varying BREAK counts and contribute to the ranking comparison; the other 276 have only zero counts.

Write exceeds total tool inversions with positive intervals in $4/6$ arms; both successful-airline arms are inconclusive. Its advantage over the typed score also has positive intervals in $4/6$ arms (Table 1). Weighting each unique permutation once yields $5/6$ positive write-minus-inversion intervals, with

Table 2: Outcome-specific inspection yield at the registered nominal 20% budget. W is write and T is typed scalar equality. Differences use paired task-bootstrap 95% intervals. Observation change is the planned primary contrast; STATE is a supplementary marginal comparison. Success loss is defined only on successful baselines and has insufficient event support for an interval or superiority verdict.

Outcome	W yield	T yield	W−T, percentage points (95%)
Observation change	63.25%	58.87%	$+4.37\ [+2.72, +6.08]$
Final-state change	4.25%	5.24%	$-0.98\ [-1.86, -0.25]$
Success loss	4.61%	5.41%	Interval and verdict withheld

sonnet airline $r = 1$ becoming positive, and $4/6$ write-minus-typed intervals. We retain this as a weighting sensitivity rather than selecting the more favourable count.

The historical generator also moved response positions. Its pre-registered write-minus-token AUROC gain has median 0.181 across six arms (Appendix Table 27); that pooled classification result predicts any observation change and is distinct from the within-trajectory count ranking above. Fixing responses also changes inversion exposure, so the comparison does not isolate a causal effect of response movement. Neither analysis measures reward-loss magnitude or includes a published typed-ID trajectory score. Appendix I.4 preserves the historical populations, simple controls and both weighting schemes.

5.4 Inspection yield and evaluator selection

We next ask which variants would be selected for replay under a fixed inspection budget. The analysis uses $12{,}416$ non-identity unique tool-only variants from $1{,}232$ trajectories and 164 tasks, excluding the aliasing trajectory. Under the selection rule in Section 3.5, the nominal 20% budget selects an actual task-macro share of 25.73%. This equalises the number of inspected variants, not predictor information access or total computation.

Write improves observation-change yield but has lower state-change yield than the typed-value baseline (Table 2). Success loss occurs in only 17 tasks, below the registered minimum of 20. This supports prioritising observation-change inspection under the evaluated rule, not a general improvement in functional failure detection. STATE is itself a change measure rather than a task-error label. The analysis was specified with earlier outcomes known and is exploratory; it does not establish a gain over the published typed-ID score. Appendix J.3 reports budgets, other scores, uncertainty and sensitivities. Supporting τ-bench and SWE-bench analyses further characterise the relationships among the three outcomes (Appendix C).

The strict-format primary analysis of a public Plan-RewardBench judge transfer (Wang et al., 2026b) was technically incomplete. A supplementary application of its original parser did not establish a change in evaluator selection (Appendix J.4).

6 Limitations

Scope of the measured relation. Fixed-call replay tests execution with supplied arguments, leaving argument generation and direct-edge recovery unresolved (Section 3). The effect-location analysis does not identify causal paths (Appendix F). Typed equality likewise does not verify provenance: matching values may already exist in the prompt or initial state, and BFCL scalar root types are unavailable after harness serialisation. The findings apply to the evaluated extractors and measured outcomes. The published typed-ID implementation originated in retail; its airline and BFCL uses are domain transfers. The schema-ID ablation yields identical edges with and without entity typing, leaving that component unidentifiable on these pairs.

Partial independent human validation. Calls-only judgments depend on the rater. C's less favourable results are a supplementary comparison; the original adjudication uses author-rater A (Section 4.3). A's C3 point estimate passes narrowly: one additional Yes would move the verdict back to CONCERN. Independent ratings cover only stage 1; stage 2 is rated by A alone, with no

further independent ratings planned. Higher J2 agreement reflects judgments made with observations revealed and cannot independently validate the measure. The overlapping samples and their observed zero miss counts do not establish a zero population miss rate (Appendix D).

Information access and severity conditions. Write status needs a re-runnable, state-inspectable environment, so its advantage is conditional on access beyond observation–argument logs (Section 3). Severity comparisons also depend on the intervention and weighting (Section 3.5); fixing responses changes perturbation exposure as well as which actions can move. Within-trajectory ranking is conditional on varying BREAK counts. Its gains measure predictive ranking, not reward loss or a causal contribution of graph information at fixed perturbation magnitude (Section 5). The inspection analysis improves observation-change yield but not state-change yield over typed scalar equality, and success-loss support is insufficient. It does not compare against a published typed-ID trajectory score. The judge transfer is technically incomplete under its primary parser and does not establish a change in evaluator choice under its supplementary parser. Explicit-binding workflows are constructed mechanism controls, not estimates of natural-agent failure prevalence.

Environment and population coverage. The later-write association survives position control in τ-bench, but the pre-registered directional hypothesis fails on BFCL (Section 4.4); the write–rule advantage therefore does not establish a universal later-call mechanism. The pre-registered BFCL correctness test indicated an effect on MECH rather than equivalence. The reported estimator is a gold-anchored fallback; a post-hoc decomposition attributes about half the solved–failed contrast to task composition, while the intended within-task comparison lacked pre-specified cluster support (Appendix B). Success concentrates at later BFCL task indices, where sensitivity is lowest. We use Appendix C for containment and endpoint screening, without interpreting between-environment variation as an axis of state mechanism.

Lift claims are limited to τ-bench and BFCL: the eligibility band excludes PDDL and SWE-bench's high base rates (Appendix B). No measured agent-generated point fills the observation-level range between 29.4% and 95.2%; BFCL's 57.3% point uses gold sequences, and the larger generator was not run because its termination regime would differ (Appendix B). We also inherit τ-bench's uneven shares of intentionally unsolvable tasks (38% airline, 6% retail; Zhu et al., 2025b), which confound domain comparisons. Arms are reported separately without a domain-mechanism claim. The original BFCL held-out adjudication retains its frozen population; the current extractor comparison uses the tag-free subset (Section 5.1). Supporting analyses retain their stated populations.

Input integrity and historical estimates. In one airline trajectory, cancellation mutates retained payment arguments. BFCL's instance-name helper also changes string arguments in five known cases, two with state effects. The current extractor and inspection estimates exclude the affected trajectories, and the current response-fixed table uses the airline exclusion estimate. Historical results remain separately identified. The isolated-call probe starts from admitted inputs; we have not relabelled the full cohort from raw inputs with a corrected harness. Same-order repeatability and matching start/end hashes do not guarantee immutable arguments throughout execution (Appendix J.1).

7 Conclusion

Relations that describe value reuse need outcome-specific validation when used to score execution impact. On supplied-argument replays, a frozen published typed-ID extractor improves on our typed-value baseline, while a gap to a predictor with additional state access remains in the six τ-bench arms. Response-fixed ranking gains depend on the arm and weighting. At the evaluated inspection budget, write selects more variants with observation changes but fewer with final-state changes than typed scalar equality; success-loss evidence is insufficient for a superiority verdict.

The supported application is observation-change inspection under the reported information and replay conditions. The evaluator transfer does not establish a change in evaluator choice, and the binding controls illustrate why supplied arguments bypass information acquisition without removing state prerequisites. Reporting the target outcome, population and information access makes these distinctions explicit. Fixed-call sensitivity does not recover information provenance, direct causal edges, or general functional quality.

ETHICS STATEMENT

This work audits a measurement construct rather than the researchers who use it. We classify the thirty-four surveyed works from their own definitional sentences; the survey materials record each source passage and its location, and Appendix H summarises the definitions. Several of these works also delimit their construct appropriately in their limitations. Our claim concerns the gap between what the construct establishes and what it is used to establish, and we make no judgement about whether any individual work overclaims.

The human comparison collected judgments from three adults who consented to participate in the research and to the use of their responses. Rater A, an author, assessed 200 items, and B and the subsequently added C each assessed the same stage-1 subset of 100. We collected no demographic information, and names or initials in response exports are omitted from reported results. The items were tool-call trajectories taken from public benchmark releases, and no new agent behaviour was elicited from human subjects.

All trajectories are either released with their benchmarks (τ-bench, SWE-bench), generated by us from public model weights on public tasks (BFCL), or produced by a classical planner (PDDL). The environments are sandboxed simulations, and no live service, account, or third-party system was contacted, with one exception: a single SWE-bench instance reaches the public internet during its own test suite (Appendix E).

REPRODUCIBILITY STATEMENT

Experimental estimates are traced to stored analysis outputs and, where these are complete, to their pair records; survey counts come from the coded literature records. We froze each protocol before its confirmatory run and disclose both prior pilot results and subsequent deviations; the PDDL archive limitation is stated below. Appendix A indexes each pre-registration against the result file it governs and summarises the deviations that affect the manuscript. The fixed-call replay definition, including identity-matched keying, is set in those protocols and not in this paper.

The environments use corpus-specific admission and fidelity gates, summarised in Appendix Table 15. Briefly, τ-bench admits unmodified replays that reproduce recorded reward; BFCL generated trajectories must execute under the pinned harness with at least three model-emitted calls, and the held-out all-pairs analysis additionally caps trajectories at twelve calls; SWE-bench replays in pinned `:v1` images, with $28/30$ instances reproducing their published submission byte-for-byte. The frozen BFCL held-out primary population is not retroactively filtered for the tag artefact. Version pins matter in one case. BFCL's v3 and v4 multi-turn data differ in $107/200$ questions and $67/200$ ground truths, and because the same code run against v3 returns a plausible wrong number instead of failing, the `bfcl_eval` release is pinned. Generation is greedy, with one trajectory per task. The typed exact-equality and effect-location/adjacent analyses use the same stored pair populations and outcomes. Their plans were committed before implementation, and their complete pair-level outputs are released. They are labelled exploratory and do not re-adjudicate the confirmatory gates. Inference for the original pair audits uses a bootstrap clustered on task with $N_{\mathrm{BOOT}} = 20{,}000$, and confirmatory verdicts are re-checked across 100 seeds with the flip rate reported. For C's intervals, the supplementary C analysis reuses the frozen human-comparison estimators, resampling the four stage-1 environment-by-MECH cells $20{,}000$ times at seed 0. It reports no new adjudication and no 100-seed stability check. Its separate script and result file retain item-level labels under rater codes and input hashes (Appendix D).

The subsequent review supplements are also exploratory. They comprise a same-order replay census, direct intervals for the write-minus-typed difference, a reduced position design, and severity controls. We wrote their plans after the historical outcomes were known, and freezing those plans before the new computations does not turn the supplements into retrospective confirmatory evidence. Because the original writer did not save the severity variants, the new archive reconstructs the fixed-seed draws by re-execution. It records their permutations, changed observations, state/reward outcomes, predictor scores and simple controls, together with a separate tool-only intervention. We distinguish agreement with the historical aggregate from recovery of a previously preserved raw file. The new results come with input/code hashes, paired draw arrays, undefined-estimator counts and runtime versions, but these supplements have no new 100-seed verdict check.

The further published/schema extractor comparison uses $50{,}000$ paired task-bootstrap draws and a 24-contrast MECH family. Current extractor, response-fixed ranking and inspection estimates exclude known input artefacts; the frozen results and adjudications remain separately reported. The inspection plan was specified with earlier outcomes known. The public-judge transfer retains its technically incomplete strict-parser primary result and its separately declared original-parser diagnostic. Explicit-binding controls use constructed workflows with fixed argument provenance. Their 400 executions illustrate the supplied-information mechanism rather than natural task failure rates. Input, implementation, selection and verification records for these analyses are indexed in Appendix J. None is a full corrected-harness replay of the original pair cohort.

Casheekar (2026) identifies a requirement that replay-based measurement must satisfy: a scored run needs both outcome finality and separation from earlier runs. They exhibit a delayed write that changes the next run's score under shared state. Here, every transposition re-executes the whole trajectory from a fixed initial configuration instead of continuing from a previous run's state, and each corpus checks baseline fidelity at the level its stated gate defines. Environments with deferred or external effects would require additional safeguards beyond those used here.

Because PDDL admission uses wall-clock timeouts, corpus composition varies, and the per-domain counts in Appendix B describe the stored aggregate run rather than a machine-independent sample. The conditional MECH rate on computed links was 1.000 in every domain across four generated corpora; this is an empirical observation. Aggregate outputs preserve the reported $27{,}634$-pair run, but its pair export contains only the first $20{,}000$ rows because the writer truncates that file. The archive therefore does not support full pair-level reconstruction of that historical table. The aggregate ceiling calculations are descriptive, and no PDDL lift hypothesis is adjudicated.

AI USE STATEMENT

In this work, we used generative AI tools, including large language models and LLM-based coding assistants, for conceptual development, hypothesis refinement, experimental design, method implementation, data preparation, result interpretation, synthetic trajectory generation, mathematical reasoning, and translation. We did not use these tools to supply human participants' ratings or to assign execution-based replay outcome labels. We also used generative AI for literature discovery and comparison, code development and debugging, manuscript drafting and editing, and preparation of tables and figures. These contributions were substantial and extended across the research workflow. The human authors reviewed the AI-assisted work; the five areas below describe the assistance, followed by our verification and responsibility statement.

1. **Writing assistance and editing.** We used AI tools to improve the clarity, grammar, organisation, and readability of existing manuscript text. This included shortening repetitive passages, improving paragraph transitions, and making descriptions of value reuse, fixed-call replay, and the observation, state, and reward outcomes more consistent. AI also assisted with English–Korean translation for the accompanying working documents, consistency checks on terminology and notation, and editing table captions and figure explanations.
2. **Literature retrieval and discovery.** We used AI tools to identify potentially relevant work on tool-use agents, dependency extraction, trajectory evaluation, and replay-based measurement. They helped formulate searches, summarise candidate papers, locate passages defining dependency, and organise comparisons of the surveyed definitions and implementations. This assistance informed the literature survey and related-work discussion. The authors checked the source passages in the original papers and decided which references, classifications, and characterisations to retain.
3. **Research ideation and execution.** AI tools assisted with developing the distinction between value reuse and fixed-call replay order sensitivity, refining hypotheses, and drafting experimental protocols, preregistrations, and analysis plans. They helped implement and debug trajectory loading and preprocessing, replay harnesses, dependency predictors, control experiments, and statistical analysis scripts. They also assisted with running experiments, inspecting execution records, preparing result tables and plots, and examining alternative interpretations, confounds, and limitations. This included the predictor comparisons, position and severity controls, and exploratory follow-up analyses described in the paper. AI assistance with mathematical reasoning included formulating and checking

the lift bounds and reward–state accounting retained in the appendices, together with their assumptions and derivations.

4. **Drafting manuscript sections.** We used AI tools to produce initial drafts and proposed revisions of the abstract, introduction, related work, methods, results, limitations, and appendix explanations. These drafts drew on the research plans, source papers, code, and stored experimental outputs. AI helped turn these materials into descriptions of the replay procedure, predictor comparisons, controls, and observed results, and proposed ways to organise the argument and distinguish confirmatory findings from exploratory analyses. The authors revised the drafts and selected the final wording and scientific claims.
5. **Synthetic data generation.** We used open-weight Qwen2.5 models at 7B, 14B, and 32B scales to generate the BFCL agent trajectories analysed in this study. The models produced tool calls and arguments for public BFCL tasks using the generation harness, greedy decoding, and admission criteria described in the paper. These trajectories constitute synthetic experimental data. The audited τ-bench and SWE-bench trajectories were obtained from existing benchmark releases, and PDDL plans were produced by a classical planner. Pair-level replay outcomes were computed from execution comparisons, not assigned by an LLM judge. The separate exploratory judge-transfer experiment used Llama 3.1 8B and Qwen2.5 7B to generate evaluator responses; those responses were evaluated against, and did not define, the replay outcomes.

The human authors made the final decisions about the research questions, experimental designs, preregistrations, methodological changes, analyses, interpretation of findings, and submission content. They reviewed and double-checked all experimental results, manuscript text, and associated research artifacts produced with AI assistance. Verification included checking citations and literature classifications against original sources, reviewing code and analysis outputs, checking the assumptions and steps of mathematical arguments, and matching reported claims, tables, and figures to the stored experimental records. The authors take full responsibility for the accuracy, integrity, and originality of the final work, including its AI-assisted text, code, data, claims, and other artifacts.

REFERENCES

Laksh Advani. From confident closing to silent failure: Characterizing false success in LLM agents. *arXiv preprint arXiv:2606.09863*, 2026. v1, 2026-06-01.

Yusra Alkhazraji, Matthias Frorath, Markus Grützner, Malte Helmert, Thomas Liebetraut, Robert Mattmüller, Manuela Ortlieb, Jendrik Seipp, Tobias Springenberg, Philip Stahl, and Jan Wülfing. Pyperplan, 2020. Zenodo.

Hengyu An, Jinghuai Zhang, Tianyu Du, Chunyi Zhou, Qingming Li, Tao Lin, and Shouling Ji. IPIGuard: A novel tool dependency graph-based defense against indirect prompt injection in LLM agents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. v1, 2025-08-21; arXiv:2508.15310.

Babak Barazandeh, Subhabrata Majumdar, and George Michailidis. OTAP: Structure-aware optimal transport for evaluating planning and execution in agent trajectories. *arXiv preprint arXiv:2607.17082*, 2026. v2, 2026-08-01.

Kinjal Basu, Ibrahim Abdelaziz, Kiran Kate, Mayank Agarwal, Maxwell Crouse, Yara Rizk, Kelsey Bradford, Asim Munawar, Sadhana Kumaravel, Saurabh Goyal, et al. NESTFUL: A benchmark for evaluating LLMs on nested sequences of API calls. *arXiv preprint arXiv:2409.03797*, 2024. v3, 2025-05-21.

Kushal Raj Bhandari, Ling Yue, Ching-Yun Ko, Dhaval Patel, Shaowu Pan, Pin-Yu Chen, and Jianxi Gao. Evoflux: Inference-time evolution of executable tool workflows for compact agents. *arXiv preprint arXiv:2606.12674*, 2026. v1, 2026-06-10.

Hongliu Cao, Ilias Driouich, and Eoin Thomas. Beyond task completion: Revealing corrupt success in LLM agents through procedure-aware evaluation. *arXiv preprint arXiv:2603.03116*, 2026. v1, 2026-03-03.

Avyay M. Casheekar. When is an agent evaluation over? outcome finality and cross-unit separation. *arXiv preprint arXiv:2608.14940*, 2026. v1, 2026-08-14.

Yaxin Du, Yifan Zhou, Yujie Ge, Jiajun Wang, Xianghe Pang, Shuo Tang, Tuney Zheng, Bryan Dai, Jian Yang, and Siheng Chen. HyperTool: Beyond step-wise tool calls for tool-augmented agents. *arXiv preprint arXiv:2606.13663*, 2026. v1, 2026-06-11.

Linfeng Fan, Ziwei Li, Yuan Tian, Yichen Wang, Rongsheng Li, and Xiong Wang. The granularity mismatch in agent security: Argument-level provenance solves enforcement and isolates the LLM reasoning bottleneck. *arXiv preprint arXiv:2605.11039*, 2026a. v1, 2026-05-11.

Shengda Fan, Xuyan Ye, Yupeng Huo, Zhi-Yuan Chen, Yiju Guo, Shenzhi Yang, Wenkai Yang, Shuqi Ye, Jingwen Chen, Haotian Chen, Xin Cong, and Yankai Lin. AgentProcessBench: Diagnosing step-level process quality in tool-using agents. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD)*, 2026b. arXiv:2603.14465.

Guangyu Feng, Huanzhi Mao, Prabal Dutta, and Joseph E. Gonzalez. Concurrency without model changes: Future-based asynchronous function calling for LLMs. *arXiv preprint arXiv:2605.15077*, 2026. v1, 2026-05-14.

Shanshan Gao and Liyi Zhou. Can agent benchmarks support their scores? evidence-supported bounds for interactive-agent evaluation. *arXiv preprint arXiv:2605.10448*, 2026. v1, 2026-05-11.

Han Han, Tong Zhu, Xiang Zhang, Mengsong Wu, Hao Xiong, and Wenliang Chen. NesTools: A dataset for evaluating nested tool learning abilities of large language models. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2025. v2, 2025-01-07; arXiv:2410.11805.

Shuo Hao, You Lu, Bihuan Chen, and Xin Peng. FlowScout: From execution feedback to reliable tool-using agent workflows. *arXiv preprint arXiv:2608.10039*, 2026. v1, 2026-08-10.

Pengfei He, Zhenwei Dai, Bing He, Hui Liu, Xianfeng Tang, Hanqing Lu, Juanhui Li, Jiayuan Ding, Subhabrata Mukherjee, Suhang Wang, et al. TRAJECT-Bench: A trajectory-aware benchmark for evaluating agentic tool use. *arXiv preprint arXiv:2510.04550*, 2025. v2, 2025-10-11.

Minyang Hu, Bo Yang, Zhinuo Zhou, Jiachen Liang, Jiahao Guo, Yiyang Yin, and Xiongwei Han. Redundant or necessary? a benchmark for detecting redundant steps in agent trajectories. *arXiv preprint arXiv:2605.29893*, 2026. v1, 2026-05-28. Introduces RedundancyBench, built on τ^2-bench.

Kyochul Jang, Donghyeon Lee, Kyusik Kim, Dongseok Heo, Taewhoo Lee, Woojeong Kim, and Bongwon Suh. DICE-BENCH: Evaluating the tool-use capabilities of large language models in multi-round, multi-party dialogues. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. v2, 2025-07-02; arXiv:2506.22853.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Dinesh Khandelwal, Gnana Prakash Punnavajhala, GPS Bhargav, Gaurav Pandey, Sachin Joshi, Hima Karanam, and Dinesh Raghu. ToolWeave: Structured synthesis of complex multi-turn tool-calling dialogues. *arXiv preprint arXiv:2605.12521*, 2026. v1, 2026-04-03.

Anay Kulkarni, ChiaEn Lu, Dheeraj Mekala, Jayanth Srinivasa, Gaowen Liu, and Jingbo Shang. TIER: Trajectory-invariant execution rewards for multi-step tool composition. *arXiv preprint arXiv:2605.16790*, 2026. v1, 2026-05-16.

Shangao Li, Yao Zhang, Volker Tresp, and Yuanyuan Yang. QuoteBench: How matched scores can hide command-path failures. *arXiv preprint arXiv:2608.13547*, 2026. v1, 2026-08-13.

Xukun Liu, Zhiyuan Peng, Xiaoyuan Yi, Xing Xie, Lirong Xiang, Yuchen Liu, and Dongkuan Xu. ToolNet: Connecting large language models with massive tools via tool graph. *arXiv preprint arXiv:2403.00839*, 2024. v1, 2024-02-29.

Yibing Liu, Yangze Liu, Xiaolong Yin, Bin Wang, Chong Zhang, Hao Yin, and Zhongyi Han. OpenClawBench: Benchmarking process-side anomalies in real-world agent execution trajectories. *arXiv preprint arXiv:2605.29253*, 2026. v1, 2026-05-28.

Zhaoyang Liu, Zeqiang Lai, Zhangwei Gao, Erfei Cui, Ziheng Li, Xizhou Zhu, Lewei Lu, Qifeng Chen, Yu Qiao, Jifeng Dai, et al. ControlLLM: Augment language models with tools by searching on graphs. *arXiv preprint arXiv:2310.17796*, 2023. v3, 2023-12-18.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, et al. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. *arXiv preprint arXiv:2408.04682*, 2024. v2, 2025-04-16.

Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, Arkil Patel, Dongchan Shin, Alejandra Zambrano, Karolina Stańczak, Peter Shaw, Christopher J. Pal, and Siva Reddy. AgentRewardBench: Evaluating automatic evaluations of web agent trajectories. In *Conference on Language Modeling (COLM)*, 2025. arXiv:2504.08942.

Yifu Lu, Shengjie Liu, and Li Dong. OrchDAG: Complex tool orchestration in multi-turn interactions with plan DAGs. *arXiv preprint arXiv:2510.24663*, 2025. v1, 2025-10-28.

Seiji Maekawa, Jackson Hassell, Pouya Pezeshkpour, Tom Mitchell, and Estevam Hruschka. Towards reliable benchmarking: A contamination free, controllable evaluation framework for multi-step LLM function calling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. v2, 2026-02-06; arXiv:2509.26553.

OpenAI. Introducing SWE-bench Verified, August 2024. URL `https://openai.com/index/introducing-swe-bench-verified/`. Published August 13, 2024; updated February 24, 2025. Accessed September 26, 2026.

Hafsa Ouajdi, Francesco Giannuzzo, Alaa Boukhary, Paolo Papotti, Gerard Conangla, and Adam Elwood. Execution-First synthetic tool-use trace generation for LLM agents. *arXiv preprint arXiv:2607.29175*, 2026. v1, 2026-07-31.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *PMLR*, 2025.

Andrei Cristian Popescu, Haitz Sáez de Ocáriz Borde, and Pietro Liò. Looped language models improve compositional tool calling. *arXiv preprint arXiv:2608.18171*, 2026. v1, 2026-08-17.

Md Nakhla Rafi, Md Ahasanuzzaman, Dong Jae Kim, Zhijie Wang, and Tse-Hsun Chen. FALAT: Tracing failures in LLM agent trajectories via dependency-guided search. *arXiv preprint arXiv:2606.00765*, 2026. v1, 2026-05-30.

Priyam Sahoo, Gaurav Mittal, Xiaomin Li, Shengjie Ma, Benjamin Steenhoek, Pingping Lin, and Yu Hu. AgentLens: Revealing the lucky pass problem in SWE-agent evaluation. *arXiv preprint arXiv:2605.12925*, 2026. v3, 2026-06-02.

Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. TaskBench: Benchmarking large language models for task automation. *arXiv preprint arXiv:2311.18760*, 2024. v4, 2024-11-01 (camera-ready); v1 2023-11-30. NeurIPS 2024.

Kou Shi, Ziao Zhang, Shiting Huang, Avery Nie, Zhen Fang, Qiuchen Wang, Lin Chen, Huaian Chen, Zehui Chen, and Feng Zhao. AsyncTool: Evaluating the asynchronous function calling capability under multi-task scenarios. *arXiv preprint arXiv:2605.27995*, 2026. v2, 2026-05-28.

Yewei Song, Xunzhu Tang, Cedric Lothritz, Saad Ezzini, Jacques Klein, Tegawendé F. Bissyandé, Andrey Boytsov, Ulrick Ble, and Anne Goujon. CallNavi, a challenge and empirical study on LLM function calling and routing. In *Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2025. v2, 2025-04-24; arXiv:2501.05255.

Tianda Sun and Dimitar Kazakov. Tool-call dependency structure is linearly decodable in LLM agent residual streams. *arXiv preprint arXiv:2605.25310*, 2026. v1, 2026-05-25.

SWE-agent contributors. mini-SWE-agent, 2025. URL `https://github.com/SWE-agent/mini-swe-agent/releases/tag/v1.0.0`. Software, version 1.0.0.

Jiaming Wang, Ziteng Feng, Jiangtao Wu, Ruihao Li, Qianqian Xie, Yuxiang Ren, He Zhu, Xueming Han, Fanyu Meng, Junlan Feng, and Jiaheng Liu. Where do deep-research agents go wrong? Span-level error localization in agent trajectories. *arXiv preprint arXiv:2606.02060*, 2026a. v1, 2026-06-01; v2, 2026-06-02 (title and author list unchanged). Introduces TELBench and DRIFT.

Jiaxuan Wang, Yulan Hu, Wenjin Yang, Zheng Pan, Xin Li, and Lan-Zhe Guo. Aligning agents via planning: A benchmark for trajectory-level reward modeling. *arXiv preprint arXiv:2604.08178*, 2026b. v2, 2026-05-11. Introduces Plan-RewardBench.

Eden Wu, Sonia Castelo, Yurong Liu, Cláudio T. Silva, and Juliana Freire. AgentTrails: Towards trust and reuse for agentic tasks. *arXiv preprint arXiv:2607.18816*, 2026. v1, 2026-07-21.

Jiaqi Wu, Qinlao Zhao, Zefeng Chen, Kai Qin, Yifei Zhao, Xueqian Wang, and Yuhang Yao. GAP: Graph-based agent planning with parallel tool use and reinforcement learning. *arXiv preprint arXiv:2510.25320*, 2025. v1, 2025-10-29.

Mengsong Wu, Tong Zhu, Han Han, Chuanyuan Tan, Xiang Zhang, and Wenliang Chen. Seal-Tools: Self-instruct tool learning dataset for agent tuning and detailed benchmark. *arXiv preprint arXiv:2405.08355*, 2024. v1, 2024-05-14.

Minrui Xu, Zilin Wang, Mengyi Deng, Zhiwei Li, Zhicheng Yang, Xiao Zhu, Yinhong Liu, Boyu Zhu, Baiyu Huang, Chao Chen, et al. EnvFactory: Scaling tool-use agents via executable environments synthesis and robust RL. *arXiv preprint arXiv:2605.18703*, 2026. v1, 2026-05-18.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Junjie Ye, Zhengyin Du, Xuesong Yao, Weijian Lin, Yufei Xu, Zehui Chen, Zaiyuan Wang, Sining Zhu, Zhiheng Xi, Siyu Yuan, et al. ToolHop: A query-driven benchmark for evaluating large language models in multi-hop tool use. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. v4, 2025-05-20; arXiv:2501.02506.

Guoyao Yu, Xiaoqing Sun, Ziqi Huang, Shaojing Fan, Zhongyi Zhang, Xiaomeng Hu, Xiaobo Xue, Yangyang Shi, Xiong Xiao, Yang Song, et al. Getting the parameters right: A difficulty-graded benchmark and probe-guided training for LLM tool calls. *arXiv preprint arXiv:2608.03071*, 2026. v1, 2026-08-04.

Chenkai Zhang, Yiran Li, Yifang Tian, Michalis Bachras, and Hans-Arno Jacobsen. When agentic executions fail: Detecting and localizing runtime faults from telemetry. *arXiv preprint arXiv:2608.14680*, 2026. v1, 2026-08-04.

Yue Zhao. GRADE: Graph representation of LLM agent dependency and execution. *arXiv preprint arXiv:2606.22741*, 2026.

Shuaiyu Zhou, Fengpeng Yue, Zengjie Hu, Yuanzhe Shen, Chenyang Zhang, Feng Hong, Cao Liu, and Ke Zeng. ToolVerse: Unlocking massive environments and long-horizon tasks for agentic reinforcement learning. *arXiv preprint arXiv:2607.15660*, 2026. v1, 2026-07-17.

Dongsheng Zhu, Weixian Shi, Zhengliang Shi, Zhaochun Ren, Shuaiqiang Wang, Lingyong Yan, and Dawei Yin. Divide-Then-Aggregate: An efficient tool learning method via parallel tool invocation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025a. v2, 2025-05-25; arXiv:2501.12432.

Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Jasjeet Sekhon, Jacob Steinhardt, Antony Kellermann, Sarah Schwettmann, Matei Zaharia, Ion Stoica, Percy Liang, and Daniel

Kang. Establishing best practices for building rigorous agentic benchmarks. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f316275b44ee2de533102913828a8107-Paper-Datasets_and_Benchmarks_Track.pdf. arXiv:2507.02825. Introduces the ABC checklist.

A PRE-REGISTRATIONS AND THE DEVIATION LOG

Five families of confirmatory protocols govern the adjudicated results in this paper. Each was committed before its confirmatory run and names its own thresholds, its family of tests, and the action to take on every outcome, failure included; prior pilots and subsequent deviations are disclosed.

Table 3: Pre-registered protocols and the results they govern.

protocol	what it governs	status of the result
fixed-call replay (protocol: dependency oracle)	τ-bench, six arms, per-edge and ladder	confirmatory
BFCL held-out	400 disjoint tasks, four hypotheses	confirmatory
human comparison (rater A)	200 blind stratified pairs, two stages	confirmatory
SWE-bench (two protocols)	random-pair arm; reward-level arm	confirmatory
BFCL generator sweep	scale, and within-task correctness	confirmatory

The BFCL base arm carries no confirmatory protocol and is reported as exploratory throughout. Two further exploratory analysis plans, for typed exact equality and for effect-location/adjacent robustness, were committed before their implementations and full-corpus outputs; they define complete reporting branches but re-adjudicate no confirmatory result. Rater C's later stage-1 ratings form a separate supplementary analysis (HV-5) and do not extend the original confirmatory adjudication.

Deviations. We maintain a single append-only deviation log. Every entry carries a timestamp and states whether it was written before or after the run it governs; those that affect numbers or interpretations in this paper are summarised below.

- **A1**: replay fidelity was reframed from a blocker into an inclusion filter. Trajectories that do not replay are excluded rather than counted, which is why every corpus reports admitted trajectories.
- **GM-1**: the initial gold BFCL figures were computed from the first fifth of the task set and were not representative. On the task population the LLM arm uses, gold is MECH 57.3% / STATE 38.0%, not $\sim 74\%$ / $\sim 60\%$. The paper uses the corrected values.
- **HV-1**: the second rater's J1 is near-degenerate. The degeneracy was logged before any agreement statistic was computed, which is why the κ of 0.025 is reported as a bound from marginals and not as a finding made afterwards.
- **HV-2, HV-3**: stage-2 parameters and rater configuration, both fixed before the stage-2 sample was drawn.
- **HV-4**: three BFCL trajectories replay non-deterministically and were excluded under the protocol's own exclusion clause, before the stage-2 draw.
- **HV-5**: independent rater C was added on stage 1 after the original analysis. All three pairwise comparisons and C's unfavourable J1 statistics are reported separately; A's adjudication is unchanged and stage 2 remains A-only.
- **TAG-1**: the harness-tag artefact of Appendix F in the BFCL base arm. After recomputation without the affected trajectories, every conclusion holds or strengthens.
- **TAG-2**: the same artefact in the held-out arm. Because excluding those pairs would add a criterion the frozen protocol does not contain, the published values stand and the exclusion is reported only as a sensitivity analysis. Under it all four outcome classifications are unchanged (H-C1–H-C3 pass; H-C4 fails) and MECH moves $24.0\% \rightarrow 23.4\%$. Six affected STATE-positive pairs are removed, yet both STATE rates round to 12.0%.

- **SR-2**: the SWE-bench reward-level arm was re-adjudicated on the benchmark's actual reward in place of a patch proxy, and it met the criterion at 3.80%.
- **SW-6**: the largest generator was abandoned on measured cost, and the two hypotheses specified at that scale were declared not adjudicable rather than re-pointed at a smaller arm after the fact (Section 6).
- **SW-7**: the cause SW-6 recorded for that abandonment was wrong. A defaulted device argument had left every sharded run with a single visible card, so the first attempt's slowdown was a harness defect and not a memory limit, and the sharding gate that had passed under it was void. With the defect fixed the gate was re-run and passed ($189/190$ across two and across four devices), the arm was measured at the rate reported in Section 6, and it was abandoned a second time, on that cost and on the termination-regime confound. The disposition set by SW-6 was not amended, so the two hypotheses are still not adjudicable and the 14B values are still descriptive.
- **N2-1**: the original position control used an inert normalised later position and truncated pair inputs. Re-running on all $33{,}137$ pairs with tool-rank normalisation preserves the $6/6$ gate verdict but changes "masking in every arm" to $5/6$. The correction was logged after the rerun.
- **EL-1**: before implementation or any location result, source inspection showed that the owning τ-bench runner identity-matches every baseline action, `RESPOND` pseudo-actions as well as tool calls. The frozen amendment restores that owning-runner rule; pair endpoints remain tool calls.

B PER-ARM TABLES

This appendix gives the full arm-level results that the main text summarises. The six τ-bench arms are in Table 23, and Table 27 reports their ladders.

Corpora and denominators. **τ-bench** (Yao et al., 2024): six arms, comprising gpt-4o in retail and airline at each of reward 0 and 1 (four arms), and sonnet-3.5 in retail and airline at reward 1 (two arms). There are $1{,}353$ trajectories, $33{,}137$ pairs and 366 task-level clusters summed over arms, no caps. The protocol was frozen before the confirmatory run, with the pilot's numbers disclosed in it. The **reward$= 0$ arms are a selection control**. Of the admitted trajectories, 48 have fewer than two tool calls and supply no pairs, leaving $1{,}305$ pair contributors. Another 72 have exactly two tool calls, leaving $1{,}233$ eligible for the 12-draw severity analysis ($14{,}796$ draws). Admission, pair contribution and severity eligibility are therefore different denominators.

BFCL v4 multi-turn (Patil et al., 2025), base arm: 190 trajectories generated by Qwen2.5-32B against BFCL's stateful Python environment, $2{,}511$ pairs transposed, of which 188 trajectories and $2{,}484$ pairs enter the corrected figures (Appendix F). This arm is exploratory and has no frozen protocol.

BFCL v4 held-out: 387 trajectories from 400 tasks disjoint from the base arm, $5{,}978$ pairs, 387 clusters. **The protocol was committed before generation**, which makes this the confirmatory replication.

No confirmatory pair-audit verdict was unstable under the seed check in Section 3.4.

Pair-level results and balanced metrics. In the six τ-bench arms, the audited token-overlap value-reuse rule flags 60–84% of all ordered pairs as dependent and carries lift $\mathbf{1.14}$–$\mathbf{1.29}$, so knowing that the rule asserts an edge raises the probability of a MECH-positive outcome under the specified in-context transposition by between 14% and 29% over knowing only which arm the pair belongs to. Meanwhile `b_write`, one bit derived from a baseline replay with per-call state access, flags 22–33% of pairs at precision 0.55–0.78 and lift $\mathbf{2.64}$–$\mathbf{3.51}$. It has higher precision in every arm and higher recall in three of six; the pre-registered paired lift difference excludes zero in $6/6$ arms after Holm correction.

Because lift reads only the predicted-positive set, we ran an exploratory balanced check over the same frozen pairs, specified after Gate 1 and not used to re-adjudicate it. The rule's MCC is 0.124–

0.175 and its informedness is 0.135–0.220; b_write reaches 0.571–0.750 and 0.621–0.781, respectively. Paired task-clustered differences exclude zero in every arm (MCC $+0.404$–$+0.617$; informedness $+0.401$–$+0.644$). Corrected as one Holm family, all 12 one-sided comparisons are significant ($p_{\text{Holm}} \leq 0.0006$). Appendix Table 5 gives coverage, precision, recall, false-positive rate, and both balanced measures.

BFCL replication. The rule falls within the same narrow band in every corpus: lift 1.14–1.29 across the six τ-bench arms, and on BFCL 1.18 (MECH) with 1.02 (STATE) on the base arm against 1.26 and 1.28 on the held-out one. Because τ-bench and BFCL differ in tooling, in domain, in trajectory generator, and in base order-sensitivity, the narrowness of that band says more than any single value inside it, and it shows that **the rule's near-inertness is not specific to one environment**. On the held-out arm this was pre-registered: the practical-null band $[0.80, 1.50]$ was fixed before generation, and the observed $1.26/1.28$ falls inside it. The BFCL base figures exclude the harness artefact of Appendix F, whereas the held-out confirmatory population is kept frozen and paired with a tag-free sensitivity. Table 4 tabulates every BFCL arm.

The advantage of b_write also replicates. On the held-out arm, the paired difference is $+0.57$ $[+0.34, +0.82]$ on MECH and $+1.61$ $[+1.25, +2.01]$ on STATE, both excluding zero as pre-registered; on the base arm, $+0.54$ $[+0.18, +0.92]$ and $+1.87$ $[+1.35, +2.44]$. This replication reverses the coverage ordering that motivated the ceiling concern: on BFCL, b_write flags 28–29% of pairs while the rule flags only 14–15%, yet its MECH MCC is still higher (base 0.255 vs. 0.041; held-out 0.295 vs. 0.059). The paired MCC differences are $+0.214$ $[+0.114, +0.310]$ and $+0.236$ $[+0.175, +0.296]$, respectively.

Table 4: BFCL v4 multi-turn, every arm measured. Lifts are against each arm's own base. The 7B arm predates the harness and hardware the rest of the study uses and its per-predictor lifts were never computed; it is listed because it is the only available point below 14B. The 14B arm is descriptive only: it was substituted for a larger arm after the protocol had fixed its hypotheses at that larger scale, so it does not count as a replication (SW-6). Predictor cells report MECH/STATE. BFCL base uses the tag-clean population (188 trajectories, 2,484 pairs). The held-out row keeps its frozen population; the parenthesised MECH rate is its tag-free sensitivity on 384 trajectories and 5,930 pairs, not a re-adjudication (Appendix F). * is the pre-registered confirmatory arm.

arm	generator	traj	pairs	MECH	STATE	rule	b_write
7B (exploratory)	Qwen2.5-7B	36	448	10.5%	4.9%	—	—
32B ablation (n=39)	Qwen2.5-32B	39	465	27.0%	12.0%	1.16 / 1.08	1.67 / 2.88
base	Qwen2.5-32B	188	2,484	23.3%	10.6%	**1.18 / 1.02**	**1.72 / 2.89**
held-out*	Qwen2.5-32B	387	5,978	24.0% *(23.4)*	12.0%	**1.26 / 1.28**	**1.83 / 2.89**
14B (descriptive)	Qwen2.5-14B	192	2,861	21.7%	10.1%	1.08 / 0.93	1.93 / 3.22
gold (*not an agent*)	—	188	3,003	57.3%	38.0%	—	—

In the arms where adjacency was computed, it takes the values base 0.56 / 0.51, held-out 0.566 $[0.528, 0.604]$ / 0.510 $[0.455, 0.564]$, 32B ablation 0.58 / 0.53, and 14B 0.53 / 0.45.

Directional associations across environments. The shared-state example in Figure 1 illustrates one possible path: a later writer moves ahead of intermediate reads. The symmetric move is available too, since pushing an earlier writer later removes its mutation from in front of a read. Because nothing in the replay definition favours either direction, any a_write/b_write asymmetry is an empirical association.

In τ-bench, a_write carries lift 0.97–1.97 against b_write's 2.64–3.51. The position analysis in Section 4.4 establishes a conditional association for b_write; it does not directly test the adjusted difference between a_write and b_write.

On BFCL held-out, the pre-registered directional hypothesis fails: a_write (2.22) is nominally ahead of b_write (1.83) on MECH. The 14B arm shows the same ordering (2.40 against 1.93). The later-call predictive advantage is therefore specific to τ-bench in these results. The weaker write–rule comparison holds across the eight primary arms: the paired lift difference excludes zero in all six τ-bench arms and both BFCL arms. Neither comparison identifies a direct causal mechanism.

Table 5: Coverage and balanced robustness for the two headline predictors. Each slash-separated cell is `rule`/`b_write`; q is the positive prediction rate and $J = \text{recall} - \text{FPR}$. Specificity is $1 - \text{FPR}$ and balanced accuracy is $(1 + J)/2$. The paired intervals are task-clustered with 20,000 resamples. The twelve one-sided τ-bench comparisons (six arms $\times$ MCC and J) form one post-Gate exploratory Holm family; all have $p_{\text{Holm}} \leq 0.0006$. BFCL is descriptive here. The labels M and S denote MECH and STATE. The base arm is tag-clean; held-out uses the frozen population. Removing its tag artefact changes ΔMCC to $+0.243$ $[+0.182, +0.304]$ on MECH and $+0.402$ $[+0.347, +0.456]$ on STATE.

arm / outcome	q	precision	recall	FPR
gpt-4o retail r=1 / M	.604/.315	.329/.777	.708/.873	.563/.098
gpt-4o retail r=0 / M	.621/.284	.344/.775	.727/.747	.577/.090
sonnet retail r=1 / M	.625/.328	.306/.727	.725/.902	.589/.122
gpt-4o airline r=1 / M	.780/.253	.213/.568	.891/.772	.754/.134
gpt-4o airline r=0 / M	.613/.222	.238/.605	.792/.728	.572/.108
sonnet airline r=1 / M	.835/.248	.178/.548	.949/.871	.814/.133
BFCL base / M	.145/.294	.275/.400	.171/.504	.137/.230
BFCL held-out / M	.135/.285	.304/.440	.171/.521	.124/.210
BFCL base / S	.145/.294	.108/.307	.148/.848	.145/.228
BFCL held-out / S	.135/.285	.153/.346	.173/.821	.130/.212

arm / outcome	J	ΔJ (95% CI)	MCC	ΔMCC (95% CI)
gpt-4o retail r=1 / M	.144/.775	$+.631$ $[+.567, +.701]$	.133/.750	$+.617$ $[+.561, +.679]$
gpt-4o retail r=0 / M	.150/.657	$+.507$ $[+.434, +.589]$	.141/.664	$+.523$ $[+.460, +.591]$
sonnet retail r=1 / M	.137/.781	$+.644$ $[+.590, +.696]$	.124/.733	$+.609$ $[+.565, +.654]$
gpt-4o airline r=1 / M	.137/.638	$+.500$ $[+.346, +.683]$	.129/.571	$+.442$ $[+.324, +.598]$
gpt-4o airline r=0 / M	.220/.621	$+.401$ $[+.220, +.565]$	.175/.579	$+.404$ $[+.252, +.529]$
sonnet airline r=1 / M	.135/.738	$+.603$ $[+.462, +.752]$	.132/.620	$+.488$ $[+.373, +.659]$
BFCL base / M	.034/.274	$+.240$ $[+.137, +.344]$	.041/.255	$+.214$ $[+.114, +.310]$
BFCL held-out / M	.047/.311	$+.264$ $[+.203, +.327]$	.059/.295	$+.236$ $[+.175, +.296]$
BFCL base / S	.003/.621	$+.617$ $[+.525, +.706]$	.003/.420	$+.417$ $[+.335, +.491]$
BFCL held-out / S	.043/.610	$+.567$ $[+.500, +.632]$	.041/.439	$+.398$ $[+.343, +.452]$

Baseline correctness and task composition. We pre-registered an equivalence test for whether baseline correctness affects MECH, with a ROPE of ± 5.0 pp, in the one setting that can hold the state mechanism fixed: BFCL's own 200 tasks, generators at two scales, gold sequences as the anchor. **The result indicates an effect rather than equivalence.** On $\Delta(t, g) = \text{MECH}(\text{gold}_t) - \text{MECH}(\text{llm}_{t,g})$ over 1,133 task–generator rows in 577 clusters, the difference-in-differences between the solved and failed strata is $\mathbf{-25.85}$ pp $[-30.27, -21.04]$ on MECH and -19.61 pp $[-23.17, -15.71]$ on STATE. Both lie outside the ROPE and exclude zero in $100/100$ bootstrap seeds, whereas in τ-bench the same comparison at the same level excludes zero in $0/100$. Four qualifications must accompany this estimate, since without them it overstates the effect by roughly a factor of two.

First, about half of it is composition rather than correctness. Decomposed by which side of the gap moves, the gold side falls -13.35 pp between the strata (45.19% MECH on tasks the agent solved, 58.54% on tasks it failed) and the LLM side rises $+12.50$ pp (31.73% against 19.24%). The tasks an agent solves are already 13 pp less order-sensitive *when measured on gold*, which is a property of the task set and not of correctness. Within-task differencing removes the task main effect from Δ, but the two strata are still different populations. *Second, that decomposition is post-hoc and descriptive.* The protocol fixed only the adjudication rule (CI excludes zero, therefore effect), and no decomposition was pre-registered. *Third, the reported estimator is the protocol's fallback rather than its primary design.* The within-task analysis would have placed agent trajectories on both sides of the comparison. It has 40 discordant clusters against a threshold of 100 fixed in advance, so it is underpowered and we report it unadjudicated. *Fourth, the correct arm uses gold sequences rather than agent-generated trajectories.*

Because solved tasks are themselves less order-sensitive, the result leaves open whether between-level variation reflects state mechanisms or sequence correctness. Accordingly, we use Appendix C only for the containment identity and endpoint screening and do not treat it as an axis of state mechanism.

Why the 72B arm is absent. There is no 72B row. As discussed in Section 6, a five-task probe placed Qwen2.5-72B across four A100-80GBs with no offload and measured 122 s per task, or about $414 for the full arm. The additional time does not reflect more tool use: on the same tasks, the model emits 26 tool calls compared with 27 for the 32B model, while taking $12\times$ longer per generation call. This leaves decode length as the likely explanation. Our trajectories record calls and not token counts, so we infer, without measuring it, that the 32B model stops at end-of-sequence after a short call list while 72B runs to the 220-token ceiling. The arm was dropped on that inference and not on price: a 72B arm would have carried a different termination regime from every other arm in the sweep, so generator scale would no longer have been the only factor changing. The available fixes (a lower ceiling for that arm alone, or a faster inference stack) would remove the confound only by breaking comparability with the existing arms. Scaling from 14B to 32B changes MECH by about 1.6 pp (21.7% to 23.3%), whereas going from 7B to 14B roughly doubles it, so we do not expect a 72B arm to fill the gap. That expectation is extrapolated from two arms and has not been measured.

Reading lift alongside coverage. For a predictor P with positive coverage and a positive outcome base, we report $\mathrm{lift}(P) = \mathrm{precision}(P) \,/\, \mathrm{base}$, where $\mathrm{base} = \Pr[\mathrm{MECH} = 1]$ within the same arm, and its positive rate $q = \Pr[P = 1]$. Since the true-positive rate as a population share is at most $\min(\mathrm{base}, q)$,

$$\mathrm{lift}(P) \;\le\; \frac{1}{\mathrm{base}}, \qquad \mathrm{lift}(P) \;\le\; \min\left\{\frac{1}{\mathrm{base}}, \frac{1}{q}\right\} \text{ when } q \text{ is fixed.} \tag{2}$$

The first bound is global, and the second gives the tighter headroom for a predictor with fixed coverage; neither is a performance score. A predictor independent of MECH has expected lift 1 at any q, but predictors with different q have different attainable precision. We therefore use lift only as positive-set enrichment and report q, recall, false-positive rate, informedness ($J = \mathrm{recall} - \mathrm{FPR}$), and Matthews correlation coefficient (MCC) beside it (Appendix Table 5). We do not normalise performance by either ceiling. The global bound does have one useful consequence: at base 0.95 even a perfect classifier cannot exceed lift 1.05, a limited headroom that Appendix B illustrates. For this reason a pre-registered rule excludes arms with base outside $[0.05, 0.60]$ from every lift claim; Section 6 identifies them.

A one-sided check on known causal links. PDDL specifies action preconditions and effects. From these descriptions and planner-generated plans, we compute plan-specific support links: for each precondition, our extractor identifies the most recent preceding action that adds it, stopping if an intervening action deletes it. These computed links supply known positive cases for the one-sided check without defining order sensitivity exhaustively. PDDL replay observations record each action's applicability and state changes.

We ran the replay test on 113 planner-generated plans across 12 PDDL domains (Alkhazraji et al., 2020), covering 27,634 pairs, with the positive-direction criterion fixed in advance at $\Pr(\mathrm{MECH} = 1 \mid \text{computed causal link} = 1) \ge 0.95$. **The measured conditional rate is 1.000 in all 12 domains**: the test misses none of the computed links in this sample. This observed rate does not establish specificity, a universal guarantee, direct-edge recovery, or validity on LLM trajectories. Lift eligibility and ceiling calculations are reported below.

Why PDDL is not a third audit environment. The pre-registration admitted only domains whose base rate fell in $[0.05, 0.60]$ into any lift comparison, and required at least three such domains. The discriminative subset came out at **zero domains and zero pairs**: all 12 domains are near ceiling, with goal-break base rates from 90.6% to 99.1%. No measured domain occupies the intermediate range. The arm's two lift hypotheses were therefore declared not adjudicable. Its positive check is reported separately from descriptive ceiling arithmetic.

Lift at ceiling. Pooled GOAL-break base is 0.953, so Equation 2 limits even a perfect classifier to lift 1.049. The computed-link predictor itself has GOAL-break precision 0.993, recall 0.120, and lift 1.042; for MECH its precision is 1.000 against the same recall of 0.120. Passing the positive check therefore amounts neither to perfect classification nor to a GOAL-break precision of one. The descriptive values show why the eligibility band is needed and adjudicate no lift hypothesis.

Table 6: PDDL, twelve domains. **The plan and pair counts in this table are descriptive and do not reproduce run to run.** The runner admits a problem only if its transposition sweep completes inside a wall-clock timeout, so composition depends on machine load; measured on one domain across three runs it returned 13 plans / 3,031 pairs, 13 / 2,325, and 14 / 3,623. The last column is the observed conditional rate $\Pr(\mathrm{MECH} = 1 \mid \text{computed causal link} = 1)$; it was 1.000 in every domain in the four generated corpora, not a universal structural guarantee. GOAL-break is this environment's reward level. The stored aggregate covers all 27,634 pairs, but the pair export retains only the first 20,000 rows.

domain	plans	pairs	GOAL-break	causal links	MECH | causal
gripper	4	1,360	90.6%	10.3%	**1.000**
tpp	6	820	91.7%	15.6%	**1.000**
logistics	12	3,139	92.0%	9.8%	**1.000**
psr-small	3	84	92.9%	50.0%	**1.000**
rovers	13	4,102	93.3%	8.9%	**1.000**
zenotravel	12	1,939	94.4%	12.0%	**1.000**
blocks	13	2,842	95.5%	15.0%	**1.000**
depot	4	904	95.5%	12.1%	**1.000**
satellite	12	3,488	96.7%	12.6%	**1.000**
elevators	14	3,330	96.8%	13.5%	**1.000**
transport	9	2,555	98.6%	9.7%	**1.000**
miconic	11	3,071	99.1%	9.4%	**1.000**
pooled	**113**	**27,634**	**95.3%**	**11.5%**	**1.000**

Table 7: SWE-bench Verified (OpenAI, 2024), `bash-only` mini-SWE-agent trajectories, 30 instances sampled uniformly at random under a fixed seed with 40 random pairs each. The rows use the same intervention type, but their reported denominators differ after level-specific fidelity exclusions. For REWARD, 144 outcomes were directly evaluated: all 114 DIFF-positive pairs and a 30-pair DIFF-negative control with $0/30$ violations. The remaining 856 are containment-implied reward-negative outcomes. Containment $\mathrm{REWARD}/\mathrm{DIFF} = 0.333$ $[0.159, 0.500]$ is the figure Appendix C uses.

level	what changed	rate	scale
MECH	an identity-matched observation	95.18% [90.4, 98.9]	1,120 pairs / 28 inst.
DIFF	the produced patch (bounds the reward)	11.40%	1,000 pairs / 25 inst.
REWARD	benchmark score; containment for untested negatives	3.80% [1.50, 6.70]	1,000: 144 direct + 856 implied

C Outcome-level differences and containment

The three levels are not noisy readings of one quantity. In the original audit populations, on solved baselines, reward changes exactly when the final state hash changes in the observed sample: across the four solved τ-bench arms, 676 state changes produce 676 reward changes and **zero** exceptions in 24,492 pairs. Failed baselines instead meet a score floor. Writing sol for a baseline that scores,

$$\Pr[\mathrm{REWARD}] = \underbrace{\Pr[\mathrm{sol}] \cdot \Pr[\Delta\mathrm{state} \mid \mathrm{sol}]}_{\text{solved baselines}} + \underbrace{\big(1 - \Pr[\mathrm{sol}]\big) \cdot \Pr[\mathrm{fix} \mid \neg\,\mathrm{sol}]}_{\text{failed baselines}}, \tag{3}$$

where probabilities refer to the evaluated pairs and the solved term uses the observed state–reward agreement on solved baselines. The pair-weighted solved share can differ from a trajectory-level solve rate; a low share can therefore conceal state changes at the reward level. The fix term is small but non-zero, so dropping it gives only a lower-bound approximation. Screening another scorer requires checking its solved-baseline relationship, retaining sampling uncertainty and accounting for the fix term: the lower bound alone cannot certify membership of the two-sided eligibility band. This accounting does not isolate an agent's state mechanism. Table 8 gives the baseline-stratum

split and falsifying cell; the correction term is detailed below. Figure 3 plots the three levels and eligibility band.

Table 8: Observed reward changes are contained in state changes in this sample. Every transposed pair from the six τ-bench arms, split by whether the baseline trajectory satisfied the benchmark's scoring function. An arm is a (model, domain, baseline reward) cell, so each arm falls entirely in one stratum: the solved stratum draws on both models, while the failed stratum is `gpt-4o` only, since no reward-0 arm was run for `sonnet-3.5`. "Reward without state" is the cell that would falsify the identity; it is empty in both strata and in every individual arm.

baseline	arms	pairs	state moves	reward moves	reward w/o state
solved	4	24,492	676	676	**0**
failed	2	8,645	361	19	**0**

Why the failed stratum is nearly blind to the score. On a baseline that already fails, 361 state changes produce 19 reward changes. The shortfall is a floor effect in the scoring rule and not a property of the environment: a trajectory scored zero has nowhere to fall, and a transposition would have to reconstruct the gold state to raise it. That cell measures the fix term of Equation 3, which is 0.00% in τ-bench airline and 0.36% in retail; in retail it supplies 0.14 of the arm's 1.31 pp marginal, about 11% of the total. Because the fix term is non-negative, dropping it gives a lower bound. On its own that bound cannot certify that a candidate lies inside $[0.05, 0.60]$, and a bound below 0.05 cannot rule the candidate out, although a lower bound above 0.60 would exclude it at the upper end, subject to estimation uncertainty. The decomposition uses the solved share of pairs and the state-change rate conditional on solved baselines.

Evidence from the second environment. On SWE-bench, the measured patch proxy has a patch-diff rate of 11.40%, while reward is 3.80% $[1.50, 6.70]$. Of the $1{,}000$ reward labels, 144 are directly evaluated outcomes (all 114 DIFF-positive pairs plus a 30-pair DIFF-negative falsification control with $0/30$ violations), and 856 are containment-implied reward-negative outcomes under $\mathrm{REWARD} \subseteq \mathrm{DIFF}$. Thus $\mathrm{REWARD}/\mathrm{DIFF} = \mathbf{0.333}$ $[0.159, 0.500]$: the patch proxy overstates reward-level sensitivity roughly threefold. Reward–state containment is observed on all $33{,}137$ τ-bench pairs, with the falsifying cell empty in all six arms. SWE-bench's weaker reward–patch containment is consistent with its control and **not an independent replication** of that identity. Its containment control evaluates only 30 of 886 DIFF-negative pairs; the other 856 are labelled by containment rather than direct evaluation. Patch containment does not establish equality between state and reward changes on solved baselines. Equation 3 requires that additional relationship, and we do not claim it for every environment.

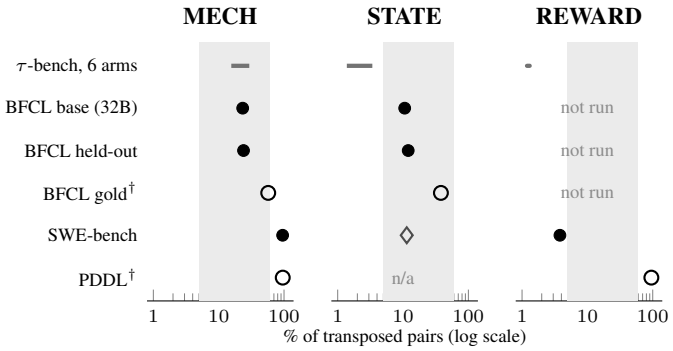


Figure 3: Replay results at three outcome levels. Rows are grouped by environment, not ranked; † marks a row whose trajectories are not agent-generated (BFCL gold sequences, planner output). Filled markers are measured agent arms, the bar on the τ-bench row spans its six arms, and the open diamond on SWE-bench's state column is the patch-diff proxy rather than a state hash. The shaded band is the pre-registered $[0.05, 0.60]$ base-rate interval: an arm outside it cannot carry a lift claim. High base rates limit headroom by Equation 2; low base rates leave few positives. Across the level-specific admitted populations, the reported rate changes by more than an order of magnitude: SWE-bench reports 95.2% on the 28-instance MECH population and 3.80% on the nested 25-instance reward population. Rates must therefore be reported with their measurement level and denominator. Eligibility for a lift claim depends on where each arm falls relative to the band. Its position depends on the outcome definition, task population, trajectory generator and admission rules as well as the scoring function; Equation 3 does not separate these influences. BFCL's own checker was not run as a reward here, so those cells are empty; PDDL has no state level distinct from its goal check. The SWE-bench reward marker combines 144 directly evaluated reward outcomes with 856 containment-implied negatives; the DIFF-negative control found $0/30$ containment violations.

D HUMAN COMPARISON, IN DETAIL

Judgments and information conditions. Each item collects two judgments in a fixed order. J1 shows the calls only, with no observations, and asks whether swapping the two calls changes what the rest of the trajectory actually does; the answer is locked before anything else is shown. J2 repeats the question with both observation sequences revealed. Both judgments offer Cannot tell. The instructions ask for functional differences, such as a call succeeding instead of failing, reading different data, acting on a different record, or producing a different end state; wording, formatting, field order, ID-only, and timestamp changes do not count. Revealing the observations first would make MECH recoverable by comparing identity-matched outputs, which motivates the order of the judgments and limits the interpretation of J2 agreement.

Table 9: Pre-registered human comparison with the MECH replay label, pooled over 200 blind stratified pairs. Effective sample size 157.5; no cannot-tell gate tripped. Bars were fixed before the first draw and adjudication uses author-rater A. Independent raters B and the later-added C cover stage 1 only; their comparisons appear separately in Tables 10 and 11.

	statistic	estimate	95% CI	bar	
C1	$\Pr(\text{J2 Yes} \mid \text{MECH+})$	0.840	$[0.760, 0.910]$	≥ 0.70	**pass**
C2	$\kappa_w(\text{J1}, \text{MECH})$	0.705	$[0.594, 0.816]$	≥ 0.60	**pass**
C3	$\Pr(\text{J1 Yes} \mid \text{MECH}-)$	0.093	$[0.041, 0.153]$	≤ 0.10	**pass**

Sampling. The eight sampling cells are environment (τ-bench, BFCL held-out) $\times$ MECH ($+$, $-$) $\times$ stage. We draw 200 pairs, 25 per cell, with at most one pair per trajectory across the pooled sample. Bootstrap resampling stays within those cells, $N_{\text{BOOT}} = 20{,}000$ at seed 0; effective sample size is 157.5. The unstratified intervals are $[0.765, 0.908]$, $[0.586, 0.815]$ and $[0.040, 0.155]$ and do not change any verdict.

Point-estimate adjudication and intervals. All three of A's criteria pass on the pooled 200 using the pre-registered point-estimate rule. Across 100 bootstrap seeds, C1's interval clears its pass line in $100/100$, C2's in $18/100$, and C3's in $0/100$; all three exclude their fail lines in $100/100$. The protocol calls intervals "mandatory context, never the adjudication rule." Thus the official verdict does not imply that every interval clears its pass threshold.

The two-stage escalation. Stage 1 ($n = 100$) returned CONCERN because C3 missed its bar by one item. That verdict triggered a clause fixed in advance, which required drawing a disjoint 100 under a named seed and adjudicating the pooled 200 once, against the same thresholds. There was no stage 3. Stage 1's verdict is interim and is *not* available as independent confirmation, because its items sit inside the pooled estimate and not beside it.

Rater A's cannot-tell responses. The cannot-tell response was available and mandatory for both judgments, with a 40% gate that would have invalidated the instrument. The observed rates were 0.045 pooled on J1, 0.030 on J1 within MECH$-$, and 0.000 on J2 within MECH$+$.

Rater A's disagreements. 33 distinct items disagree with the replay label, giving 41 disagreeing (item, judgment) pairs, 25 on J1 and 16 on J2. The nine that decide C3 are all J1 predictions the rater withdrew at J2, and all nine have $\text{STATE} = 0$: the rater expected an order effect from reading the calls, and retracted it on seeing that the two executions produced identical observations.

Reliability and supplementary rater C. A is an author and rated both stages, while B and C each rated the same 100 stage-1 items. C's responses arrived on 22 September 2026, after the original analysis, and C had not seen A/B responses or prior MECH results. B's marginals (J1 No on $99/100$ items) bound the original A–B J1 κ to $[-0.021, +0.025]$. C answered J1 Yes on $41/100$, but C's agreement with MECH is still low. No independent ratings were collected for stage 2, and none are planned; A had seen the stage-1 analysis before rating stage 2. Two one-sided drift triggers

Table 10: Pairwise agreement on the same stage-1 items. Cannot tell is excluded separately for each pair and judgment. κ is ordinary Cohen's agreement coefficient; it is distinct from the population-weighted J1–MECH κ_w used for C2.

	J1: calls only			J2: observations revealed		
Pair	n	**Agreement**	κ	n	**Agreement**	κ
A–B	95	55.8%	0.025	95	85.3%	0.665
A–C	95	72.6%	0.443	99	94.9%	0.896
B–C	100	60.0%	0.029	94	86.2%	0.691

Table 11: Stage-1 comparison with MECH, reported without re-adjudication. A's values here use only stage 1, unlike Table 9. C's intervals use $20{,}000$ bootstrap samples within the four environment-by-MECH cells, seed 0; each item is a distinct trajectory. The frozen weights are 0.48044 for MECH-positive and 1.51956 for MECH-negative items. Cannot tell is excluded for the statistic using it.

Statistic	**A**	**B**	**C**	**C: 95% CI**
C1: $\Pr(\text{J2 Yes} \mid \text{MECH}+)$	0.820	0.533	0.840	$[0.740, 0.940]$
C2: $\kappa_w(\text{J1}, \text{MECH})$	0.671	0.030	0.265	$[0.101, 0.438]$
C3: $\Pr(\text{J1 Yes} \mid \text{MECH}-)$	0.104	0.000	0.260	$[0.160, 0.380]$

were fixed before stage 2 was labelled. Neither was triggered, and both moved away from the drift direction, which is consistent with no drift but does not exclude it.

C1 counts are $41/50$, $24/45$, and $42/50$ for A, B, and C; C3 counts are $5/48$, $0/50$, and $13/50$. C has no J1 Cannot tell responses and one at J2, on a MECH-negative item. C's C3 is $2/25$ in τ-bench and $11/25$ in BFCL, a descriptive breakdown without separate adjudication. Its 0.260 point estimate is one Yes response above $12/50 = 0.240$, and its interval crosses the 0.25 fail boundary, and C2's interval likewise crosses its 0.40 fail boundary. These statistics are supplementary; we neither replaced B nor adjudicated by majority vote. The script `mech_human_validation/analyze_rater_c.py` reuses the original estimators and writes `results_raterC_supplement.json` with input hashes and coded item responses. This analysis has no 100-seed stability check.

Response transitions on MECH-negative items. Among MECH-negative items, J2 Yes counts are $0/100$ for A, $0/50$ for B, and $0/49$ for C, with one C response of Cannot tell. A withdrew all nine pooled J1 Yes predictions at J2. Of C's 13 such predictions, 12 became No and one became Cannot tell; all these A and C items have $\text{STATE} = 0$. The pattern separates what raters predict from the calls alone from what they judge after observing execution, but overlapping items and recoverable labels prevent interpreting it as independent evidence of a zero miss rate.

Observation change versus functional endorsement. Six MECH-positive items receive J2 No from all three raters: `item004`, `item009`, `item015`, `item041`, `item062`, and `item071`. In `item009` and `item041`, the changed observations differ only in post IDs, which the rater instructions exclude as a functional difference. That account comes from the stored traces; the raters were not asked to explain their decisions. The other four include changes in returned data, so unanimous No cannot generally be classified as a formatting-only effect. MECH and functional endorsement are distinct measurements.

E SWE-BENCH BYPRODUCTS

The trajectory-fidelity gate revealed three properties of the published SWE-bench evaluation that are incidental to our claims but relevant to reproduction.

The evaluation is not network-hermetic. `psf__requests-1724` loses 27 tests to `ConnectionError` against `httpbin`. Its 2013-era suite reaches the public internet, so its score depends on external availability, and a harness that assumes instances are self-contained will silently mis-score it.

At least one published verdict does not reproduce. `pylint-dev__pylint-6528` is `resolved: false` in the official report. Our container returns `true` in all five baseline runs. The three tests it gains are `test_multiprocessing[1]`, `test_multiprocessing[2]` and `test_custom_should_analyze_file`, all multiprocessing tests, which are classically sensitive to host CPU count. The discrepancy therefore appears to reflect host-dependent behaviour in the published evaluation.

A published log disagrees with its own pinned image. For `astropy__astropy-14369` the pinned `:v1` image's pytest writes parametrised test ids in `ascii_escaped` form, which does not match the ids in the published log for the same instance.

F EXPLORATORY EXTRACTOR AND EFFECT-LOCATION ROBUSTNESS

Both analyses in this section run on the already frozen pair populations and stored MECH/STATE outcomes, regenerate no agent trajectory, and re-adjudicate no confirmatory verdict. Their plans were committed before implementation and extraction. All contrasts use task-clustered bootstrap intervals with 20,000 resamples and a 100-seed sign ledger, and we read a confidence interval containing zero as inconclusive, never as evidence of equivalence.

Pair-comparison inference and analysis status. For reference, the historical pair comparisons use 20,000 task-clustered bootstrap samples, recomputing paired differences within each sample. The original τ-bench family uses Holm–Bonferroni across six arms; confirmatory verdicts are rechecked across 100 seeds and reported as unstable if the flip rate exceeds 5%. Balanced comparisons were specified after Gate 1, and the typed analysis was designed with earlier results available but frozen before extraction. Both are exploratory and do not re-adjudicate the original gates. The later published/schema extractor comparison is also exploratory; its 50,000-draw paired task bootstrap, marginal 95% intervals and Bonferroni adjustment over 24 τ-bench MECH contrasts are documented in Appendix J.2. BFCL and STATE in that comparison are supplementary. Moving these newer comparisons to the main text does not change their inferential status.

Typed exact equality. Destination arguments are parsed as JSON in τ-bench and as literal Python call arguments in BFCL. Source observations are parsed as JSON only when their representation supports it and are otherwise kept as raw strings. Null and empty strings are excluded, booleans never equal numbers, numerically equal integer and decimal spellings match, and strings require exact Unicode equality. `typed_all` marks any type-and-value match; `typed_unique` requires a single prior tool-call producer. Both labels describe trace representations and do not verify provenance.

Extractor comparison results. The extractor comparison tests how much of the gap comes from the token-overlap implementation. In an exploratory analysis frozen before extraction, `typed_all` replaces token-overlap matching with exact equality of representation type and canonical value. On the same available pairs its MECH MCC is 0.13–0.29, against 0.05–0.18 for the token-overlap rule. The paired gain excludes zero in $6/8$ arms (all three retail arms, sonnet airline, and both BFCL arms); the two gpt-4o airline arms are inconclusive, which is not evidence of equivalence. The stricter `typed_unique` variant has lower MECH MCC than `typed_all` in all eight arms. Its point estimate is below the token-overlap rule's in all six τ-bench arms and above it in both BFCL arms, but only the held-out paired interval excludes zero at the reporting seed, so requiring the exact prior match to be unique does not give a portable substitute for provenance. These results retain the historical populations; Figure 2 now shows the later published-extractor comparison after known-artefact exclusions. Tables 12 and 13 give the complete exploratory matrix.

Effect location and adjacency. For an identity-matched action k (including non-tool actions), let $D(k)$ denote an observation change. MID_ONLY requires some $D(k)$ strictly between i and j and

Table 12: Typed exact-equality robustness for MECH. N is the typed-available population and coverage is relative to the frozen arm. MCC columns use that same population. Δ is `typed_all` minus the token-overlap rule with a paired task-clustered 95% interval. The six wholly positive intervals are stable in $100/100$ seeds; the other two contain zero in $100/100$.

arm	N	cov.	rule	typed_all	typed_unique	b_write	ΔMCC (95% CI)
gpt-4o retail $r{=}1$	5,695	100%	.133	.231	−.132	.750	$+.098$ $[+.065, +.130]$
gpt-4o retail $r{=}0$	5,250	100%	.141	.235	−.160	.664	$+.094$ $[+.058, +.129]$
sonnet retail $r{=}1$	14,539	100%	.124	.255	−.155	.733	$+.131$ $[+.105, +.156]$
gpt-4o airline $r{=}1$	989	100%	.129	.188	−.045	.571	$+.059$ $[-.033, +.181]$
gpt-4o airline $r{=}0$	3,395	100%	.175	.209	.027	.579	$+.034$ $[-.044, +.102]$
sonnet airline $r{=}1$	3,269	100%	.132	.292	−.033	.620	$+.160$ $[+.024, +.278]$
BFCL base	2,373	95.5%	.046	.134	.099	.261	$+.088$ $[+.029, +.149]$
BFCL held-out	5,740	96.0%	.063	.138	.108	.299	$+.075$ $[+.037, +.113]$

Table 13: The corresponding STATE MCC matrix on the typed-available population. The last column classifies the paired `typed_all`–rule interval. "Unavailable" means rare outcome classes made more than 5% of bootstrap replicates undefined; it is not a zero effect.

arm	rule	typed_all	typed_unique	b_write	paired result
gpt-4o retail $r{=}1$	.051	.076	−.057	.111	positive
gpt-4o retail $r{=}0$	.057	.136	−.068	.113	positive
sonnet retail $r{=}1$	.072	.128	−.043	.212	positive
gpt-4o airline $r{=}1$	.032	.005	−.082	.209	unavailable
gpt-4o airline $r{=}0$	−.010	.042	−.013	.146	inconclusive
sonnet airline $r{=}1$	.017	−.005	−.055	.114	unavailable
BFCL base	.004	.087	.083	.426	positive
BFCL held-out	.039	.092	.094	.442	positive

none at an endpoint or after j; END_OR_TAIL requires a change at either endpoint or after j. Location is available only when replay completes, all expected identities are recovered, recomputed MECH agrees with the stored label, and no pre-treatment observation changes. Adjacent cells require at least 200 pairs, 20 positives and negatives, and positives from 10 task clusters, plus predictor support and bootstrap availability.

This supporting analysis retains the original pair populations. Effect location describes where the changes appear. Depending on the arm, $\mathbf{14.9}$–$\mathbf{53.8}\%$ of MECH-positive pairs are **MID_ONLY**: only observations of intermediate actions change. END_OR_TAIL is the majority in $6/8$ arms and reaches 85% in both BFCL arms, so changes are not confined to intermediate observations. Their causal path is not identified, however. Interval interactions can also propagate to an endpoint or later action, and the location categories therefore cannot establish how much sensitivity is caused by direct endpoint interaction.

We also compare write status and the rule after restricting the outcome to END_OR_TAIL. The `b_write` MCC difference stays positive in every primary arm, and its interval excludes zero in $5/6$ τ-bench arms and both BFCL arms, with no negative interval. The write advantage therefore is not confined to intermediate-only changes (Appendix Table 14).

Adjacency predictor and restricted audit. The original predictor `adj` is one when $j = i + 1$ in the original action sequence, with no intervening action of any kind. The audit also records `same_tool` and dist $= j - i$ as covariates. Restricting the audit to adjacent pairs removes intervening calls; the frozen support rules above leave sparse cells non-adjudicable rather than pooling them.

Adjacent transpositions have no intervening action. Compared with the full-arm base, their MECH rate is lower: in the four τ-bench arms with adjacent pairs, adjacency carries MECH lift $\mathbf{0.02}$–$\mathbf{0.21}$, below one throughout. The two sonnet arms have no adjacent pairs under the original-action-index definition, so their adjacent lift is undefined. On BFCL base, adjacency lift is $0.56/0.51$ for MECH/STATE; on held-out it is 0.566 $[0.528, 0.604]$ and 0.510 $[0.455, 0.564]$. Every nonempty

primary adjacent subset therefore has lower observed sensitivity than its full arm, though these arm-level observations are not six statistically independent replications.

The adjacent restriction is also sparse. It is outcome-adjudicable only for BFCL base (109 positive pairs in 78 task clusters) and held-out (254 in 162), and in both `b_write` keeps its MCC advantage over the token-overlap rule ($+0.180$ $[+0.083, +0.276]$ and $+0.189$ $[+0.124, +0.256]$). Under the frozen support rule, every τ-bench adjacent cell is non-adjudicable, including the two empty arms. The four nonempty τ-bench cells thus describe negative enrichment and say nothing either way about a predictor gap. Appendix Table 14 reports every arm.

Table 14: Effect location among MECH-positive pairs and the `b_write`–rule MCC contrast. END/TAIL intervals are computed over all location-available pairs. Adjacent +/clusters gives positive pairs/task clusters; only BFCL meets the frozen adjudicability rule, so other adjacent contrasts are shown as non-adjudicable regardless of their point estimates.

arm	loc. cov.	MID only	END/tail	ΔMCC END/tail	adj. +/clusters	ΔMCC adjacent
gpt-4o retail r=1	100%	53.8%	46.2%	$+.120$ $[+.068, +.172]$	3/3	non-adjudicable
gpt-4o retail r=0	100%	44.0%	56.0%	$+.072$ $[-.005, +.147]$	8/6	non-adjudicable
sonnet retail r=1	100%	52.1%	47.9%	$+.158$ $[+.106, +.210]$	0/0	non-adjudicable
gpt-4o airline r=1	100%	28.3%	71.7%	$+.292$ $[+.160, +.428]$	5/2	non-adjudicable
gpt-4o airline r=0	99.4%	34.4%	65.6%	$+.199$ $[+.073, +.318]$	14/6	non-adjudicable
sonnet airline r=1	100%	24.7%	75.3%	$+.362$ $[+.229, +.516]$	0/0	non-adjudicable
BFCL base	100%	14.9%	85.1%	$+.182$ $[+.075, +.287]$	109/78	$+.180$ $[+.083, +.276]$
BFCL held-out	99.8%	15.0%	85.0%	$+.198$ $[+.132, +.261]$	254/162	$+.189$ $[+.124, +.256]$

In the held-out tag-free sensitivity, typed coverage is 95.99%, `typed_all` reaches a MECH MCC of .144 against .067 for the rule, and MID_ONLY is 14.3%, so neither conclusion changes. Complete confusion matrices, informedness contrasts, location patterns, seed ledgers, and pair-level availability reasons are released in `typed_value_equality.json` and `effect_location_adjacent.json`; no arm is pooled.

Correcting a BFCL harness artefact. BFCL's execution helper prefixes method names with an instance name containing the run tag and a UUID. Its replacement also operates inside string literals, so for a small number of trajectories it changes argument text and reaches the observation; in two cases it also reaches stored state. Baseline and transposed runs then differ for a reason unrelated to order, and the pair is labelled MECH-positive regardless. The defect was found by the human-comparison packet checker; the replay-fidelity gate could not detect it. Excluding the affected trajectories moves the base arm from MECH $24.1\% \to 23.3\%$ and STATE $10.8\% \to 10.6\%$ (2 trajectories, 27 pairs, all 27 MECH-positive and exposed to the artefact). The lower base rate raises the headroom in Equation 2, but this bound alone does not determine a predictor's lift, because the predictor's positive-set precision changes too. The corrected comparisons therefore recompute both quantities, and the recomputation preserves the reported inferential conclusions. Reported Historical BFCL-base estimates in this appendix use the corrected data. The frozen held-out population is not retroactively corrected and is instead reported with a tag-free sensitivity.

Exploratory same-order replay and predictor stability. We froze the E0 plan and code before running this supplemental diagnostic, although the earlier results were already known by then. For all 1,305 pair-contributing τ-bench trajectories we recovered the original admission order and pair keys, and each trajectory then received two same-order baselines, two write-set collections and two token-graph collections, all computed with the owning functions. We also checked all 190 raw BFCL-base and 387 frozen held-out trajectories: the same calls were run under tags `b`, `b`, and `p0_1`, each with a fresh UUID, and all write-status prefixes were collected twice. Observation equality kept its environment-specific definition (Equation 1). All 1,882 requested trajectories completed, and none was left unchecked or lost to a harness failure.

Null-MECH counts were $0/1{,}305$ in τ-bench, $2/190$ in raw BFCL base and $3/387$ in frozen held-out. In both BFCL comparisons the null replay found the previously known cases and no others: base IDs 30 and 194, and held-out miss-function IDs 30 and 194 plus miss-parameter ID 194. In each case one observation changed. The two ID-30 trajectories also showed null-STATE and prefix-state changes, affecting six stored pairs each; their tweet-content string contains a function-like

token that the helper rewrites with the instance name. A separate mechanism probe that used fresh instances and a fixed name removed the null difference in all five cases, and changing only that name reproduced the observation differences and both state differences. The difference therefore arises in interpreted string arguments and not solely in output logging. No field masking was applied to primary outcomes.

In every checked trajectory, write bits, token-overlap labels, typed labels on available pairs, and typed availability were stable and matched the frozen baseline predictors, and τ-bench reward matched both across repeats and against the recorded reward. The absolute prefix-state variation in the two ID-30 cases left their write bits unchanged. These checks cover the baseline predictors and do not rerun the stored transposition outcomes. No null difference was observed in the 188 tag-clean base or the 384 tag-free held-out trajectories, so the $1{,}880$-trajectory primary population retains three null-MECH cases (48 pairs of exposure), one of which is also a null-STATE case (6 pairs).

The diagnostic's exclusion set equals the existing tag-free set. On the same typed-available held-out pairs, the descriptive MECH MCC difference `b_write` minus `typed_all` is $+.1612$ in the frozen primary population ($5{,}740$ pairs) and $+.1674$ in the tag-free sensitivity ($5{,}692$ pairs); both are point estimates, and no new paired intervals were computed. Held-out STATE rates are $716/5{,}978$ and $710/5{,}930$. That both round to 12.0% does not imply the absence of a state artefact. The frozen populations and verdicts are unchanged. The separate null-replay archive records trajectory-level hashes, prefixes, predictor vectors, errors, and reward/length strata. Because this census used a finite number of repeats, its zero observed differences do not prove general determinism.

G REPRODUCTION

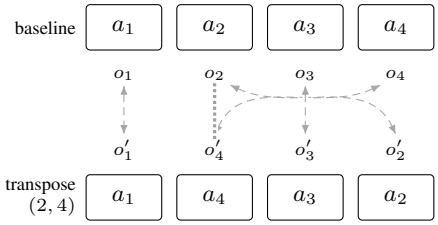
Table 15: Corpus-specific replay admission and fidelity criteria. No single byte-identity rule is applied across environments.

corpus	admission / fidelity criterion	analysis consequence
τ-bench	An unmodified replay of an upstream historical trajectory reproduces its recorded reward.	Reward-0 trajectories can enter; observation byte identity is not the gate.
BFCL	A generated trajectory executes without error under the pinned v4 harness and has at least three model-emitted calls; the held-out all-pairs set also requires at most twelve.	Base primary removes the known tag artefact. Held-out primary remains the frozen 387-trajectory population and uses the tag-free result only as sensitivity.
SWE-bench	MECH replay in a pinned `:v1` image reproduces the published submission byte-for-byte ($28/30$). The reward arm additionally checks published grading, baseline scoring and stability, and regenerated DIFF labels.	The reward primary has 25 instances; DIFF reproduction is $146/146$, and the DIFF-negative containment control has $0/30$ violations.
PDDL	`pyperplan` produces a 3–40 action plan, its baseline reaches the goal, and both solving and the all-pairs sweep finish under separate 45-s caps.	Wall-clock admission makes corpus composition machine-load dependent; the positive-link check is empirical and the lift hypotheses are not adjudicable.

Observation comparison. Identity-matched keying was fixed in the pre-registration; positional keying was forbidden because a displaced slot compares different calls (Figure 4). The function $\mathcal{C}_{\text{env}}$ in Equation 1 follows the outcome-generating runner. For τ-bench, the rendered observation is parsed as JSON when possible and serialised with recursively sorted object keys and compact separators; whitespace inside JSON string values is preserved. If parsing fails, whitespace runs are collapsed and leading/trailing whitespace is stripped. For BFCL, the runner compares each observation's `str()` representation exactly, without JSON canonicalisation or whitespace normalisation. Both audits thus match observations by call identity while applying different observation equality rules, and all reported labels keep these original rules.

Observation-comparison diagnostics. In the τ-bench pilot, positional keying compares different calls at displaced slots; the pre-registration records approximately 99.3% positives. The figure is empirical. Differing call identities do not guarantee a positive, since different calls can return the same value, and the pilot separately found raw and canonical observation base rates equal at the reported decimal precision. Both checks concern the τ-bench comparison procedure. All reported

MECH analyses use identity-matched observations, with τ-bench canonicalisation and BFCL exact-string equality as specified in Equation 1.



Dashed: o_2 versus o'_2 (same call). Dotted: o_2 versus o'_4 (same slot, different calls).

Figure 4: Observation matching for fixed-call replay, transposing $(2, 4)$. Each call's observation is compared with its own baseline wherever it moves (Equation 1). Comparing the same position would instead compare different calls at a displaced slot; unequal outputs there alone do not establish a change to either call's output.

Token-overlap implementation. For `rule`, both runners extract case-sensitive regex matches of length at least four and assert an edge when the two token sets intersect. The shared pattern is

```
[A-Za-z0-9_#][A-Za-z0-9_#.\-]{3,}
```

In τ-bench the inputs are `str(observation)` and `str(kwargs)`; in BFCL they are `str(observation)` and the destination call text after its first opening parenthesis. Tokens can therefore come from field names as well as values, and two tokens match only when they are equal; arbitrary character-substring containment does not count. Section 2.1 distinguishes this implementation from the published character-substring and schema-typed oracles. Earlier protocols and result files called it the substring rule, and the stored field is still named `rule`. The corrected wording changes no labels or pre-registered verdicts.

Table 16: Information used to construct each predictor or reference. Complete observation–argument logs suffice for value matching in principle; the baseline column records our implemented procedure. The final row constructs an outcome-based reference. Pairwise replay is also needed to obtain the outcome labels against which all predictors are evaluated.

Predictor / reference	Information	Baseline replay used	Per-call state	Pairwise sweep
`rule`	Source observations and destination arguments	τ-bench graph reconstruction	No	No
`typed_all`, `typed_unique`	Parsed scalar types, values, and prior matches	Observation recovery in both corpora	No	No
Public substring / ID	Observation values, arguments and frozen normalisation	Shared baseline cache	No	No
Schema-ID	Same logs plus fixed tool/path entity mapping	Shared baseline cache	No	No
`a_write`, `b_write`	Data hash before and after each call	Yes	Yes	No
`adj`	Original action indices	No	No	No
MECH for $V_{\texttt{mech}}$	Baseline and transposed observations	Yes	No	Yes

Predictor information access and execution cost. Table 16 separates predictor construction from outcome measurement. We compare the typed and token-overlap predictors on the same typed-available pairs. Write status is a diagnostic baseline that draws on additional execution and state

information, more than either of them needs, so its performance does not come from a comparison at equal information access.

Implemented execution cost. Information access and replay count are different requirements. With m tool calls and n total actions, τ-bench collects write status during one baseline pass (n action executions). BFCL instead replays each prefix of length $0, 1, \ldots, m$ in a fresh instance: $m + 1$ resets and $m(m + 1)/2$ tool-call executions. Its current implementation thus needs $O(m)$ prefix replays and $O(m^2)$ calls, compared with one instrumented pass in τ-bench. A complete pair sweep uses $\binom{m}{2}$ full replays, or $O(m^2 n)$ action executions, in addition to its baseline. All of these are execution counts; we did not measure wall time, and the counts do not assert equal information access.

What each environment needs. τ-bench uses historical trajectories shipped with the upstream repository, not trajectories generated here. BFCL requires a pinned `bfcl_eval` release carrying the v4 multi-turn data and `transformers` 4.57.1; generation is greedy, so there is one trajectory per task. PDDL uses `pyperplan` and its benchmark suite. SWE-bench replays released `bash-only` mini-SWE-agent v1.0.0 (SWE-agent contributors, 2025) trajectories from the 2025-07-26 submission using `claude-sonnet-4-20250514`.[2] We replay the recorded commands without regenerating model responses, using pinned `:v1` images with digests recorded. The released reward configuration pins SWE-bench Verified dataset revision `c104f840cc67f8b6eec6f759ebc8b2693d585d4a` and SWE-bench harness v4.1.0 for log parsing; the official evaluation trace determines the test command.

Three things that will not reproduce, and why. First, the PDDL corpus composition (Table 6). Second, byte-identity of BFCL trajectories across hardware: regenerating the 32B arm on a different accelerator reproduces the aggregate (success $94/561$ against $93/562$) while byte-identity is 83.7%, and a control attributed the difference to the hardware move and not to any code change. Third, exact BFCL numbers under an unpinned install, for the version reason given in the reproducibility statement.

Released materials. The materials include pair records, aggregate analysis outputs, scripts, the five confirmatory protocol families, both frozen exploratory plans, the coded survey records, and the deviation log. Statistical tables are transcribed from their analysis outputs; protocol, information-access, and survey tables also use design documents and coding records. The historical PDDL pair export is truncated at $20{,}000$ of $27{,}634$ rows, so only its stored aggregate supports the complete reported table, and a new timeout-limited run would not restore the missing historical rows. The historical severity writer likewise retained only aggregates. The separate review supplement builds a new variant archive by fixed-seed re-execution, but this does not retroactively make the historical export complete; its aggregate-reproduction check and the intervention/weighting sensitivities are reported in Appendix I. The local run guide keeps the replay and statistical environments separate, writes to a fresh output directory, and leaves the frozen primary results in place.

H THE DEPENDENCY-CONSTRUCT SURVEY

We recoded the same 34 works on 2026-09-26 using their pinned versions. Within the scope below, 19 use value/data provenance alone (sole), 12 combine it with other relations (mixed), and three have no qualifying value relation (other). Thus $31/34$ include value/data provenance. This is a recoding of a fixed cohort, not a new literature search or a re-screening of previously excluded works.

Frame. The discovery record dated 2026-08-31 reports forty queries over arXiv title, abstract and full text, combining `tool call`, `function call`, `dependency`, `nested`, `data dependency`, `tool dependency graph`, `output of one ... input of another`, and named-benchmark title queries. It also reports a forward-citation sweep from Sun & Kazakov (2026), Barazandeh et al. (2026) and He et al. (2025): 799 distinct entries, 240 passing

[2] Public submission prefix: `bash-only/20250726_mini-v1.0.0_claude-sonnet-4-20250514/` in `https://swe-bench-submissions.s3.amazonaws.com/`.

a domain filter, 46 full-text reads, and 34 inclusions with 12 exclusions. These historical retrieval counts have not been independently reproduced by the present recoding. The recorded selection sought works that define, construct or consume inter-call dependencies. The retained cohort also contains tool-type, resource, reasoning-step and graph-free perturbation methods; it cannot be described as 34 concrete tool-call-pair graphs. Table 22 reproduces the historical exclusion list.

The sample is neither exhaustive nor a systematic review. BFCL (Patil et al., 2025) was not found in the frozen arXiv search frame; this is not a claim that no arXiv version exists. One coder checked the included works' definitions, methods, examples and appendices and recorded pinned versions and PDF-page evidence. No independent second-rater check was made.

Coding unit and rule. The counting unit is a work. We code the relations it explicitly defines, constructs or uses within tool-call plans, workflows and trajectory evaluation. This includes output–input binding, runtime graph extension and information flow between evaluation milestones, as well as graph topology. We do not infer new dependency definitions from general programming capabilities, ordinary serial execution, or constraints within a single call.

V denotes an earlier output, field or artifact, or an explicitly source-preserving transformation, supplying a later argument, explicit input or consumed resource. Planned source binding counts; generic type compatibility without a producer–consumer binding does not. Sole means that only V justifies the coded relations. Mixed means that the work also uses a relation that can hold without V, including type, state, control or temporal conditions; these may occur in different graph paths or domains. Other means no qualifying V relation was identified, including graph-free methods. These categories do not describe every feature or every possible execution of a system.

Relation meaning, extraction evidence, initial specification and later updates, and validation target are recorded separately. LLM inference and artifact tracing are extraction methods, not alternative relation meanings. External benchmark labels used only as validation references are recorded under validation, rather than replacing a work's own construct: Sun's TaskBench replication is such a case. Tables 19, 20 and 21 summarise all 34 works.

Sensitivity to coding scope. Table 17 varies explicit coding rules, not statistical sampling uncertainty. Counting only graph topology for TaskBench, ControlLLM and ToolSandbox omits their output-binding or information-flow checks and moves those three from mixed to other. Counting IPIGuard's initial TDG alone omits runtime successor inheritance and moves it to sole. Broader readings of FALAT's general step-output influence and HyperTool's state-transition capabilities are also shown. The primary count uses concrete input/resource binding for V and explicitly defined relations, rather than all programming capabilities. It does not claim that Sun's external TaskBench labels are all V.

Table 17: Independent scope changes, except the explicitly combined final row. V incl. is sole plus mixed. These alternatives are not confidence intervals.

Coding scope	Sole	Mixed	Other	V incl.
Primary: all scoped operational relations	19	12	3	31/34
Graph topology only for the three works above	19	9	6	28/34
IPIGuard initial TDG only	20	11	3	31/34
FALAT general step-output influence included in V	19	13	2	32/34
HyperTool general state-transition capabilities included	18	13	3	31/34
Graph-topology restriction plus initial TDG only	20	8	6	28/34

Validation target. We found no included study evaluating whether an asserted edge predicts execution changes after transposing two calls while keeping call identities and concrete arguments fixed. This is a finding about the inspected versions, not a reproduction of every experiment. Several studies do validate other targets: AsyncFC compares access annotations with human labels and evaluates execution accuracy and latency; FlowScout executes candidate workflows and refines nodes and edges using execution feedback; OTAP compares scores with verifier-checked validity on 720 PlanBench plans. ToolSandbox scores stateful executions against milestones. FALAT (Rafi et al., 2026) uses an LLM-based search for corrected-step counterfactuals, rather than environmental

re-execution of a fixed reordered pair. These tests differ from our target; they are not evidence of an absence of execution-based validation in general.

Historical mention search. The earlier discovery record reports a full-text mention search, including reference lists, over nineteen works in its value-reuse group: nine mentioned no other included work, none mentioned Sun & Kazakov (2026), and the forward-citation query for that work returned no citing works. This nineteen-work subset is not the nineteen sole works under the new coding. Its exact work-level search ledger and historical query responses have not been independently reconstructed here. The reported absence of mentions does not establish independent invention or a complete account of intellectual contact.

Sensitivity to the search terms. The historical record marks sixteen of forty queries as presupposing value flow and five as title lookups. It reports $26/30$ recovery using construct-neutral, non-title queries and identifies four missed works: Seal-Tools, NESTFUL, NesTools and CallNavi. Conditional on that exclusion list, applying the new coding gives 15 sole, 12 mixed and three other among 30 works: $15/30 = 50\%$ sole and $27/30 = 90\%$ value-inclusive, compared with $19/34 = 55.9\%$ and $31/34 = 91.2\%$ in the full cohort. Sole is exactly half, not a majority, in this restricted set. This arithmetic is not a new reproduction of the reported $26/30$ recovery or the forty-query search.

Scope and implementation differences. We make no claim of completeness, field-wide prevalence or independent invention. The count describes only the retained cohort. The value-inclusive works implement distinct relations and extraction procedures, including exact matching, declared source binding, source-preserving transformations and constrained LLM inference. Their inclusion does not mean that all use Sun's substring rule or our token-overlap proxy. Source pages in the tables refer to PDF pages of the pinned versions, starting from the first page.

Source tasks and comparison scope. Table 18 distinguishes each source study's original target from the components and outcomes evaluated in the present audit.

Table 18: Source tasks and the scope of the present audit. Shared terminology does not make these extractors or scores interchangeable.

Study	Input and original target	Relation to this audit
Sun & Kazakov (2026)	Outputs/JSON arguments, schema-ID fields, and model residuals; decode an oracle dependency graph	We compare frozen public extractors on replay outcomes. Our own token/scalar matchers are separate baselines; the representation result is not tested.
OTAP (Barazandeh et al., 2026)	Attributed execution and reference graphs; transport alignment with semantic and structural costs	Its mixed-corruption ladder includes inversions and other changes. Our inversion count is not its alignment score or a reproduction of its validity experiment.
Plan-RewardBench (Wang et al., 2026b)	Judge prompts and formatting for pairwise plan comparison	Public prompt/formatter transferred to our reorderings; no reproduction of its dataset, labels or leaderboard.
Present audit	Fixed recorded calls, observations and optional state access; predicting replay changes	Pair-level proxy comparison and controlled inversion-count evaluation, with outcomes measured by re-execution.

Published oracles and the audited proxies. Sun & Kazakov (2026) define an oracle tool-call DAG by *value reuse*: an edge $i \to j$ is added when a contiguous, case-sensitive, whitespace-normalised substring of length at least four characters from call i's output occurs verbatim in the serialised JSON arguments of call j, so that call i's result supplies an argument of call j. Their independent typed-ID equality oracle performs no substring matching: its 312 edges are a subset of the 414 substring edges. With typed-ID edges as predictions and substring edges as the reference, precision is 1.000, recall 0.754, and overall agreement 0.910. These are oracle-agreement measures, not execution-grounded edge accuracy. TaskBench supplies independent DAGs for a separate representation-probe replication, not ground truth for the same retail edges. Their limitations

section scopes the construct: a dependency mediated by a transformed value (a count, a reformatted field), or by pure control flow, with call j issued because call i returned rather than because it reused the value, lies outside both of their oracles. Their study tests representation rather than behavioural control.

Our `rule` instead intersects regex-token sets from recorded outputs and argument text (Section 3). It is not an implementation of their character-substring oracle: equality of whole extracted tokens does not detect every contained substring, and the argument-rendering paths differ. Our exploratory `typed_all` compares type-preserving scalar values in recorded observations and arguments; it is not restricted to the typed-ID fields and entity returns used by their schema-typed oracle. Neither of our proxies verifies provenance. We therefore evaluate these trace-derived proxies against replay outcomes and do not attempt to reproduce or refute their representation results. A subsequent exploratory comparison applies the frozen public substring and typed-ID functions themselves to the same logs (Appendix J.2). Its public substring normalisation differs from the paper-literal rule above; both are preserved as separate comparators. Our new schema-ID matcher is a further implementation, not a claim to reproduce every source-paper oracle detail.

A work found outside the frozen frame. GRADE (Zhao, 2026) was identified after the survey was frozen and is not counted above. Its three corpus-specific attachment rules connect consequential writes to earlier database reads in τ-bench, accesses to the most recent access of the same resource in file/web tasks, and consumed identifiers to their most recent producing steps in τ^2-bench. The last is a V relation under our coding rule. GRADE evaluates failure prediction against run-size baselines and degree-matched attachment controls. It notes the lack of large-scale ground-truth reliance labels; it does not perform the fixed-call pair-transposition predictive test used here.

Survey claim revision. The pre-registration calls published evaluations *consumers* of the value-reuse rule. The recoding instead identifies a family of value/data provenance relations, often combined with other relations. It establishes neither adoption of a single rule, independent origins nor field-wide prevalence. The historical mention-search claim has the separate evidential limits stated above.

Perturbation scope. OTAP's Table 1 reports ROUGE-L pairwise ranking accuracy (PRA) of 84.4% for causal inversions (Barazandeh et al., 2026). PRA compares corrupted and uncorrupted trajectories; it is not edge precision or execution-effect accuracy. Surface metrics can thus detect reordering even when a value-reuse proxy predicts little fixed-call executional sensitivity. Our claim concerns which pairs predict replay changes and what edge-based severity scores rank. It does not concern whether disorder is visible at all. We claim no novelty on the vocabulary of faults: dependency corruption, unnecessary tool calls and blind retry are already named injected perturbations.

Table 19: The 19 sole works. Categories concern the scoped operational relations, not every feature of a system. Source versions are pinned in the survey record.

Work	Coded relation and extraction/update scope	PDF pp.
Sun & Kazakov (2026)	Substring and typed-ID output–argument matching on recorded traces.	3, 9, 13, 16
Fan et al. (2026a)	Runtime, before execution: exact source matching, transformed-source merging and conservative/LLM fallbacks.	5, 6, 18
Wu et al. (2026)	Post hoc producer–consumer graph: exact skeleton, semantic candidates and constrained LLM refinement.	1, 2, 4
Yu et al. (2026)	Upstream-output transfers as an inter-call difficulty feature; intra-call constraints are outside scope.	3, 5, 10
Wu et al. (2024)	Constructed nested calls pass an earlier response as a later parameter.	2, 5
Basu et al. (2024)	Constructed data-dependency DAG with explicit output-variable references.	2, 3, 6, 7
Han et al. (2025)	Constructed nested parameters reference previous call return values.	1, 5, 20
Ye et al. (2025)	Generated tool chains bind one tool's output to another's input.	3, 4, 6
Song et al. (2025)	Hard-split parameters require prior call responses.	3, 4, 6
He et al. (2025)	Generator specifies which predecessor outputs populate successor inputs.	5, 6
Maekawa et al. (2026)	A downstream function can consume its predecessor's output variable.	4, 6
Lu et al. (2025)	Constructed source-output-key to target-input-key binding.	3, 4, 5, 8
Khandelwal et al. (2026)	Generator maps predecessor output-schema fields to successor inputs.	2, 7, 12
Kulkarni et al. (2026)	Constructed chains reference prior responses; the parser resolves bindings.	3, 4, 14, 15
Ouajdi et al. (2026)	Executed output fields enter visible state and supply later inputs.	4, 5
Popescu et al. (2026)	Later arguments are functions of predecessor observations.	3, 17
Wu et al. (2025)	Planned dependency requires a predecessor output as input.	4, 6
Bhandari et al. (2026)	Typed dataflow graph; execution-guided mutations revise the plan.	3, 5, 6
Du et al. (2026)	Chaining subtype directly passes outputs to later arguments.	1, 11, 12

Table 20: The 12 mixed works. Categories concern the scoped operational relations, not every feature of a system. Source versions are pinned in the survey record.

Work	Coded relation and extraction/update scope	PDF pp.
Jang et al. (2025)	V plus contextual dependence in constructed interactions.	4, 6, 8
Shi et al. (2026)	Output-dependent generation plus state/control precedence in actual traces, including directory changes.	4, 5, 19, 21, 23
Xu et al. (2026)	Output/input matching plus logical refinement during construction.	5, 7, 8
Zhou et al. (2026)	Parameter binding plus semantic precedence in generated workflows.	4, 12, 13
Zhu et al. (2025a)	Planned input–output dependencies and logical causal relations.	3, 6
An et al. (2025)	Initial data TDG plus runtime query-node connections and successor inheritance.	4, 5, 6
Feng et al. (2026)	Runtime scheduler resolves output futures and overlapping state-access regions.	5, 12, 14
Hao et al. (2026)	Data/control workflows; real execution feedback guides node and edge revision.	2, 4
Shen et al. (2024)	Type-compatible or temporal source graphs plus explicit output-to-argument bindings.	4, 5, 6, 8, 20
Liu et al. (2023)	Resource-type graph plus assignment of concrete produced resources to later inputs.	4, 5, 6, 13, 23
Lu et al. (2024)	State/temporal milestones plus prior-output-dependent checks of later arguments.	4, 6, 16, 17
Barazandeh et al. (2026)	Artifact consumption or execution precedence; exact/explicit graphs, tracing and total-order fallback.	2, 4, 7

Table 21: The three other works. Categories concern the scoped operational relations, not every feature of a system. Source versions are pinned in the survey record.

Work	Coded relation and extraction/update scope	PDF pp.
Liu et al. (2024)	Usage-weighted tool transitions; no required output–input binding.	3, 4, 5
Wang et al. (2026b)	Dependency corruption is a text-perturbation template; no operational dependency graph.	5, 14
Rafi et al. (2026)	LLM-labelled influence among outputs, decisions and assumptions; no required concrete input/resource binding.	5, 6, 7, 15

Table 22: The 12 exclusions transcribed from the historical discovery record. The present audit recodes the retained 34 works; it does not re-adjudicate these exclusions under the expanded operational scope.

Work	Pinned arXiv version	Recorded exclusion reason
ComplexFuncBench	2501.10132v1	Parameter reasoning from user information; no defined inter-call relation.
Dynamic Tool Dependency Retrieval	2512.17052v4	Trajectory-conditioned retrieval; no explicit edge rule.
SimpleTool	2603.00030v1	Intra-call decoding dependency.
MeNTi	2410.13610v3	Nested unit conversion; no trajectory relation.
Proof-of-Use	2510.10931v2	Reasoning grounded in cited evidence identifiers.
UniToolCall	2604.11557v2	Call-level representation/evaluation; no dependency construct.
SING	2606.16591v2	Tool-discovery intention graph.
ToolRobustBench	2608.23635v1	Stage-wise perturbation evaluation.
HammerBench	2412.16516v2	Slot filling/argument shifting; no dependency relation.
FamilyTool	2504.06766v2	Knowledge-graph hops rather than call-output relations.
WebClipper	2602.12852v2	No inter-call dependency construct recorded.
Guided by Trajectories	2601.23032v1	No inter-call dependency construct recorded.

Table 23: Historical pre-registered per-edge audit on τ-bench. `rule` is the audited token-overlap value-reuse edge; `b_write` asks only whether the later call of the pair mutates the environment. The paired difference lift(`b_write`) − lift(`rule`) is the primary statistic, recomputed inside each cluster-bootstrap resample. Verdict stability is 100.0% over 100 seeds in all six arms.

			MECH	lift			
arm	**traj**	**pairs**	**base**	`rule`	`b_write`	**paired diff (95% CI)**	p_{holm}
gpt-4o retail r=1	251	5,695	28.1%	1.17	**2.77**	+1.60 [+1.30, +1.99]	0.0003
gpt-4o retail r=0	174	5,250	29.4%	1.17	**2.64**	+1.46 [+1.12, +1.99]	0.0003
sonnet retail r=1	565	14,539	26.4%	1.16	**2.75**	+1.59 [+1.35, +1.91]	0.0003
gpt-4o airline r=1	84	989	18.6%	1.14	**3.05**	+1.91 [+1.14, +4.05]	0.0003
gpt-4o airline r=0	102	3,395	18.4%	1.29	**3.28**	+1.99 [+1.13, +2.98]	0.0003
sonnet airline r=1	177	3,269	15.6%	1.14	**3.51**	+2.37 [+1.42, +5.27]	0.0003

I SUPPLEMENTARY DIRECT COMPARISONS AND SEVERITY CONTROLS

We designed these analyses after the historical results and the same-order replay control were known, and hashed their dated plan and code before running the new computations. They are exploratory supplements that preserve the original labels and verdicts and involve no new agent generation. The tables in this appendix retain their original analysis populations; the current exclusion-based comparisons are identified separately in Sections 5.2 and 5.3. The argument-integrity diagnostic in the last subsection was added after we had inspected these results, and it is post hoc.

I.1 DIRECT WRITE–TYPED COMPARISON

The original lift comparison and its frozen populations are retained in Table 23. Its paired write–rule difference excludes zero after Holm correction in all six τ-bench arms. The historical rule lift is 1.14–1.29, against 2.64–3.51 for write status. These estimates are separate from the exclusion-based published-extractor comparison in Section 5.2.

Both MCCs are computed on the same typed-available pair population and the same task draw. We retain the BFCL-base tag-clean population and the frozen held-out primary, and treat the held-out tag-free row as a separate sensitivity. From 20,000 resamples at seed 0 we report marginal 95% percentile intervals and nominal 99.375% Bonferroni intervals for the eight-arm MECH family; STATE is a separate eight-arm family. When more than 5% of bootstrap MCC differences are undefined, we suppress the interval instead of substituting zeros. This never happens for MECH, whose differences are defined in all draws. The same computation reproduces the historical same-support MCCs and typed-minus-rule intervals.

Across arms, `b_write` has MECH MCC 0.26–0.75. On MECH, the MCC of write status exceeds that of `typed_all` by +0.127–+0.519. In all eight arms, both the marginal 95% interval and the nominal Bonferroni-adjusted 99.375% interval exclude zero. The held-out difference is +0.161 [+0.105, +0.218], and its separate tag-free sensitivity gives +0.167 [+0.111, +0.224]. These intervals are exploratory and add no confirmatory verdict, and on STATE the results are mixed or insufficiently supported (Tables 24 and 25).

On this same support the archive also reports MCC and coverage for the rule, earlier-write and either-write predictors. Either-write is a descriptive comparator and was not selected as a replacement for the primary later-write predictor. We continue to report the BFCL reversal of the earlier/later directional association.

I.2 POSITION-CONFOUND CONTROL AND SENSITIVITY

Primary control and decision rule. The corrected primary analysis uses all 33,137 pairs in the six τ-bench arms. It fits $\text{MECH} \sim \texttt{b_write} + \texttt{rule} + i + j + \text{dist} + u_i + u_j$ on the standardised scale, where i, j are original action indices, $\text{dist} = j - i$, and u_k is the zero-based rank of tool call k among its trajectory's m tool calls, divided by $m - 1$. Inference uses 2,000 task-clustered bootstrap resamples. A model-free check compares MECH for $\texttt{b_write} = 1$ and $\texttt{b_write} = 0$ within

Table 24: Exploratory direct MECH comparison. N/T is available pairs/tasks; T/W are typed_all/b_write MCC. The final column adjusts the nominal interval level for eight comparisons. The sensitivity row is outside that family. Availability is 100% in τ-bench, 95.5% in BFCL base, 96.0% in held-out, and 96.0% in its tag-free subset.

Arm	N/T	MCC T/W	Δ	95% CI	99.375% CI
gpt-4o retail r=1	5,695/88	0.231/0.750	+0.519	[+0.465, +0.573]	[+0.445, +0.592]
gpt-4o retail r=0	5,250/68	0.235/0.664	+0.429	[+0.364, +0.506]	[+0.341, +0.535]
sonnet retail r=1	14,539/104	0.255/0.733	+0.478	[+0.431, +0.527]	[+0.410, +0.545]
gpt-4o airline r=1	989/32	0.188/0.571	+0.383	[+0.271, +0.533]	[+0.231, +0.605]
gpt-4o airline r=0	3,395/36	0.209/0.579	+0.370	[+0.233, +0.485]	[+0.175, +0.525]
sonnet airline r=1	3,269/38	0.292/0.620	+0.328	[+0.253, +0.457]	[+0.226, +0.518]
BFCL base	2,373/188	0.134/0.261	+0.127	[+0.040, +0.216]	[+0.005, +0.250]
BFCL held-out	5,740/387	0.138/0.299	+0.161	[+0.105, +0.218]	[+0.084, +0.239]
Held-out tag-free	5,692/384	0.144/0.311	+0.167	[+0.111, +0.224]	–

Table 25: The separate STATE comparison does not support an all-arm claim. Undefined fractions refer to paired bootstrap differences.

Arm	ΔMCC	95% CI	99.375% CI	Undefined
gpt-4o retail r=1	+0.035	[−0.060, +0.101]	[−0.099, +0.115]	0.005%
gpt-4o retail r=0	−0.022	[−0.136, +0.079]	[−0.181, +0.106]	0.000%
sonnet retail r=1	+0.085	[+0.020, +0.133]	[−0.011, +0.148]	0.000%
gpt-4o airline r=1	+0.204	–	–	36.280%
gpt-4o airline r=0	+0.104	[+0.010, +0.238]	[−0.020, +0.291]	0.155%
sonnet airline r=1	+0.120	–	–	12.530%
BFCL base	+0.339	[+0.264, +0.407]	[+0.233, +0.432]	0.000%
BFCL held-out	+0.350	[+0.301, +0.397]	[+0.283, +0.416]	0.000%
Held-out tag-free	+0.350	[+0.301, +0.397]	–	0.000%

(dist, u_j) bins, size-weighted as a risk difference. The pre-specified gate requires the later-write association to persist in at least four of six arms; otherwise the claim is restated as "actions late in a trajectory are more often MECH-positive." Disagreement between regression and stratification was pre-specified as "believe neither." N2-1 records the correction of truncated inputs and an inert normalised-position column (Appendix A).

Position concentration and controlled results. A motivating diagnostic on gpt-4o retail r=1 placed state-changing calls at mean relative position 0.904, median 1.000, with 97% in the back half of the trajectory. Such clustering could account for the apparent advantage of the later call, so we applied the pre-specified control of Section 4.4.

With task-clustered CIs, the controlled `b_write` coefficient excludes zero in every arm, and the stratified risk difference is positive in every arm, so the "believe neither" branch does not fire. Moreover, **the controlled coefficient is larger than the uncontrolled one in five of the six arms**: 2.000 against 1.917 in gpt-4o retail r=1, and 1.896 against 1.612 in sonnet-3.5 airline. The exception is gpt-4o airline r=0, whose controlled coefficient is marginally smaller (1.268 against 1.291). These coefficient changes alone do not demonstrate causal masking, since logistic coefficients can change with adjustment even without that mechanism. The later-write association therefore persists after the measured position controls, although the analysis does not identify a causal effect of write status.

The stratified risk differences range from 0.457 to 0.683 (descriptive point estimates; Table 26). These results retain the original populations; excluding the argument-aliasing trajectory also preserves the positive interval in the affected arm (Appendix I.6).

Exploratory reduction. In the original design the $j = i + \mathrm{dist}$ column is redundant: the augmented design has 8 columns but rank 7. Removing j leaves 7 columns of rank 7 in every arm. Both fits standardise predictors separately inside each resample and use scikit-learn 1.4.2 `LogisticRegression`, L2 `lbfgs`, $C = 10^6$, tol $= 10^{-4}$ and max_iter $= 2{,}000$ (NumPy 1.26.4, SciPy 1.13.1, Python 3.9.6). In this runtime the same 2,000 task draws at seed 0 reproduce

all six historical v2 point coefficients, intervals and stratified risk differences exactly, and no point or bootstrap fit raises a convergence warning. Under the reduced design the point coefficient moves by at most 0.0013, and every positive interval stays positive. Changes in a coefficient under adjustment do not alone identify causal masking.

Table 26: Exploratory position sensitivity. Coefficients are for standardised later-write; the reduced model omits j. Condition numbers use the standardised design augmented with the intercept. RD is the original stratified risk difference.

Arm	v2 β	Reduced β	Reduced 95% CI	Condition	RD
gpt-4o retail r=1	2.000	1.998	$[+1.680, +2.438]$	5.93	0.683
gpt-4o retail r=0	1.605	1.605	$[+1.428, +1.847]$	6.12	0.650
sonnet retail r=1	1.992	1.993	$[+1.699, +2.342]$	7.68	0.628
gpt-4o airline r=1	1.502	1.503	$[+1.203, +2.423]$	4.66	0.457
gpt-4o airline r=0	1.268	1.267	$[+0.941, +1.769]$	5.52	0.528
sonnet airline r=1	1.896	1.897	$[+1.401, +3.816]$	5.16	0.469

I.3 Severity reconstruction and eligibility

Pre-registered comparison and decision criteria. The original comparison used rule, write and MECH-based scores; typed scores were added only in the supplementary controls. Both decision criteria were fixed in advance: $\rho(V_{\texttt{rule}}, V_{\texttt{mech}}) < 0.60$, and $\Delta\mathrm{AUROC} \geq 0.10$ with a paired CI excluding zero. Both are met in all six arms (Table 27). The maximum rank correlation is 0.353 and the median write-minus-rule AUROC difference is 0.181. Write AUROC is 0.874–0.959, against 0.983–1.000 for the MECH-based reference. The AUROC target is any observation change, while the correlation compares two score rankings; neither measures reward-loss magnitude or establishes a performance ceiling.

We recovered and validated the original admission order and trajectory identifiers while preparing the null control. Rerun with the original generator body and seed $7919\,\mathrm{jid} + 104729\,v$ for $v = 0, \ldots, 11$, the new $14{,}796$-draw archive reproduces all six historical counts, BREAK rates, Spearman correlations, three AUROCs, write-minus-rule differences and their paired intervals to within 10^{-12}, and the median historical difference is still $+0.181$. The agreement is aggregate and comes from new executions; no original variant-level file was recovered. A separate tool-only generator adds another $14{,}796$ draws on the same $1{,}233$ trajectories, with every trajectory checked and no harness failures. Whenever a permutation repeats, including across the two generators, its observation changes, state hashes and rewards are identical.

Tool-only draws hold every response slot fixed and keep the original seed. At each of the original $k = \max(1, \mathrm{round}(s(m-1)))$ swaps, the first tool rank is drawn uniformly from $0, \ldots, m-2$ and the second is that rank plus two or three, capped at $m-1$. Because this generator changes the distribution of actual inversions and displacement as well as response movement, the comparison does not causally isolate response movement at identical perturbation magnitude.

I.4 Simple comparators and within-trajectory ranking

The supplementary analysis adds typed and either-write inversion counts, total tool and action inversions, length, displacement and moved-response counts. Every predictor score is available on all τ-bench variants and is computed on the same eligible trajectories. Besides pooled AUROC for $\mathrm{BREAK} > 0$, we compare variants within each trajectory by their BREAK counts: outcome ties are omitted, score ties receive 0.5, and trajectory concordances are macro-averaged. Constant-outcome trajectories are reported as ineligible.

We resample tasks $20{,}000$ times at seed 0, retaining all eligible trajectories of each sampled task, and compute paired score differences within the same draw. Results use both original draw weights and one weight per unique within-trajectory permutation. The marginal intervals are exploratory and carry no new multiple-comparison verdict. Since length is constant within a trajectory, its concordance is 0.5.

Table 27: Historical-generator severity scores on τ-bench. AUROC predicts $\mathbf{1}[\mathrm{BREAK} > 0]$, where BREAK counts changed identity-matched observations in the variant's actual execution. $V_{\texttt{mech}}$ is a replay-informed reference; $V_{\texttt{write}}$ uses baseline execution and per-call state access without a pairwise sweep. ΔAUROC is $V_{\texttt{write}}$ over $V_{\texttt{rule}}$, paired and task-clustered.

			AUROC			
arm	**var.**	$\rho(V_{\texttt{rule}}, V_{\texttt{mech}})$	$V_{\texttt{rule}}$	$V_{\texttt{write}}$	$V_{\texttt{mech}}$	**ΔAUROC (95% CI)**
gpt-4o retail r=1	2,964	+0.339	0.714	0.920	0.988	+0.206 [+0.157, +0.249]
gpt-4o retail r=0	1,896	+0.339	0.708	0.874	0.985	+0.166 [+0.106, +0.227]
sonnet retail r=1	6,720	+0.196	0.725	0.910	0.998	+0.184 [+0.105, +0.251]
gpt-4o airline r=1	564	+0.353	0.782	0.959	0.991	+0.177 [+0.084, +0.297]
gpt-4o airline r=0	924	+0.260	0.651	0.874	0.983	+0.224 [+0.137, +0.313]
sonnet airline r=1	1,728	+0.217	0.804	0.955	1.000	+0.150 [+0.081, +0.211]

Table 28: Admission and severity denominators. Each eligible trajectory receives 12 draws, which need not produce 12 distinct permutations.

Arm	Admitted	$m < 2$	$m = 2$	$m \geq 3$	Draws
gpt-4o retail r=1	251	3	1	247	2,964
gpt-4o retail r=0	174	6	10	158	1,896
sonnet retail r=1	565	0	5	560	6,720
gpt-4o airline r=1	84	17	20	47	564
gpt-4o airline r=0	102	18	7	77	924
sonnet airline r=1	177	4	29	144	1,728
Total	1,353	48	72	1,233	14,796

Table 29: The same draw budget does not imply the same intervention exposure. Unique permutations are counted within trajectory. Non-monotone counts refer to trajectories whose tool-inversion count decreases at least once as the nominal generator setting increases.

Quantity	Historical	Tool-only
Draws	14,796	14,796
Unique permutations	13,624	12,917
Repeated draws	1,172	1,879
Baseline-identical draws	462	735
Draws moving a response	11,461	0
Trajectories moving a response	1,233	0
Non-monotone trajectories	1,232	1,233

A single endpoint swap at action positions $i < j$ reverses $2(j - i) - 1$ action pairs, whereas its tool-pair count uses tool-rank distance. Neither identity makes the cumulative generator setting a measure of actual severity. Comparing against total inversions assesses predictive ranking, without holding inversion magnitude fixed or identifying a causal effect of edge information.

Table 30: Historical draws: same-trajectory ranking of BREAK counts. I is total tool inversions; T and W are typed and later-write inversion scores. Eligible/all excludes constant-outcome trajectories. Paired 95% intervals compare W and T with I on the same support.

Arm	Eligible/all	I/T/W	W−I (95%)	T−I (95%)
gpt-4o retail r=1	135/247	0.706/0.792/0.933	+0.228 [+0.182, +0.273]	+0.087 [+0.060, +0.113]
gpt-4o retail r=0	95/158	0.747/0.811/0.870	+0.123 [+0.070, +0.174]	+0.063 [+0.029, +0.096]
sonnet retail r=1	123/560	0.761/0.815/0.906	+0.145 [+0.090, +0.188]	+0.053 [+0.010, +0.091]
gpt-4o airline r=1	15/47	0.773/0.781/0.914	+0.140 [+0.048, +0.231]	+0.008 [−0.112, +0.092]
gpt-4o airline r=0	47/77	0.743/0.739/0.904	+0.160 [+0.100, +0.220]	−0.004 [−0.061, +0.049]
sonnet airline r=1	14/144	0.764/0.813/0.907	+0.143 [+0.034, +0.251]	+0.049 [−0.048, +0.193]

For historical draws, the comparison is defined for $429/1{,}233$ trajectories; the other 804 have constant BREAK counts, all at zero. With the original draw weights, W−I concordance differences range from $+0.123$ to $+0.228$, with positive paired intervals in all six arms. T−I intervals are positive in the three retail arms only, and W−T intervals are positive in $5/6$ arms. When each unique within-trajectory permutation receives one weight, these counts remain $6/6$, $3/6$ and $5/6$, respectively. The archive reports all points, marginal intervals and undefined fractions.

Table 31: Original response-fixed tool-only draws before the argument-aliasing exclusion: same-trajectory ranking of BREAK counts. I is total tool inversions; T and W are typed and later-write inversion scores. Eligible/all excludes constant-outcome trajectories. Paired 95% intervals compare W and T with I on the same support.

Arm	Eligible/all	I/T/W	W−I (95%)	T−I (95%)
gpt-4o retail r=1	211/247	0.694/0.828/0.895	$+0.201$ $[+0.171, +0.231]$	$+0.134$ $[+0.109, +0.158]$
gpt-4o retail r=0	134/158	0.697/0.795/0.858	$+0.161$ $[+0.122, +0.199]$	$+0.098$ $[+0.073, +0.123]$
sonnet retail r=1	475/560	0.702/0.811/0.880	$+0.178$ $[+0.153, +0.203]$	$+0.109$ $[+0.090, +0.129]$
gpt-4o airline r=1	24/47	0.813/0.812/0.864	$+0.051$ $[-0.025, +0.150]$	-0.002 $[-0.055, +0.047]$
gpt-4o airline r=0	65/77	0.773/0.782/0.865	$+0.093$ $[+0.040, +0.150]$	$+0.009$ $[-0.027, +0.043]$
sonnet airline r=1	48/144	0.807/0.817/0.871	$+0.063$ $[-0.001, +0.128]$	$+0.010$ $[-0.039, +0.055]$

In the response-fixed tool-only draws, $957/1{,}233$ trajectories are eligible; the other 276 have constant BREAK counts, all at zero. With the original draw weights, W−I differences range from $+0.051$ to $+0.201$, but the interval excludes zero in only $4/6$ arms: both successful-airline arms are inconclusive (Table 31). W−T intervals are positive in $4/6$ arms. Under the same unique-permutation weighting, intervals are positive in $5/6$ arms for W−I, $3/6$ for T−I, and $4/6$ for W−T. The additional positive W−I interval is sonnet airline $r = 1$; gpt-4o airline $r = 1$ remains inconclusive. The archive again reports all points, marginal intervals and undefined fractions.

Table 32: Pooled AUROC for any observation change on the new tool-only intervention. Simple magnitude/length comparators and typed/write scores use the same variants; this target differs from within-trajectory BREAK ranking.

Arm	Tool inv.	Action inv.	Tool length	Displacement	Typed	Write
gpt-4o retail r=1	0.646	0.697	0.529	0.688	0.781	0.903
gpt-4o retail r=0	0.665	0.688	0.570	0.682	0.771	0.883
sonnet retail r=1	0.618	0.662	0.499	0.656	0.755	0.880
gpt-4o airline r=1	0.698	0.694	0.578	0.695	0.837	0.913
gpt-4o airline r=0	0.546	0.588	0.398	0.581	0.700	0.811
sonnet airline r=1	0.761	0.788	0.722	0.785	0.853	0.937

The full archive additionally reports action length, tool displacement, moved-response count, either-write and replay-informed MECH, both weighting schemes, pooled paired differences and the within-trajectory write-minus-typed contrast. The MECH score is included as a replay-informed reference and is not a performance ceiling.

I.5 Two mechanically selected examples

For each specified discordance condition we take the first historical draw that satisfies it, scanning in a fixed arm, trajectory-ID and variant order instead of selecting for apparent semantic seriousness. Both examples come from gpt-4o retail $r = 1$.

An observation-only change missed by the rule. Trajectory 112, task 83, variant 0 (seed 886,928) has $V_{\text{rule}} = V_{\text{typed}} = 0$, $V_{\text{write}} = 1$ and BREAK $= 1$. Swapping original actions 10 and 13 moves an invalid `return_delivered_order_items` call after a successful return at action 12. The unchanged arguments are order `#W9571698`, item `6065192424` and payment method `credit_card_1565124`; the successful return uses `gift_card_7250692`. The invalid call's observation changes from a payment-method error to a non-delivered-order error. One response identity moves, but final state and reward do not change, so a MECH-positive example need not represent additional task failure.

An inversion without a replay change. Trajectory 0, task 0, variant 0 (seed 0) swaps actions 11 and 14: `get_product_details` for product `1656367028` and `exchange_delivered_order_items` for order `#W2378156`. The exchange retains new item IDs `7706410293`/`7747408585` and payment method `credit_card_9513926`. Here $V_{\text{rule}} = 3$ and $V_{\text{typed}} = V_{\text{write}} = 2$, yet all identity-matched observations, final state and reward are unchanged, which makes the draw a false alarm for the write score as well. The case archive preserves every original argument, permutation and observation/state difference for both examples. The two cases illustrate scope and are not a new performance test.

I.6 Post-hoc argument-integrity diagnostic

A final hash check found one exception to immutable replay inputs, in gpt-4o airline $r = 0$ (trajectory 76, task 0). Of the 1,305 admitted pair-contributing trajectories, 1,304 have action sequences that exactly match a released source sequence; the exception already contains an extra refund entry in action 15 after admission. Of the 1,233 severity trajectories, 1,232 retain their input digest through the new replay, while the same exception acquires further refund entries in actions 13 and 21. The simulator stores `payment_methods` by reference as reservation payment history and extends that list on cancellation, so resetting the database does not isolate a retained call object from this mutation.

A separate probe exactly reproduces the archived baseline and all 24 variants for this trajectory. When each action is deep-copied immediately before execution, the admitted input digest is preserved but $15/24$ variant records change. Because the probe starts from the admitted calls, it neither restores the raw pre-admission sequence nor relabels the frozen pair data. We therefore also run post-hoc sensitivities that exclude the entire trajectory (78 pairs). With 3,317 remaining pairs and 36 task clusters, the direct MECH MCC difference is $+.395$ $[+.258, +.505]$ (nominal eight-arm interval $[+.196, +.542]$), and the reduced-design write coefficient is 1.483 $[1.177, 1.876]$, with all 2,000 draws converging for each position model. In the tool-only severity comparison, within-trajectory write-minus-tool-inversion concordance is $+.096$ $[+.042, +.154]$ after exclusion and $+.093$ $[+.040, +.150]$ after replacing only this trajectory with the isolated-call probe. The unique-permutation results are also positive, and neither diagnostic changes the six-arm $4/6$ (draw) and $5/6$ (unique) positive-interval counts. The historical data and verdicts are left unchanged.

This exception limits an unconditional fixed-input interpretation of the historical harness. We have not relabelled the whole cohort with isolated calls, and matching start and end digests cannot on their own exclude transient mutation. The separate diagnostic archive records source comparisons, mutations, changed outcomes and all sensitivity intervals.

The main response-fixed estimate now excludes this trajectory: $64/76$ trajectories are eligible in gpt-4o airline $r = 0$, and the write-minus-tool-inversion concordance difference is $+0.096$ $[+0.042, +0.154]$. Across all six arms, the eligible total is $956/1{,}232$. The separately stored isolated-call replacement remains a diagnostic, starting from admitted inputs; it is not a reconstruction from raw pre-admission inputs or a full corrected-harness relabeling. Appendix J.1 connects these estimates to the historical populations and BFCL tag exclusions.

J Published extractors, inspection decisions and binding controls

These analyses were designed with the earlier replay outcomes known. Plans were frozen before the new computations, but the analyses are exploratory and do not supply an independent confirmatory holdout. Moving their results into the main text does not re-adjudicate any original gate.

J.1 Current populations and input-integrity evidence

The current extractor comparison excludes the known airline argument-aliasing trajectory and all known BFCL tag cases. The response-fixed ranking and inspection analyses exclude the airline trajectory. Table 33 separates these estimates from historical results. The latter remain the record of the original analyses, including the BFCL directional-hypothesis failure. Supporting human, PDDL, SWE-bench, position, effect-location and outcome-level results retain their stated populations.

Table 33: Pair populations. Common support is the subset on which the new extractors are all available within each arm; schema-ID is not applied to BFCL. The original held-out adjudication remains frozen.

Population	τ-bench	BFCL base	BFCL held-out	Total
Executed original audit	33,137	2,511	5,978	41,626
Original reported population	33,137	2,484	5,978	41,599
Current exclusions	33,059	2,484	5,930	41,473
Current extractor common support	33,059	2,373	5,692	41,124

The original audit executed $33{,}137$ transpositions in six τ-bench arms, including successful and failed baselines, and $8{,}489$ in BFCL multi-turn base and held-out. Removing the known BFCL-base artefact (27 pairs) left $41{,}599$ pairs in the original reported population. The current exploratory comparisons additionally exclude one airline trajectory (78 pairs) with argument aliasing and the held-out tag cases (48 pairs), leaving $41{,}473$ pairs. These exclusion-based estimates do not replace the original pre-registered adjudications.

The airline exception contributes 78 pairs. A simulator aliases retained payment arguments into database history, which a later cancellation extends. Admission already changes one retained call. Isolating arguments by deep copy changes $15/24$ variant records in the diagnostic replay, despite preserving the principal aggregate comparisons. Because the probe starts from admitted inputs, it neither restores raw pre-admission calls nor corrects the complete pair cohort. Appendix I.6 records the source comparison and the separate exclusion and isolated-call estimates.

BFCL's instance-name helper rewrites method-like strings inside quoted arguments. Fresh identifiers therefore change actual arguments in five known trajectories; two cases also change final state. This is not solely output logging noise. The same-order census found no additional null changes among $1{,}882$ checked trajectories, but repeatability and equal start/end argument hashes cannot exclude every transient mutation. The current estimates remove known exceptions; a full corrected-harness rerun from raw inputs remains unperformed. The frozen held-out common-support sensitivity gives typed/published-ID/write MCC $0.138/0.157/0.299$, compared with $0.144/0.161/0.311$ after exclusion. In the affected airline arm the corresponding frozen values are $0.209/0.343/0.579$, compared with $0.221/0.370/0.616$ after exclusion.

J.2 Published and schema-aware extractor comparison

We preserve the public substring and typed-ID function bodies from the frozen upstream implementation accompanying Sun & Kazakov (2026). Public substring recursively extracts argument values, removes non-alphanumeric characters, lowercases and tests output inclusion with minimum length four. Public typed-ID compares normalised candidates under `id`, `_id` and `_ids` keys and scalar returns; it does not compare user/order entity types. Its consumer search is restricted to top-level typed-ID keys and scalar/list values. We do not revise these rules. The paper-literal substring comparator instead implements whitespace-normalised, case-sensitive character inclusion and is reported separately. This public-code application does not reproduce the source paper's representation results or establish which code version produced every original estimate. Typed-ID's source domain is retail; airline and BFCL are domain transfers.

Our schema-ID matcher freezes a tool/path mapping to entity type, representation type and canonical value. It covers retail user, order, product, item and payment IDs, and airline user, reservation, payment, flight number and airport IDs. Strings match exactly without a minimum length; numbers retain typed decimal canonicalisation. Booleans, nulls and empty strings are not ID candidates. Dictionary keys are candidates only on paths declared to contain actual IDs, such as airport codes. Names, addresses, dates, counts and status values are outside this ID-only matcher. The mapping uses schema and static return-format knowledge, not transposition outcomes, write labels or a lookup of actual database IDs. Unrecognised structures are unavailable, rather than forced negatives. No matcher takes transitive closure or selects a unique producer. The `id_value_only` ablation uses identical candidates and normalisation, ignoring entity type alone.

Each of 1,882 original-order trajectory records was replayed twice to recover a common baseline cache. Extraction had no failures; existing predictors matched their frozen labels, and new predictors were stable between repeats. Raw observation differences occurred only in the known BFCL tag cases. The stored transposition MECH/STATE outcomes were not relabelled. BFCL positional arguments were bound to pinned function signatures where possible, with failed binding marked unavailable. The common-support availability is 100% for τ-bench and $95.5\%/96.0\%$ for BFCL.

Table 34: MECH MCC on current common support. R: token overlap; T: typed scalar; PS: public substring; PI: public ID; S: schema-ID; I: ID-value-only; W: write; L: paper-literal substring. BFCL held-out is tag-free.

Arm	R	T	PS	PI	S	I	W	L
gpt-4o retail $r = 1$	0.133	0.231	0.222	0.331	0.223	0.223	0.750	0.083
gpt-4o retail $r = 0$	0.141	0.235	0.230	0.281	0.200	0.200	0.664	0.086
sonnet retail $r = 1$	0.124	0.255	0.243	0.332	0.241	0.241	0.733	0.066
gpt-4o airline $r = 1$	0.129	0.188	0.182	0.316	0.147	0.147	0.571	0.125
gpt-4o airline $r = 0$	0.191	0.221	0.241	0.370	0.191	0.191	0.616	0.119
sonnet airline $r = 1$	0.132	0.292	0.285	0.364	0.195	0.195	0.620	0.078
BFCL base	0.046	0.134	0.062	0.171	–	–	0.261	-0.005
BFCL held-out	0.067	0.144	0.063	0.161	–	–	0.311	0.002

Table 35: Public ID paired MECH MCC differences. All four contrasts per arm share one planned Bonferroni family of 24 comparisons; these analyses are exploratory. Predictor abbreviations follow Table 34.

Arm	Contrast	Difference	95% interval	Adjusted interval
gpt-4o retail $r = 1$	PI−T	$+0.100$	$[+0.079, +0.123]$	$[+0.067, +0.138]$
gpt-4o retail $r = 1$	W−PI	$+0.419$	$[+0.367, +0.472]$	$[+0.337, +0.501]$
gpt-4o retail $r = 0$	PI−T	$+0.046$	$[+0.026, +0.069]$	$[+0.012, +0.081]$
gpt-4o retail $r = 0$	W−PI	$+0.383$	$[+0.319, +0.453]$	$[+0.286, +0.498]$
sonnet retail $r = 1$	PI−T	$+0.077$	$[+0.056, +0.099]$	$[+0.043, +0.111]$
sonnet retail $r = 1$	W−PI	$+0.401$	$[+0.358, +0.443]$	$[+0.332, +0.466]$
gpt-4o airline $r = 1$	PI−T	$+0.128$	$[+0.048, +0.200]$	$[-0.004, +0.251]$
gpt-4o airline $r = 1$	W−PI	$+0.255$	$[+0.141, +0.407]$	$[+0.078, +0.524]$
gpt-4o airline $r = 0$	PI−T	$+0.149$	$[+0.071, +0.232]$	$[+0.034, +0.277]$
gpt-4o airline $r = 0$	W−PI	$+0.246$	$[+0.096, +0.382]$	$[+0.014, +0.454]$
sonnet airline $r = 1$	PI−T	$+0.072$	$[-0.014, +0.151]$	$[-0.054, +0.190]$
sonnet airline $r = 1$	W−PI	$+0.256$	$[+0.124, +0.441]$	$[+0.062, +0.538]$

Table 36: Schema ID paired MECH MCC differences. All four contrasts per arm share one planned Bonferroni family of 24 comparisons; these analyses are exploratory. Predictor abbreviations follow Table 34.

Arm	Contrast	Difference	95% interval	Adjusted interval
gpt-4o retail $r = 1$	S−T	-0.009	$[-0.019, +0.001]$	$[-0.026, +0.007]$
gpt-4o retail $r = 1$	W−S	$+0.527$	$[+0.475, +0.581]$	$[+0.444, +0.609]$
gpt-4o retail $r = 0$	S−T	-0.035	$[-0.058, -0.011]$	$[-0.071, -0.002]$
gpt-4o retail $r = 0$	W−S	$+0.464$	$[+0.408, +0.529]$	$[+0.380, +0.573]$
sonnet retail $r = 1$	S−T	-0.014	$[-0.027, -0.002]$	$[-0.035, +0.004]$
sonnet retail $r = 1$	W−S	$+0.492$	$[+0.446, +0.538]$	$[+0.420, +0.565]$
gpt-4o airline $r = 1$	S−T	-0.042	$[-0.094, +0.001]$	$[-0.128, +0.004]$
gpt-4o airline $r = 1$	W−S	$+0.424$	$[+0.317, +0.560]$	$[+0.263, +0.660]$
gpt-4o airline $r = 0$	S−T	-0.030	$[-0.056, -0.009]$	$[-0.075, -0.002]$
gpt-4o airline $r = 0$	W−S	$+0.425$	$[+0.292, +0.532]$	$[+0.209, +0.580]$
sonnet airline $r = 1$	S−T	-0.097	$[-0.173, -0.007]$	$[-0.215, +0.009]$
sonnet airline $r = 1$	W−S	$+0.425$	$[+0.340, +0.562]$	$[+0.305, +0.634]$

The four planned MECH contrasts in each of six τ-bench arms use $50{,}000$ paired task-bootstrap draws, marginal 95% intervals and a Bonferroni adjustment over 24 comparisons. Schema-ID minus

Table 37: Supplementary STATE MCC on the same current common support. W is not uniformly superior. Intervals and undefined fractions are retained in the result archive.

Arm	Typed scalar	Public ID	Schema ID	Write
gpt-4o retail $r = 1$	0.076	0.121	0.078	0.111
gpt-4o retail $r = 0$	0.136	0.138	0.099	0.113
sonnet retail $r = 1$	0.128	0.193	0.133	0.212
gpt-4o airline $r = 1$	0.005	0.065	0.012	0.209
gpt-4o airline $r = 0$	0.060	0.136	0.027	0.210
sonnet airline $r = 1$	-0.005	0.032	0.002	0.114
BFCL base	0.087	0.106	–	0.426
BFCL held-out	0.092	0.101	–	0.442

typed scalar equality is negative in all six point estimates, with adjusted intervals negative in two. Schema-ID and ID-value-only edges are identical in every arm: no edge is removed by entity-type discrimination. This is lack of an informative ablation contrast, not evidence that semantic typing is generally unhelpful. The public typed-ID and schema matchers also differ in candidate coverage and normalisation, so their performance difference is not the effect of entity typing alone.

Schema-ID adds 33 airline pairs absent from typed scalar matching through actual airport dictionary keys; four are MECH-positive and none STATE-positive. It also excludes non-ID matches, some with observation changes. The fixed case-selection rule yielded 40 category entries covering 30 distinct pairs; empty strata and shortfalls are retained. These cases describe extractor disagreements rather than independent human provenance annotations. Equal-task-weighted point estimates preserve write $>$ public ID $>$ typed scalar equality in all eight arms. With known cases included, the public-ID minus typed and write-minus-ID MECH differences are respectively $+.134$ $[+.049, +.224]$ and $+.236$ $[+.094, +.364]$ in airline $r = 0$; in frozen BFCL held-out they are $+.019$ $[-.019, +.055]$ and $+.142$ $[+.087, +.201]$. These sensitivity intervals are marginal 95%.

In the current BFCL common-support MECH comparison, the write-minus-published-ID MCC differences are $+0.090$ $[+0.007, +0.179]$ for base and $+0.150$ $[+0.095, +0.209]$ for held-out. These are supplementary marginal 95% intervals, outside the 24-comparison τ-bench family.

STATE does not support a uniform write advantage. In gpt-4o retail $r = 0/1$, write MCC is $.113/.111$ and public-ID MCC is $.138/.121$. BFCL's public-ID MECH gain over typed scalar equality is also uncertain: base $+.037$ $[-.024, +.091]$ and held-out $+.017$ $[-.022, +.054]$. All confusion matrices, per-method availability, precision, recall, STATE, task support and undefined-bootstrap fractions are retained in `agent_atlas/results/extractor_comparison_20260925/pair_metrics.json`.

J.3 Budgeted inspection details

The inspection analysis (A) reuses the response-fixed archive without new environment replay or agent generation. It includes 1,232 trajectories, 164 distinct tasks and 12,416 non-identity unique variants, with no technical missingness. Observation yield is defined over tool observations; response positions are fixed. For each score, select the top $k = \lceil qn \rceil$ variants per trajectory, assigning each tied boundary variant its equal expected selection weight. Average trajectory yield within each task and then average tasks. Paired task resampling uses 20,000 draws. Tasks overlap across arms, so arm counts cannot be summed.

At nominal $q = .20$, the actual task-macro selected share is 25.73%. Write and typed selection weights differ by 41.94% [39.97, 43.93]; this is a fractional-set difference, not expected Jaccard similarity. Observation yield increases by 4.37 [2.72, 6.08] percentage points. The registered exploratory verdict is `WRITE_HIGHER_EXPLORATORY`; it does not establish an effect of at least five percentage points. STATE yield decreases by 0.98 [0.25, 1.86] points in the supplementary marginal comparison. Success loss is defined on 998 successful baselines from 148 tasks, with only 17 positive tasks, below the required 20; intervals and superiority verdicts are withheld.

At nominal budgets 10%/30%, write observation yield is 67.04%/60.06% and typed yield 61.44%/56.03%, with differences $+5.60$ [3.75, 7.59] and $+4.03$ [2.64, 5.46] points. Arm-specific

Table 38: Inspection scores at nominal $q = .20$. Entries are yield percentages; intervals are marginal 95%. The reference requires replay outcomes and is neither deployable nor a mathematical ceiling.

Score	Observation	Final state	Success loss
Write	63.25 [58.43, 67.95]	4.25 [2.58, 6.16]	4.61
Typed scalar	58.87 [54.20, 63.44]	5.24 [3.18, 7.57]	5.41
Token overlap	57.07 [52.41, 61.60]	4.72 [2.86, 6.83]	5.04
Total tool inversions	50.82 [46.34, 55.19]	4.78 [2.88, 6.93]	4.84
Uniform random	36.86 [33.50, 40.22]	3.71 [2.27, 5.33]	3.71
Replay-informed reference	66.28 [61.27, 71.13]	5.49 [3.34, 7.95]	5.87

20% differences are $+7.64$ $[.89, 14.73]$ (gpt-4o airline $r = 0$), $+5.64$ $[-1.80, 14.25]$ (gpt-4o airline $r = 1$), $+3.00$ $[-.08, 6.21]$ (gpt-4o retail $r = 0$), $+4.01$ $[1.11, 7.10]$ (gpt-4o retail $r = 1$), $+5.31$ $[-.10, 11.17]$ (sonnet airline), and $+4.23$ $[2.10, 6.40]$ (sonnet retail). They do not establish all-arm superiority.

The five planned sensitivities give positive observation-yield intervals: including identity permutations, $+4.38$ $[2.75, 6.08]$ points; historical unique permutations, $+5.27$ $[3.44, 7.31]$; tool-only draw weights, $+4.38$ $[2.77, 6.06]$; including the known aliasing trajectory, $+4.34$ $[2.70, 6.04]$; and replacing it with the isolated-call probe, $+4.34$ $[2.70, 6.04]$. The last probe starts from admitted inputs. These settings neither equalise state-access costs nor extend the comparison to public typed-ID scores.

J.4 Public-judge transfer and unexecuted original plans

The original plan B sought to apply OTAP's actual evaluator, but the examined public sources did not establish its native extractor and executable implementation. We did not substitute an invented implementation; its status is `NOT_RUN_IMPLEMENTATION_UNRESOLVED`. Plan C sought independent argument provenance in 998 successful natural-log candidates. Literal arguments and observations did not establish the source of every argument leaf, and inferring producers by the value equality being tested would be circular. Its status is `NOT_RUN_PROVENANCE_UNAVAILABLE`. These are feasibility limits, not negative performance results.

The alternative B2 uses the frozen public Plan-RewardBench prompt and formatter with Llama 3.1 8B and Qwen2.5 7B, both Q4_K_M, on local Ollama/Metal. It is a transfer to our inputs and labels, not a reproduction of the original dataset or leaderboard. Among 998 candidate trajectories, 715 passed pre-sampling conditions, 268 exceeded a 28,000-byte input cap for all pairs, and 15 lacked structural contrast. The eligible set covered 105 tasks; an outcome-independent hash order selected 20 retail and 20 airline tasks. The length exclusion limits generalisation.

We select minimum/maximum structural-score permutations per trajectory, using inversion counts over the verified public substring edges. Judges see the reordered calls and recorded baseline observations, not replay outcomes. Each of 71 pairs receives two presentation orders for two models, 284 requests. Ground-truth preferences are separately defined by fewer structural inversions and fewer tool-observation changes in isolated replay. All 40 selected baselines agreed across repeated runs; 193 isolated replays were completed. Among the selected pairs, 32 preferences agree, 39 observation counts tie, and none reverse. Selected variants have zero final-state changes and success losses.

The initial 192-token output limit produced six saved responses, five truncated. We preserved them and reran all 284 requests at a 1,024-token limit without selectively mixing responses. Under the pre-specified strict JSON parser, $115/284$ requests (40.49%) fail technical/format criteria. A pair is excluded from both models if any of its four responses is missing. Only four non-tied pairs from three tasks remain, so the primary verdict is `TECHNICALLY_INCOMPLETE`. Technical failures are not scored as incorrect answers. Llama/Qwen accuracies are $58.33\%/66.67\%$ under both labels; the relative difference is 8.33 points and its change across labels is zero, with intervals withheld for insufficient task support.

After inspecting the initial format failures, we declared an original-parser diagnostic, applying the unchanged public `judge_pair` function to all saved responses without new model calls. It recovers

32 pairs from 21 tasks with no parsing failures; the diagnostic verdict is CHANGE_UNCERTAIN. Llama/Qwen accuracies are $51.19\%/60.71\%$ under both labels. The Qwen-minus-Llama difference is 9.52 $[0.00, 20.24]$ points for each label, and the difference of differences is 0.00 $[0.00, 0.00]$, using $20{,}000$ paired task-bootstrap draws and Bonferroni 98.333% intervals for three contrasts. The zero interval follows algebraically because every supported pair has the same label under both criteria. It does not establish population equivalence. The public parser reads markers from the full response, including reasoning; preserving that behaviour does not certify semantic correctness. It changes no winner among strictly valid responses and does not supersede the primary verdict.

The 39 outcome ties are cases in which structural preference alone cannot justify an observation-outcome preference. They were not converted into new correct labels after seeing the results. STATE and success-loss comparisons lack support. No evaluator-selection or functional-evaluation improvement is established by this transfer.

J.5 Explicit-binding workflow controls

The alternative C2 constructs five workflow families with 20 instances each, using unchanged τ-bench tools and initial databases. Before execution, every argument leaf is assigned to an initial field or a producer/output path. Each instance is run under recorded-value supply and online return-value binding, each in original and reordered execution: 400 runs in total. The two original-order baselines agree in arguments, observations and final state. Completion failure means an unavailable input or tool error prevents the planned calls from completing without error. It is neither a benchmark reward-loss label nor an estimate of natural-agent failure prevalence.

Table 39: Constructed binding controls. Each family has 20 instances. Reordered failure rates are descriptive; all original-order failures are zero. Interaction is the change in the order effect under online binding relative to recorded-value supply, in percentage points.

Family	Recorded-value failure	Online-binding failure	Interaction
Information only	0%	100%	100
ID initially available	0%	0%	0
State prerequisite only	100%	100%	0
Information and state	100%	100%	0
Independent calls	0%	0%	0

For information-only workflows, recorded values allow a consumer to execute before its producer; online binding stops because the input is unavailable. An initially supplied ID requires no such producer. For state prerequisites, an order's address cannot be updated after cancellation even when the ID is known. For information-and-state workflows, a reservation does not exist before creation: recorded-value replay returns a tool error, while online binding stops at the missing input. Independent calls complete under both policies. These outcomes follow the specified tool semantics and bindings and illustrate the intended mechanism. They do not estimate adaptive replanning, natural-log frequency or general agent reliability. New airline reservation IDs are also predictable strings in this simulator. C2 uses a different constructed population from B2 and does not identify a mediator of judge performance.

J.6 Frozen plans and verification records

The extractor plan is commit `804cde4`, with implementation and schema mapping frozen at `d45248ac`; its result directory is `agent_atlas/results/extractor_comparison_20260925/`. The inspection plan/implementation commits are `958ad69`/`6b0cffc`, with outputs in `agent_atlas/results/severity_decisions_20260925/`. B2/C2 use plan `d3687e1`, input/implementation freeze `b60a1fc`, output-limit amendment `553db14`, and public-parser diagnostic declaration `a50150b`; outputs are in `agent_atlas/results/severity_alternatives_20260925/`.

Extractor verification compared MCC with binary Pearson correlation, checked one-to-one pair joins and reproduced all 258 reported estimates and intervals using integer accumulation with the same bootstrap draws. Inspection verification independently reproduced task-resampling estimates within

10^{-12} and checked all six settings against their source archives. B2 verification checked prompts, request/model correspondence, argument hashes, parser support and task-weighted arithmetic. C2 verification independently checked all 700 argument instances and 400 executions. Frozen inputs and initial interrupted responses remain preserved. These checks establish the reported computations' provenance without adding independent outcome validation.