IMPACT: MEASURING CONDITIONAL WORKFLOW RESPONSES TO SINGLE-SPAN MODEL SUBSTITUTION IN STATEFUL LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Selectively replacing individual language-model invocations with smaller models can reduce inference cost in LLM agents. Yet an agent call is not an independent request: its output can alter subsequent actions, state transitions, observations, and computation. Local model quality and invocation cost therefore provide only a partial view of a substitution, whose effects may propagate through the rest of the workflow. We present IMPACT, a workload-agnostic intervention methodology for measuring conditional workflow responses to single-span model substitution. Starting from a common restored pre-invocation state, IMPACT uses a four-condition protocol to separate replay fidelity, continuation variation, baseline-model resampling, and candidate substitution. It estimates substitution effects from repeated matched baseline continuations, treating the recorded trajectory as one realization of the baseline workflow. IMPACT characterizes each intervention through local behavior, downstream propagation, final task quality and collateral state outcomes, and resource use, and relates these responses to conditions available before the target invocation. We instantiate IMPACT across six cohorts spanning AppWorld, LoCoMo, and SWE-bench, four agent implementations, and 17,100 live continuations. Across these populations, baseline executions vary, local agreement and final quality often diverge, and resource savings do not consistently align with task outcomes. The resulting conditional response atlas summarizes how substitution effects vary with execution context and population and provides evidence for offloading decisions under explicit quality, risk, and resource objectives.

1 INTRODUCTION

LLM agents invoke language models to choose tools, construct arguments, interpret observations, and revise plans. These calls occur inside a stateful execution, where one output can change later actions, state transitions, and observations. Assigning selected functions to smaller models is attractive for reducing inference cost (Belcak et al., 2025), but local model quality and invocation cost do not determine the resulting workflow. We therefore ask: *how does a stateful agent workflow respond when one model invocation is replaced by another?*

A local substitution can affect the workflow in qualitatively different ways. Different query arguments may still retrieve the required information; an invalid API choice may trigger diagnosis and retry; a locally valid action may instead propagate an error through later state transitions. Even when the task eventually succeeds, the execution path and resource use may differ substantially. Evaluating a substitution therefore requires following its consequences through the live continuation.

The continuation itself is also stochastic. Even without substitution, regenerating the target with the original model can change the suffix, and fixing the recorded target output still leaves variation in later execution. A recorded trajectory therefore captures only one realization of the baseline workflow (Bjarnason et al., 2026). This makes model substitution a controlled-intervention problem: the effect of changing one invocation must be separated from the variability that occurs under the baseline model itself.

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.

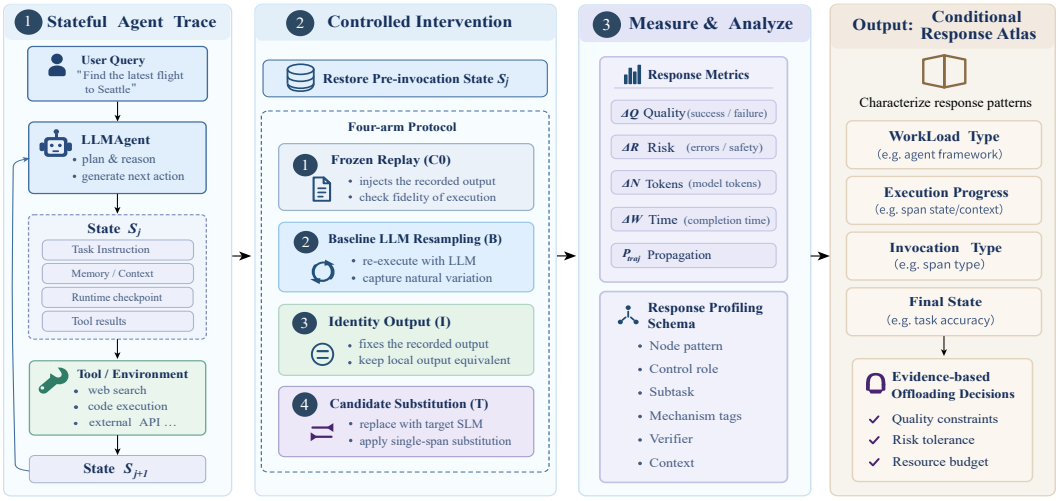


Figure 1: IMPACT has three measurement phases; the conditional response atlas is their output. $C0$ audits replay fidelity before the three live arms. Population context, pre-decision conditions, and retrospective diagnostics remain distinct from response outcomes.

IMPACT (*Intervention Measurement and Profiling of Agent-Call Trajectories*) addresses these measurement problems with an end-to-end intervention pipeline. Phase 1 records validated physical LM invocations and restorable execution state. Phase 2 executes matched $C0/I/B/T_{\text{span}}$ conditions from a common pre-invocation state. Phase 3 measures local, downstream, final, and resource responses and relates them to execution conditions and retrospective mechanisms. Figure 1 summarizes the pipeline.

Research questions. We organize the study around three questions:

1. **RQ1: Controlled measurement.** How can the workflow effect of a single model substitution be identified under stochastic agent continuation?
2. **RQ2: Workflow response.** How does a single-span substitution change local validity, downstream execution, final quality, collateral risk, token use, and completion time?
3. **RQ3: Conditional structure.** Which pre-decision execution conditions are associated with distinct workflow-response regimes, and how stable are these associations across repeated measurements and populations?

Contributions. IMPACT provides a workload-agnostic intervention methodology that connects trace collection and replayable execution state to the four-condition $C0/I/B/T_{\text{span}}$ protocol. This decomposition separates replay validity, fixed-output continuation variation, baseline-model regeneration, and candidate substitution as distinct sources of workflow variation. It introduces a baseline-variability-aware comparison in which substitution effects are estimated from repeated matched B continuations, with $C0$ auditing replay fidelity and I exposing continuation variation. It also characterizes substitution as a conditional workflow response that keeps local validity, downstream propagation, final quality, collateral risk, token use, and completion time as distinct response dimensions. The resulting population-conditioned response atlas organizes this evidence for offloading decisions under declared objectives.

Key findings. Across six cohorts and 17,100 live continuations spanning transactional tool use, long-horizon memory, and software engineering, repeated baseline executions can diverge from the recorded trajectory. Local agreement, final task quality, and resource response also separate: changed local actions can still lead to successful completion, while downstream continuation can amplify or erase target-level savings. These responses vary with execution context and population, and coarse span labels do not consistently identify the same regimes across agents and workloads.

2 CONDITIONAL WORKFLOW RESPONSE AS EVIDENCE FOR OFFLOADING DECISIONS

Model substitution changes one physical LM invocation s_j at restored pre-invocation state S_j. Let H denote the fixed agent implementation, including its parser and continuation policy. Baseline arm B regenerates the target with deployment D_L; treatment T_{span} uses candidate deployment D_m only at s_j. Both then follow the same baseline continuation policy. Repeated matched executions estimate the treatment-versus-control contrast (Rubin, 1974).

2.1 CONDITIONAL WORKFLOW RESPONSE

Repeated execution from a matched state induces a response distribution

$$\mathcal{P}_a(\cdot \mid S_j, H, D_L, D_m), \qquad a \in \{B, T_{\text{span}}\}, \tag{1}$$

because neither target generation nor the live suffix is deterministic. For a declared profile x, IMPACT summarizes the response as

$$\mathcal{R}(x, D_m) = (\Delta Q, \Delta R, \mathcal{P}_{\text{traj}}, \Delta N, \Delta W), \tag{2}$$

where ΔN and ΔW are treatment-minus-baseline model-token use and post-intervention completion time, and $\mathcal{P}_{\text{traj}}$ retains propagation diagnostics. With strict success $S_a = \mathbb{I}[F_a = 1 \wedge K_a = 0]$, where F_a is task success and K_a a prohibited state change,

$$\Delta Q = \mathbb{E}[S_{T_{\text{span}}} \mid x, D_m] - \mathbb{E}[S_B \mid x], \quad \Delta R = P(K_{T_{\text{span}}} = 1 \mid x, D_m) - P(K_B = 1 \mid x). \tag{3}$$

Profiles are population- and deployment-conditioned: repeated trials at one checkpoint estimate local variation, not coverage over unseen tasks. Conditional effect estimation motivates separating subgroup discovery from estimation (Athey & Imbens, 2016); IMPACT therefore reports interpretable response strata.

2.2 QUALITY, RISK, RESOURCE, AND EVIDENCE

For a declared cost C, define $\text{NetSaving}(x, D_m) = \mathbb{E}[C_B \mid x] - \mathbb{E}[C_{T_{\text{span}}} \mid x, D_m]$. A population supports offloading only when quality meets a chosen non-inferiority margin ϵ, treatment collateral risk stays within budget δ, and net saving is positive. We report ΔR as the substitution-induced risk change, while the decision criterion uses the absolute treatment-risk budget δ. Finite-sample support requires

$$\text{LCB}(\Delta Q) \geq -\epsilon, \qquad \text{UCB}(P(K_{T_{\text{span}}} = 1 \mid x, D_m)) \leq \delta, \qquad \text{LCB}(\text{NetSaving}) > 0. \tag{4}$$

These bounds concern a conditioned population and require replay validity, pre-specified inference, and profile support. Tokens and time remain separate resource objectives. An output is *workflow-compatible* when its continuation stays within declared tolerances, even if it does not reproduce the native trajectory. Only pre-decision information may support selection; retrospective descriptors explain realized responses. Appendix E gives inference details.

2.3 DOWNSTREAM COMPENSATION

For additive resources,

$$C_a = C_a^{\text{target}} + C_a^{\text{cont}} + C_a^{\text{overhead}}, \tag{5}$$

so

$$\text{NetSaving} = \underbrace{\mathbb{E}[C_B^{\text{target}} - C_{T_{\text{span}}}^{\text{target}}]}_{\text{DirectSaving}} - \underbrace{\mathbb{E}[C_{T_{\text{span}}}^{\text{cont}} - C_B^{\text{cont}}]}_{\text{CompensationCost}} - \Delta O. \tag{6}$$

Here $\Delta O = \mathbb{E}[C_{T_{\text{span}}}^{\text{overhead}} - C_B^{\text{overhead}}]$ is treatment-minus-baseline intervention-specific overhead. The reported cohorts use native recovery only, with no added verification, fallback, or rollback policy; resource accounting is detailed in Appendix E. Continuation response is signed: substitution can lengthen or shorten the suffix, but extra work alone does not imply repair and fast failure is not useful saving.

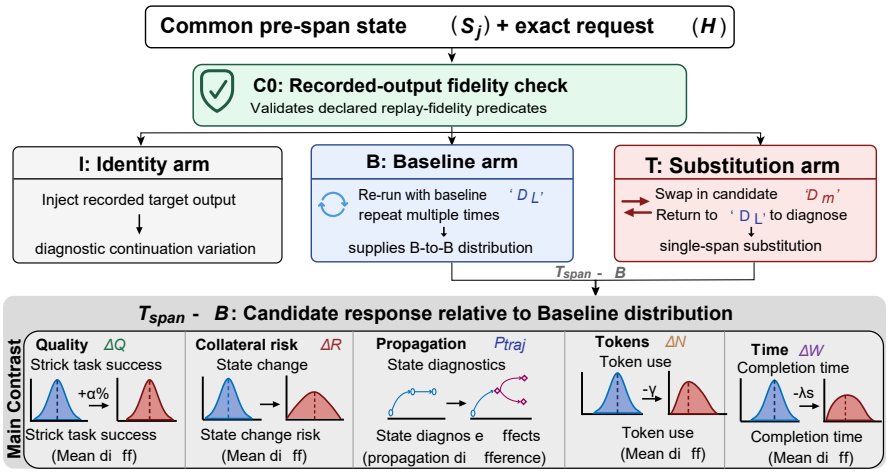


Figure 2: Four-condition measurement protocol. $C0$ passes its declared replay-fidelity audit before the live arms. I diagnoses continuation variation with the target fixed; repeated B runs supply the baseline workflow distribution; $T_{\text{span}} - B$ measures the single-span substitution response under the same continuation policy.

3 Trace Collection and Span Construction

Phase 1 records what the agent actually executes before assigning analytical workflow labels. Workload adapters expose restorable state, action/tool interfaces, evaluator hooks, and side-effect semantics; agent adapters identify physical LM invocations and continuation points.

Physical span and raw trace. A *span* is one physical LM invocation. Its raw record contains the exact request and output, tool/action events, checkpoint identifiers, timestamps, evaluator provenance, and restoration/continuation handles. Control roles, mechanism tags, and response profiles are derived later in Phase 3. Appendices A–B specify adapter and record interfaces.

Eligibility and validation. Successful native traces supply candidate points. Eligibility requires reconstructing the request and pre-state and evaluating the live continuation; native success alone does not establish replay fidelity. Parse failures after valid restoration remain outcomes. Risky but measurable spans remain eligible. Appendix C specifies replay predicates.

4 Intervention and Measurement Protocol

Phase 2 restores independent copies of the same pre-span state while fixing the request, schemas, parser, continuation policy, and budgets. In the four-condition $C0/I/B/T_{\text{span}}$ protocol, $C0$ is a pre-measurement replay-fidelity audit; I, B, and T_{span} are the live continuation arms. $C0$ injects the recorded target output and checks the declared request/state/action/evaluator predicates. All six cohorts passed C0; a pass certifies these predicates, not environmental determinism.

4.1 Controls and Live Continuation

Recorded-output continuation I fixes the recorded target output and measures live continuation variability. Baseline-model resampling B regenerates the target and supplies the matched execution distribution. Single-span substitution T_{span} changes only the target model; all later calls return to the baseline deployment and respond to the resulting state. The primary contrast is $T_{\text{span}} - B$, with native recovery retained in both arms. Figure 2 summarizes the protocol; Appendix D gives the step-by-step specification.

Table 1: Identification logic: each condition or contrast isolates one source of workflow variation that would otherwise be conflated. Figure 2 shows the corresponding execution flow.

Condition / contrast	Source of variation isolated
$C0$	Replay-fidelity validity; pre-measurement gate
I	Live suffix variation with the target output fixed
B	Ordinary baseline workflow distribution
$B - I$	Added variation from baseline target regeneration
T_{span}	Candidate-substituted workflow
$T_{\text{span}} - B$	Single-span substitution response

This decomposition lets the primary $T_{\text{span}} - B$ contrast measure substitution against the baseline workflow distribution, while $B - I$ diagnoses the added effect of baseline-model resampling; neither contrast is identifiable from a single execution arm.

4.2 Response, Cost and Inference

Run IDs link target outputs, downstream actions, final outcomes, and resources. The cost window extends from target-call start to completion or termination. Native retries belong to continuation; intervention-specific verification, fallback, or rollback belongs to overhead. Failures retain consumed resources, and tokens and elapsed time are measured separately.

Arm order is randomized within checkpoint blocks and comparable runs are interleaved. Inference preserves task/trace dependence; repetitions are not independent tasks. We report uncertainty alongside point estimates (Agarwal et al., 2021). Appendix E gives accounting, resampling, and confirmation requirements.

5 Response Profiling and Experimental Setup

Phase 3 relates measured responses to execution conditions using a workload-agnostic schema.

5.1 Response Profiling Schema

For an analyzed span, IMPACT uses

$$\phi_s = (\text{node pattern}, \text{control role}, \text{subtask}, \text{mechanism tags}, \text{verifier}, \text{context}), \tag{7}$$

with task condition z_t, agent condition a_H, and deployment D_m kept separate. We distinguish pre-decision conditions, retrospective mechanism descriptors, and outcomes. Only pre-decision information may support prospective selection; retrospective descriptors explain realized response. Appendix H defines feature timing and the atlas inventory.

5.2 Workloads, Agents, and Cohorts

We evaluate transactional tool use with AppWorld (Trivedi et al., 2024), long-horizon memory with LoCoMo/LangGraph/LangMem (Maharana et al., 2024), and software engineering with SWE-bench/mini-SWE-agent (Jimenez et al., 2024; Yang et al., 2024; SWE-agent, 2025). AppWorld uses SimplifiedFunctionCallingAgent and Tau2-LLMAgent, giving four agent implementations overall. Appendix A details adapters and evaluators.

The baseline endpoint is Qwen3.8-27B and the candidate is Qwen/Qwen3.5-9B. All live arms use native recovery only. Six cohorts yield 17,100 live continuations; SWE-bench contributes 134 intervention points over four Django instances with ten repeats per arm. Appendix F gives the full cohort inventory and execution/analysis configuration; Appendix C records the six successful C0 audits.

We report strict task success, adapter-specific collateral evidence, signed token response, and target-start-to-termination completion time. Local equivalence and control-flow metrics characterize propagation. AppWorld/LoCoMo new-cohort intervals use task-cluster resampling; SWE uses a 2,000-replicate span bootstrap over the measured spans, not task-population inference. Appendix E defines uncertainty scope.

6 Empirical Findings and Evaluation

E1–E3 answer RQ1–RQ2; E4 characterizes the conditional structure and stability addressed by RQ3.

Table 2: Cohort-level treatment response. Strict success and local equivalence are treatment-arm rates. ΔQ is $T_{\text{span}} - B$ in percentage points; positive net-token and time values favor treatment. Follow-18* is an outcome-selected stability follow-up. SWE-bench uses the final 134-point intervention cohort.

Cohort	T/B strict (%)	T local eq. (%)	ΔQ	Net tok. (k)	Time save (s)
Sim-48	57.9 / 55.8	35.4	+2.1	−47.44	+35.79
Follow-18*	73.0 / 59.6	27.8	+13.3	+62.53	+67.37
Tau2-44	64.8 / 64.1	13.6	+0.7	−3.03	+9.65
Sim-84	53.2 / 58.2	28.7	−5.0	+28.54	+27.05
LoCoMo-257	79.5 / 90.1	41.4	−10.6	+2.19	+0.96
SWE-bench	99.5 / 99.6	9.0	−0.1	−12.61	−2.98

6.1 Why the Native Trace Is Not the Counterfactual Baseline

E1 / RQ1. A native trace supplies only one baseline realization. Regenerating the target and fixing its recorded output produce different reference distributions: in the selected AppWorld follow-up, $B - I = -17.0$ percentage points while $T_{\text{span}} - B = +13.3$ points. Repeated B runs contain both successes and failures at 1/48, 6/18, 11/44, 23/84, 43/257, and 3/134 checkpoints in the six cohorts, directly exposing ordinary execution variability. $B - I$ is a resampling contrast, not a variance estimate. We therefore interpret substitution against repeated matched B runs. Appendix G reports all contrasts and checkpoint counts.

6.2 Local Agreement Does Not Determine Final Success

E2 / RQ2. Local agreement and final task quality can diverge. Figure 3 shows this separation across the six cohorts: SWE-bench treatment reproduces the original target command in only 9.0% of runs while 99.5% pass the final evaluator. AppWorld cases also show locally equivalent outputs followed by final failure (Appendix J). Local agreement therefore does not determine workflow quality.

The continuation changes as well: SWE-bench command overlap, branching, observation rhythm, and downstream model calls shift after substitution. Appendix G, Table 18, reports these diagnostics and exact outcome counts. Collateral evidence remains adapter-specific: SWE uses repository/trajectory diagnostics alongside official task success, not AppWorld's transactional-state predicate.

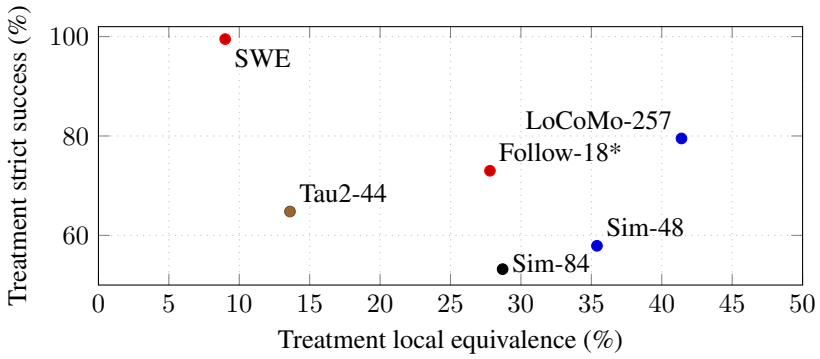
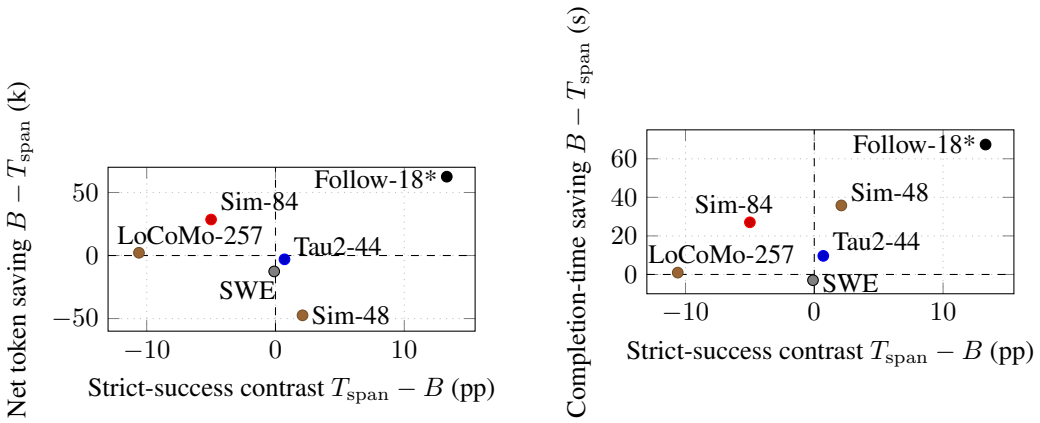


Figure 3: Local agreement is not a proxy for final workflow quality. The most extreme case is SWE-bench: only 9.0% of treatment runs reproduce the original target command, while 99.5% pass the final SWE-bench evaluator. Points are descriptive cohort aggregates; Follow-18* is outcome-selected.

6.3 A Local Substitution Induces a Signed Workflow Response

E3 / RQ2. Direct target savings do not determine workflow savings. All six cohorts save target tokens, but continuation response changes the net sign. LoCoMo uses 2.19k fewer mean tokens and finishes 0.96 s sooner while losing 10.6 percentage points of strict success. SWE-bench preserves near-saturated success, yet downstream continuation reverses its target-token saving and erases the target-latency gain. Figure 4 summarizes the six quality/resource responses; Appendix G gives target, continuation, and completion-time decompositions with uncertainty.



(a) Token response

(b) Time response

Figure 4: Quality and resource responses across the six measured cohorts. Positive horizontal values favor treatment quality; positive vertical values indicate resource saving. The quadrants show that resource reduction and quality preservation are distinct response coordinates. Follow-18* is the outcome-selected stability follow-up; SWE-bench uses the final 134-point intervention cohort.

6.4 Conditional Response Atlas

E4 / RQ3. Coarse labels do not determine response across populations. Role-conditioned AppWorld behavior differs by agent; all measured LoCoMo spans are read-only despite a negative aggregate quality contrast; and SWE-bench shows near-saturated final success despite extensive local command changes. These descriptors do not consistently identify workflow-response regimes.

Decision-time progress is one concrete example. In SWE-bench, the low/middle/high prefix-message groups have mean net token savings of -78.0k, $+36.6$k, and $+3.1$k, while treatment strict success remains 0.98, 1.00, and 1.00. The corresponding normalized final-position analysis shows the same broad token pattern, but its denominator is retrospective. We therefore treat prefix progress as decision-time evidence and normalized position as an explanatory diagnostic.

The atlas records favorable, unfavorable, redundant, non-discriminative, and unsupported evidence together, distinguishing pre-decision conditions from retrospective mechanisms. Table 3 gives representative entries; Appendix H, Table 19, provides the complete inventory of descriptor families, timing, populations, and response evidence.

These are descriptive extremes, not causal estimates, but they illustrate how similar local and final outcomes can coexist with opposite workflow-level resource responses.

6.5 Stability and Cross-Population Heterogeneity

The outcome-selected 18-checkpoint follow-up shows weak screening/follow-up stability, including four sign reversals; regression to the mean is a competing explanation. The disjoint AppWorld-84 cohort, LoCoMo, and SWE-bench further show that response structure changes across measured populations. Atlas entries are therefore population-conditioned evidence. Appendix I reports the full stability analysis.

Table 3: Representative entries from the conditional response atlas. Status describes the indicated response coordinate, not joint offloading eligibility; Appendix H reports the complete inventory and uncertainty scope.

Condition	Timing	Population	Measured response	Interpretation
Role / mechanism	Pre / retro	AppWorld	Response directions differ across agents.	Heterogeneous
Declared read-only	Pre	LoCoMo	$\Delta Q = -10.6$ pp while mean tokens and time decrease.	Mixed: quality-unfavorable, resource-favorable
Prefix progress	Pre	SWE-bench	Low/mid/high net saving -78.0k/$+36.6$k/$+3.1$k; treatment success $\approx .98/1/1$.	Informative in this population
Native target size	Retro	AppWorld	$\rho = .668$ versus .068 across the two agents.	Heterogeneous
Prefix tool-call count	Pre	SWE-bench	$\rho = .999$ with prefix-message count.	Redundant
Prior nonzero return code	Pre	SWE-bench	No quality discrimination across groups.	Non-discriminative

Table 4: Two interventions on the same SWE-bench task (`django-10097`) with identical local mismatch and final success but opposite workflow-level resource responses.

Intervention point	Local eq.	B/T_{span} success	Net token saving	Time saving
Early, $k = 7/32$	0/10	10/10 / 10/10	-334.16k	-223.73 s
Late, $k = 24/32$	0/10	10/10 / 10/10	$+281.73$k	$+245.21$ s

7 Related Work

Agent benchmarks and stateful evaluation. Benchmarks expose executable state, tool interaction, and intermediate trajectories across web, tool-use, memory, and software-engineering tasks (Liu et al., 2024; Zhou et al., 2024; de Chezelles et al., 2025; Ma et al., 2024; Trivedi et al., 2024; Lu et al., 2025; Yao et al., 2025; Froger et al., 2026; Jimenez et al., 2024; Maharana et al., 2024). Shared-control and safety benchmarks extend this coverage to user–agent interaction, risky tools, prompt injection, and harmful requests (Barres et al., 2026; Ruan et al., 2024; Debenedetti et al., 2024; Andriushchenko et al., 2025). They supply tasks and evaluators. IMPACT takes one physical LM invocation as the intervention unit and repeatedly restores its pre-state to measure substitution response against baseline execution variability (Bjarnason et al., 2026). Our Tau2-agent cohort uses AppWorld, not the τ^2-bench environment.

Efficient agentic inference and small models. Smaller-model deployment motivates selective substitution (Belcak et al., 2025). Routing systems assign requests, reasoning steps, or workflow components to cheaper models (Chen et al., 2024; Ong et al., 2025; Lee et al., 2026; Liu et al., 2026; Feng et al., 2026; Chaudhry et al., 2026); context-reduction methods reduce accumulated input (Kang et al., 2026; Xiao et al., 2026). IMPACT supplies workflow-level evidence for such allocation decisions by isolating one substituted invocation and following its live continuation. Its measurement unit separates target efficiency from the quality and resource response of subsequent execution.

Trajectory and intervention evaluation. ReAct and Reflexion connect actions with later observations and feedback (Yao et al., 2023; Shinn et al., 2023). Trajectory analyses expose intermediate structure, failures, and downstream effects (AlSayyad et al., 2026; Gao et al., 2026; Kim et al., 2026; Bouzenia & Pradel, 2025; Cemri et al., 2025; Zhu et al., 2026). Fault-injection studies such as MAS-FIRE perturb agent or inter-agent behavior to characterize propagation and recovery (Jia et al., 2026; Mazhar et al., 2026), whereas IMPACT intervenes on the deployed model at one native LM invocation and estimates its response relative to repeated matched baseline execution. It combines matched baseline resampling, single-span treatment, and joint local/downstream/final/resource measurement (Rubin, 1974; Athey & Imbens, 2016); the atlas organizes observed conditional associations without attributing them to causal effects of the descriptors.

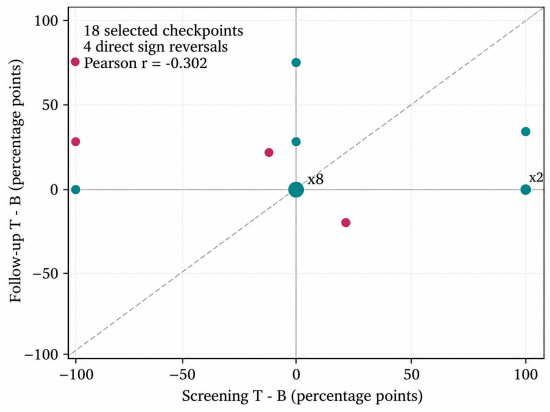


Figure 5: Strict-success contrast ($T_{\text{span}} - B$) at the same 18 checkpoints in five-repeat screening and fifteen-repeat selected follow-up. Four sign reversals and a wide spread show limited repeated-checkpoint stability; selection on screening extremes limits population interpretation.

Closest intervention studies. The Replay Gap uses live branching rollouts and matched same-model controls to show that static replay can mischaracterize model switching in SWE-bench agents (Gonuguntla, 2026). IMPACT addresses a broader measurement question: given one physical LM invocation, how should the workflow response to its substitution be estimated and characterized? Its four-condition design separately audits replay fidelity, measures fixed-output continuation variation, resamples the baseline target, and evaluates the candidate substitution. The resulting estimand is paired with local, propagation, final-quality, collateral-risk, token, and time responses, and is organized by population and pre-decision conditions.

8 Discussion and Conclusion

IMPACT separates local agreement, final quality, collateral effects, and resource response when measuring single-span model substitution in stateful agents. Repeated matched baselines show that a native trace is only one realization of the baseline workflow. Across the measured cohorts, local mismatch can coexist with final success, and downstream continuation can amplify or erase target-level savings. AppWorld, LoCoMo, and SWE-bench exhibit different response regimes, motivating population-conditioned workflow measurement.

The evidence covers sampled AppWorld traces, one LoCoMo conversation with 95 QA instances, and four SWE-bench Django instances. The selected AppWorld follow-up is outcome-selected; SWE uncertainty characterizes measured spans, not task-population generalization. The reported responses instantiate one baseline/candidate deployment pair, consistent with the estimand in Eq. (1)–(2), which conditions response on the agent and deployments H, D_L, D_m; IMPACT does not assume a model-independent offloadability profile, and additional model pairs constitute new deployment populations measured with the same protocol. The atlas organizes evidence for offloading decisions while separating pre-decision conditions from retrospective explanation. IMPACT measures one-span substitution; persistent post-fork model switching and deployment-specific hardware or monetary effects remain outside the present estimand.

AI USE STATEMENT

AI assisted with organizing research materials, checking bibliographic metadata, editing prose, assembling LaTeX, and checking compilation and layout. Measured numbers are derived from the project experimental records and analysis outputs. All AI-assisted work was reviewed by the authors, who remain responsible for the scientific claims, citations, equations, results, code, and final manuscript.

REPRODUCIBILITY STATEMENT

Appendices A–E specify workload/agent integration, raw span records, replay controls, intervention arms, response metrics, and statistical procedures. The released analysis snapshot contains the final six cohorts reported in Section 5, including the final 134-point SWE-bench cohort. Reproduction requires the corresponding checkpoints, model/agent revisions, evaluator versions, C0 audit records, exact resampling settings, and run-linked resource records. Appendix F reports the final experimental inventory and validity provenance; Appendices G–J provide the supporting results and conditional analyses.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=uqv8-U4lKBe.

Adam AlSayyad, Kelvin Yuxiang Huang, and Richik Pal. AgentTrace: A structured logging framework for agent system observability, 2026. URL https://arxiv.org/abs/2602.10133. arXiv:2602.10133v1.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, J Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. AgentHarm: A benchmark for measuring harmfulness of LLM agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AC5n7xHuR1.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016. doi: 10.1073/pnas.1510489113. URL https://doi.org/10.1073/pnas.1510489113.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik R Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=OC2z7iSQKa.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic AI, 2025. URL https://arxiv.org/abs/2506.02153. arXiv:2506.02153v2.

Bjarni Haukur Bjarnason, André Silva, and Martin Monperrus. On randomness in agentic evals. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/10016323.

Islem Bouzenia and Michael Pradel. Understanding software engineering agents: A study of thought-action-result trajectories. In *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 2846–2857, 2025. doi: 10.1109/ASE63991.2025.00234. URL https://ieeexplore.ieee.org/document/11334353.

A. Colin Cameron and Douglas L. Miller. A practitioner's guide to cluster-robust inference. *Journal of Human Resources*, 50(2):317–372, 2015. doi: 10.3368/jhr.50.2.317. URL https://doi.org/10.3368/jhr.50.2.317.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(70):2079–2107, 2010. URL http://jmlr.org/papers/v11/cawley10a.html.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*, 2025. URL https://openreview.net/forum?id=MqBzKkb8eK.

Gohar Irfan Chaudhry, Esha Choukse, Haoran Qiu, Íñigo Goiri, Rodrigo Fonseca, Adam Belay, and Ricardo Bianchini. Murakkab: Resource-efficient agentic workflow orchestration in cloud platforms. In *20th USENIX Symposium on Operating Systems Design and Implementation (OSDI 26)*, pp. 567–587, Seattle, WA, 2026. USENIX Association. ISBN 978-1-939133-55-7. URL https://www.usenix.org/conference/osdi26/presentation/chaudhry.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=cSimKw5p6R.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/biomet/26.4.404. URL https://doi.org/10.1093/biomet/26.4.404.

Thibault Le Sellier de Chezelles, Maxime Gasse, Alexandre Lacoste, Massimo Caccia, Alexandre Drouin, Léo Boisvert, Megh Thakkar, Tom Marty, Rim Assouel, Sahar Omidi Shayegan, Lawrence Keunho Jang, Xing Han Lù, Ori Yoran, Dehan Kong, Frank F. Xu, Siva Reddy, Graham Neubig, Quentin Cappart, Russ Salakhutdinov, and Nicolas Chapados. The BrowserGym ecosystem for web agent research. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=5298fKGmv3. Expert Certification.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, pp. 82895–82920, 2024. URL https://papers.neurips.cc/paper_files/paper/2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The Hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1383–1392. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1128. URL https://aclanthology.org/P18-1128/.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015. doi: 10.1126/science.aaa9375. URL https://doi.org/10.1126/science.aaa9375.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1), 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

Tao Feng, Haozhen Zhang, Zijie Lei, Peixuan Han, and Jiaxuan You. GraphPlanner: Graph memory-augmented agentic routing for multi-agent LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZdGB7MNQDT.

C. A. Field and A. H. Welsh. Bootstrapping clustered data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(3):369–390, 2007. doi: 10.1111/j.1467-9868.2007.00593.x. URL https://doi.org/10.1111/j.1467-9868.2007.00593.x.

Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, Kunal Malkan, Dheeraj Mekala, Pierre Menard, Gerard Moreno-Torres Bertran, Ulyana Piterbarg, Mikhail Plekhanov, Mathieu Rita, Andrey Rusakov, Vladislav Vorotilov, Mengjue Wang, Ian Yu, Amine Benhalloum, Grégoire Mialon, and Thomas Scialom. Gaia2: Benchmarking LLM agents on dynamic and asynchronous environments. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9gw03JpKK4.

Tianci Gao, Haoxuan Li, Jian He Li, Tianxiang Zhao, Shi Runze, Weiran Wang, Zezhao Wu, and Lu Mi. Graph of trace: Visualizing execution traces of scientific agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 297–306, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-demo.29. URL https://aclanthology.org/2026.acl-demo.29/.

Ashritha Gonuguntla. The replay gap: Static evaluation of model switching in LLM agents scores the wrong world. In *COLM 2026 Workshop on Efficient Reasoning*, 2026. URL https://openreview.net/forum?id=8gqqiNrzyA.

Jin Jia, Zhiling Deng, Zhuangbin Chen, Yingqi Wang, and Zibin Zheng. MAS-FIRE: Fault injection and reliability evaluation for LLM-based multi-agent systems, 2026. URL https://arxiv.org/abs/2602.19843.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. ACON: Optimizing context compression for long-horizon LLM agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=5EmOOLtH5P.

Wonjoong Kim, Sangwu Park, Yeonjun In, Sein Kim, Dongha Lee, and Chanyoung Park. Beyond the final answer: Evaluating the reasoning trajectories of tool-augmented agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=PRfEvn1UHp.

Daniël Lakens. Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 8(4):355–362, 2017. doi: 10.1177/1948550617697177. URL https://doi.org/10.1177/1948550617697177.

Sangmook Lee, Dohyung Kim, Hyukhun Koh, Nakyeong Yang, and Kyomin Jung. Confidence-guided stepwise model routing for cost-efficient reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(37):31483–31491, 2026. doi: 10.1609/aaai.v40i37.40413. URL https://ojs.aaai.org/index.php/AAAI/article/view/40413.

Shiyi Liu, Haiying Shen, Shuai Che, Mahdi Ghandi, and Mingqin Li. AIMS: Cost-efficient LLM-based agent deployment in hybrid cloud-edge environments. In *Proceedings of the 21st European Conference on Computer Systems*, pp. 1862–1878. ACM, 2026. doi: 10.1145/3767295.3803622. URL https://dl.acm.org/doi/10.1145/3767295.3803622.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zAdUB0aCTQ.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025. doi: 10.186

53/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An analytical evaluation board of multi-turn LLM agents. In *NeurIPS*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/877b40688e330a0e2a3fc24084208dfa-Abstract-Datasets_and_Benchmarks_Track.html.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Anna Mazhar, Huzaifa Suri, and Sainyam Galhotra. Trace-level analysis of information contamination in multi-agent systems. In *ACM Conference on AI and Society (CAIS)*, 2026. URL https://www.caisconf.org/program/2026/papers/trace-level-analysis-of-information-contamination-in-multi-agent-systems/.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Gilda Piaggio, Diana R. Elbourne, Stuart J. Pocock, Stephen J. W. Evans, Douglas G. Altman, and CONSORT Group. Reporting of noninferiority and equivalence randomized trials: Extension of the CONSORT 2010 statement. *JAMA*, 308(24):2594–2604, 2012. doi: 10.1001/jama.2012.878002. URL https://doi.org/10.1001/jama.2012.87802.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of LM agents with an LM-emulated sandbox. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GEcwtMk1uA.

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974. doi: 10.1037/h0037350. URL https://doi.org/10.1037/h0037350.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=vAElhFcKW6.

SWE-agent. mini-SWE-agent. Software artifact, 2025. URL https://github.com/SWE-agent/mini-swe-agent. Official mini-SWE-agent implementation; accessed September 2026.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16017–16042, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Yuan-An Xiao, Pengfei Gao, Chao Peng, and Yingfei Xiong. Reducing cost of LLM agents with trajectory reduction. *Proceedings of the ACM on Software Engineering*, 3(FSE):1241–1263, 2026. doi: 10.1145/3797084. URL https://dl.acm.org/doi/10.1145/3797084.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=mXpq6ut8J3.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. τ-Bench: A benchmark for tool-agent-user interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

Kunlun Zhu, Muxin Tian, Bingxuan Li, Zijia Liu, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, Xiaoteng Ma, Xiaodong Yu, Gowtham Ramesh, Jialian Wu, Xiangru Tang, Zicheng Liu, Heng Ji, Pan Lu, James Zou, and Jiaxuan You. AgentDebug: Where LLM agents fail and how they can learn from failures. In *Workshop on Failure Modes of Agentic AI at ICML 2026*, 2026. URL https://openreview.net/forum?id=VIO4WbZuVW.

A Workload and Agent Integration

IMPACT uses a workload adapter and an agent adapter so that trace collection is independent of any one benchmark. The workload adapter provides the task/state object, checkpoint/restore operations, action interface, side-effect semantics, and evaluator. The agent adapter identifies physical LM invocations, tool/action events, and continuation points. Adapter revisions and evaluator revisions are part of the experiment manifest.

Table 5: Workload-adapter contract used by the three measured workload families. The contract fixes what must be exposed to Phase 1; analytical span labels are added only in Phase 3.

Field	AppWorld	LoCoMo	SWE-bench
Persistent state	app/database state	conversation + memory state	repository/worktree
Action interface	API/tool calls	memory search + QA actions	shell/git/test commands
Checkpoint boundary	agent context + app state	graph context + memory state	agent context + repository state
Strict evaluator	AppWorld final/state evaluator	LoCoMo QA evaluator	official SWE-bench pass
Local equivalence	structured action/tool contract	tool-call or answer contract	target command multiset
Collateral/state evidence	declared app-state predicates	read-only measured QA scope	repository/trajectory diagnostics
Time boundary	target start $\rightarrow$ termination	target start $\rightarrow$ termination	target start $\rightarrow$ termination

AppWorld. AppWorld supplies simulated application/database state, API actions, and state-based task evaluation (Trivedi et al., 2024). The measured populations include the 48-checkpoint Simplified screen, its selected 18-checkpoint follow-up, a disjoint re-collected 84-checkpoint Simplified screen, and the 44-checkpoint Tau2 cohort. These are sampled intervention populations, not estimates of a workload-wide replaceable-span fraction.

LoCoMo. LoCoMo supplies long-range conversational-memory questions (Maharana et al., 2024). The measured adapter uses a LangGraph ReAct loop with LangMem-backed memory access. The LoCoMo cohort contains 257 QA-stage physical invocations over 95 questions from conversation 26: 162 tool-argument-generation calls and 95 final-synthesis calls. The measured spans are read-only with respect to external application state, which makes the workload informative for testing whether a coarse read-only label predicts quality safety.

SWE-bench and mini-SWE-agent. SWE-bench supplies repository-changing software-engineering tasks (Jimenez et al., 2024). We use the official mini-SWE-agent implementation maintained by the SWE-agent project (Yang et al., 2024; SWE-agent, 2025). Its official repository asks users to cite the SWE-agent NeurIPS 2024 paper for academic attribution (Yang et al., 2024). The SWE-bench cohort covers four Django instances (10097, 10880, 10914, and 13933). A physical span is the k-th LM invocation; shell/git/Python/test commands form the action interface, and official SWE-bench pass is the strict task outcome. The cohort contains 134 intervention points, with 10 repetitions per arm.

B Trace Collection and Raw Span Records

Phase 1 records execution before applying the analytical schema. The harness logs LM calls, tool/action events, observations, state/checkpoint identifiers, timing, and evaluator provenance. A physical span is one LM invocation, not an entire logical node and not a memory/tool operation by itself.

Table 6: Phase-1 raw span record. These fields support restoration and audit without requiring the Phase-3 analytical schema.

Field family	Recorded purpose
Identity	workload/agent/task/trace/span/run identifiers and physical invocation index
Target request	exact model request, prompt/context hash, parser or output contract, deployment identity
Target output	generated/injected output and executed target tool/action events
Environment	checkpoint/state handle, restoration status/hash, workload-side action results
Continuation	live suffix linkage, downstream LM/tool events, termination reason
Evaluation	evaluator revision/source, final outcome, workload-specific state/risk evidence
Resources	target/downstream token counts, target generation latency, target-start-to-termination wall time
Provenance	native trace path or source record, arm, repetition/block identity, experiment revision

Table 7: Examples of the same raw-span abstraction across workloads. Values are drawn from synchronized experiment records; the final row illustrates the later Phase-3 mapping rather than a Phase-1 requirement.

	AppWorld	LoCoMo	SWE-bench
Example task	`23cf851_3`	`conv-26:q088`	`django__django-10097`
Physical target	invocation 0	invocation 1	k-th LM invocation
Restorable state	app + agent context	graph + memory context	repository + agent context
Action record	API/tool execution	memory/QA action	shell/git/test command
Evaluator	AppWorld final evaluator	answer-match evaluator	official SWE-bench evaluator
Phase-3 example	initial/tool selection	execution/memory operation	progress/command-flow descriptors

Raw span record. Each raw span stores the exact model request and output contract, generated output, tool calls/results, pre-state restoration handle, target start time, end/termination time, agent/workload identity, deployment identity, and run linkage. The continuation point identifies where live execution resumes. Raw records keep source/runtime identity even when later analysis groups several spans under one role or subtask.

No Phase-1 semantic labeling assumption. Node pattern, control role, subtask, mechanism tags, verifier class, first-use status, propagation class, and offloadability are not required to collect the trace. They are derived or attached in Phase 3. This prevents the collection layer from baking in one workload's control vocabulary.

C Replayability and Checkpointing

An intervention point is replay-eligible when its exact target request, parser/schema, pre-state, and live continuation can be reconstructed and evaluated. Successful native execution supplies a candidate point but does not certify replay fidelity.

Checkpoint contents. The checkpoint includes workload state, agent-visible conversation/context, tool/environment state needed for continuation, request/schema information, model/deployment identity, and evaluator provenance. Hashes or normalized fingerprints are retained where the adapter supports them. Ephemeral fields may be normalized only when the fidelity predicate declares them non-semantic.

$C0$: replay-fidelity audit. $C0$ restores the checkpoint, injects the recorded target output, executes the declared action/continuation, and checks the adapter-specific request/state/action/evaluator predicates. A C0 pass certifies those declared predicates without assuming complete environmental determinism. All six measured cohorts passed their declared pre-run C0 audit. Table 8 is the cohort-level record of audit provenance and outcomes; audited-source populations and measured intervention populations are distinguished.

Table 8: Final C0 audit provenance and outcomes. All measured cohorts passed the adapter-specific frozen request/state/action/evaluator predicates. Source-audit counts and measured intervention points describe different populations; numerical audit totals are shown where recorded.

Population	C0 audit result	Measured points	Audit provenance
Simplified source	53/53 passed	48	screen sampled from the audited source
Selected follow-up	All passed	18	inherited audited checkpoints from the screen
Tau2 source	52/52 passed	44	frozen sampling manifest from the audited source
Simplified expanded	All passed	84	separate audit of the disjoint re-collected batch
LoCoMo source	577/577 passed	257	sampled QA-stage invocations over 95 questions
SWE-bench	All passed	134	final cohort over four Django instances

D Controlled Intervention Protocol

Figure 2 (main text) summarizes the four-condition protocol; this appendix specifies each arm in detail. All live arms restore independent copies of the same eligible pre-span state and fix the request, parser/schema, environment configuration, baseline continuation policy, and budgets.

I: recorded-output continuation. I injects the original target output and then executes a live baseline continuation. It diagnoses suffix variability conditional on a fixed native target output.

Table 9: Operational single-span measurement procedure. Invalid behavior after a valid restoration is retained as an outcome.

Step	Procedure
1	Select a replay-eligible physical LM invocation and freeze its exact pre-span checkpoint, target request/contract, baseline deployment, evaluator, and budgets.
2	Run $C0$: restore the checkpoint, inject the recorded target output, execute the declared continuation, and evaluate adapter-specific fidelity predicates.
3	For each repetition of I, restore a fresh checkpoint copy, inject the recorded target output, and execute a live baseline-model continuation.
4	For each repetition of B, restore a fresh copy, regenerate the target with the baseline model, and execute the same baseline-model continuation policy.
5	For each repetition of T_{span}, restore a fresh copy, generate only the target with the candidate model, then return every downstream LM invocation to the baseline deployment.
6	Retain parse errors, invalid tool calls, retries, exceptions, failed final evaluations, and all consumed tokens/time after valid restoration; do not post-select successful continuations.
7	Link target, downstream, final-state/evaluator, token, and timing records by run/checkpoint identity; compute $T_{\text{span}} - B$ as the primary substitution contrast and use I only as a continuation-variation diagnostic.
8	Aggregate with the declared checkpoint/task dependence structure; do not treat within-checkpoint repetitions as independent tasks.

B: baseline-model resampling. B regenerates the target using the baseline model and continues live with the baseline policy. Repeated B runs supply the decision-relevant reference distribution for ordinary model and workflow variability.

T_{span}: single-span substitution. T_{span} changes only the target deployment; every later LM call returns to the baseline deployment and responds to the resulting live state. The primary substitution contrast is $T_{\text{span}} - B$. Parse errors, failed tool calls, and failed final evaluations after valid restoration remain treatment outcomes rather than post-hoc exclusions.

Blocking and replication. Arm repetitions are blocked by checkpoint and comparable arm runs are interleaved where supported. The AppWorld/LoCoMo cohorts use the repetition counts reported in Table 11; the SWE-bench cohort uses ten repetitions for each arm. Replication improves precision at sampled checkpoints but does not create new independent tasks.

E Response Metrics and Statistical Methods

IMPACT keeps response coordinates separate until a declared deployment objective combines them.

Quality, risk, and local response. Strict success requires final task success together with satisfaction of any declared prohibited-state predicate. Local equivalence is adapter-specific: structured tool/API equivalence for AppWorld/LoCoMo and equality of the target command multiset for the SWE-bench adapter. Local parse/verifier outcomes, patch equivalence, and final task success are reported separately. Collateral-state evidence is therefore evaluated separately from final-answer evaluation.

Propagation and control-flow response. Downstream traces support path overlap, branch novelty, decision/observation rhythm, retries, and first-divergence diagnostics. SWE-bench maps shell actions to command classes and computes metrics such as ToolSeqJaccard, NewBranchRate, ObserveChaining, SameToolPersistence, ParamMatchRate, and downstream LM rounds. These are retrospective response descriptors, not same-invocation routing features.

Token and time accounting. For additive cost, DirectSaving compares target use and CompensationCost compares live continuation use. Net token saving is $N_B - N_{T_{\text{span}}}$. Completion time is

measured from target start to completion/termination:

$$W_a = t_{\text{completion/termination,a}} - t_{\text{target start,a}}.$$

The elapsed-time field is `replay_meta.elapsed_ms`. The cohorts add no intervention-specific model verification, fallback, or rollback policy. Native recovery is charged to continuation. Recorded tokens and target-to-termination elapsed time retain any overhead included in those measurement boundaries. For SWE-bench the task wall-clock export uses this target-start timing boundary; target latency and downstream wall time are also retained for decomposition. Positive time saving means $W_B - W_{T_{\text{span}}} > 0$.

Uncertainty and dependence. Repeated runs at one checkpoint are not independent tasks, so test choice must follow the experimental unit and dependence structure (Dror et al., 2018). Bootstrap intervals follow the general resampling principle (Efron, 1979); clustered populations require resampling and inference rules that respect the cluster structure and cluster count (Field & Welsh, 2007; Cameron & Miller, 2015). The AppWorld/LoCoMo new-cohort intervals reported in the main paper use task-cluster resampling as documented by their analysis export. SWE control-flow and subgroup intervals use a 2,000-replicate span bootstrap; because the cohort contains only four Django instances, these intervals characterize the measured span set and are not presented as task-level generalization. Prospective offloadability additionally requires pre-specified non-inferiority, risk, and utility bounds rather than a nonsignificant difference alone (Piaggio et al., 2012; Lakens, 2017). Zero observed collateral events likewise do not establish zero deployment risk; exact independent-binomial bounds are only a reference point when replay observations are dependent (Clopper & Pearson, 1934).

Confirmation and selection. Screening can discover profiles, but confirmation fixes hypotheses, features, margins, and analysis before measurement; task-disjoint transfer additionally withholds test tasks from feature and threshold selection. These are requirements for decision evidence, not a claim that the descriptive atlas validates frozen-profile transfer. Eq. (4) requires explicit non-inferiority, absolute risk, and resource bounds; a nonsignificant contrast alone is insufficient.

F Measurement Validity and Experimental Inventory

Serving, recovery, and variability controls. The baseline endpoint is Qwen3.8-27B and the candidate endpoint is Qwen/Qwen3.5-9B. All six cohorts use native recovery only. At each checkpoint, the live arms share the restored state, request, parser, baseline continuation policy, and budgets. I measures continuation variability with the recorded target output fixed; B additionally regenerates that output. Their contrast is interpreted within this matched execution configuration. Appendices A–C define the adapter-specific checkpoint and evaluator contracts.

Table 10: Execution and analysis configuration. Analysis-resampling seeds are distinct from model-generation seeds.

Field / scope	Configuration
Endpoints; all cohorts	Baseline Qwen3.8-27B; candidate Qwen/Qwen3.5-9B
Recovery; all cohorts	Native recovery only; no added model verification/fallback/rollback policy
Checkpoint controls	Independent copies of the same pre-invocation state; fixed target request, parser, continuation policy, and budgets
Evaluator semantics	AppWorld task/state predicates; LoCoMo QA evaluation; SWE-bench official pass plus repository/trajectory diagnostics (Appendix A)
C0 provenance	All six cohorts passed their declared replay-fidelity audit (Appendix C)
Replication	Cohort-specific repeats per arm in Table 11
Expanded AppWorld / LoCoMo resampling	Task-cluster bootstrap, 10,000 replicates, analysis seed 20260924
SWE resampling	Span bootstrap, 2,000 replicates; intervals characterize the measured spans across four Django instances

Table 11: Final intervention inventory used in all reported analyses. The selected follow-up reuses outcome-selected Simplified checkpoints.

Cohort	Points (tasks/questions)	Repeats/arm	Analyzed live runs
AppWorld Simplified screen	48 (17)	5	720
AppWorld selected follow-up	18 (12)	15	810
AppWorld Tau2	44 (4)	10	1,320
AppWorld Simplified expanded	84 (12)	10	2,520
LoCoMo LangGraph	257 (95)	10	7,710
SWE-bench mini-SWE-agent	134 (4)	10	4,020
Total analyzed continuations	–	–	17,100

Repeated B runs contain both success and failure at 1/48, 6/18, 11/44, 23/84, 43/257, and 3/134 sampled points, respectively. This is the empirical reason to retain repeated B rather than compare treatment only with the native trace. The selected 18-point follow-up is a stability study rather than an independent population. All reported SWE-bench summaries use the same final 134-point intervention manifest.

G Full Intervention Results

Measured resampling contrast. Strict-success contrasts $(B - I, T_{\text{span}} - B, T_{\text{span}} - I)$ are $(-8.8, +2.1, -6.7)$, $(-17.0, +13.3, -3.7)$, $(-8.4, +0.7, -7.7)$, $(-0.4, -5.0, -5.4)$, $(-3.9, -10.6, -14.5)$, and $(+0.4, -0.1, +0.3)$ percentage points for the Simplified screen, selected follow-up, Tau2 screen, expanded Simplified screen, LoCoMo cohort, and SWE-bench cohort, respectively. Thus I and B can differ materially, but the magnitude is population-dependent; the primary substitution estimand remains matched $T_{\text{span}} - B$.

Repeated B runs include both successes and failures at 1/48, 6/18, 11/44, 23/84, 43/257, and 3/134 checkpoints across the six cohorts. These within-checkpoint changes directly demonstrate baseline variability. $B-I$ is a resampling contrast, not a variance statistic, and I is a fixed-output continuation diagnostic rather than an oracle or cost baseline. A substitution must therefore be interpreted against the matched B distribution, not a single recorded trajectory.

Tables 12 and 13 report cohort-level strict success and resource response for all six analyzed populations. Table 14 retains the task-cluster intervals for the re-collected AppWorld and LoCoMo cohorts, while Table 15 records the SWE aggregate with its separately defined span-bootstrap uncertainty.

Table 12: Strict success: successes/runs (percent); Δ is $T_{\text{span}} - B$ in percentage points.

Cohort	I	B	T_{span}	Δ
Simplified, 48×5	155/240 (64.6)	134/240 (55.8)	139/240 (57.9)	$+2.1$
Follow-up, 18×15	207/270 (76.7)	161/270 (59.6)	197/270 (73.0)	$+13.3$
Tau2, 44×10	319/440 (72.5)	282/440 (64.1)	285/440 (64.8)	$+0.7$
Simplified, 84×10	492/840 (58.6)	489/840 (58.2)	447/840 (53.2)	-5.0
LoCoMo, 257×10	2416/2570 (94.0)	2316/2570 (90.1)	2043/2570 (79.5)	-10.6
SWE-bench, 134×10	1329/1340 (99.2)	1334/1340 (99.6)	1333/1340 (99.5)	-0.1

Table 13: Mean resource response per intervention. Token columns are thousands; time reduction is $(W_B - W_T)/W_B$.

Cohort	Direct	Compensation	Net tokens	Time red.
Simplified screen	$+1.06$	$+48.50$	-47.44	25.0%
Selected follow-up	$+1.14$	-61.39	$+62.53$	25.3%
Tau2 screen	$+0.13$	$+3.16$	-3.03	4.3%
Simplified expanded	$+0.86$	-27.69	$+28.54$	10.6%
LoCoMo cohort	$+0.26$	-1.93	$+2.19$	8.3%
SWE-bench	$+0.61$	$+13.22$	-12.61	-1.2%

Table 14: New-cohort checkpoint-level responses. Brackets are 95% task-cluster bootstrap intervals. Net tokens are $B - T_{\text{span}}$; $\Delta W = W_{T_{\text{span}}} - W_B$.

Cohort	ΔQ (pp)	Net tokens (k)	ΔW (s)
Simplified expanded	-5.0 $[-13.9, +4.6]$	$+28.54$ $[-36.68, +101.17]$	-27.05 $[-86.98, +32.15]$
LoCoMo cohort	-10.6 $[-14.9, -6.6]$	$+2.19$ $[-0.07, +4.53]$	-0.96 $[-1.72, -0.19]$

The intervals resample tasks with replacement and retain all sampled checkpoints within each resampled task (10,000 replicates; analysis seed 20260924). They quantify dependence-aware uncertainty for these descriptive cohort contrasts; they are not the non-inferiority/risk bounds required by Eq. (4).

Table 15: SWE-bench aggregate response over the final 134 intervention points. ΔQ is $T_{\text{span}} - B$ strict-success difference. Net tokens and time saving are $B - T_{\text{span}}$. Intervals are the existing 2,000-replicate span bootstrap, so they characterize the measured span set rather than task-level generalization.

Metric	Point estimate	95% CI	Direction
ΔQ (pp)	-0.07	$[-0.52, +0.37]$	quality neutral at this resolution
Net token saving (k)	-12.61	$[-39.84, +15.05]$	no aggregate token gain
Target latency saving (s)	$+3.05$	$[-0.86, +7.01]$	target call faster
Completion-time saving (s)	-2.98	$[-24.99, +16.86]$	no aggregate time gain

Table 16: SWE-bench arm-level outcomes for the final 134 intervention points (1,340 runs/arm). Patch equivalence compares the resulting patch with the recorded native patch when available.

Metric	I	B	T_{span}
Strict success (%)	99.18	99.55	99.48
Target local equivalence (%)	100.00	60.82	8.96
Patch equivalence (%)	86.34	86.27	82.76
Target tokens	0	18.18k	17.57k
Downstream tokens	441.70k	435.97k	449.19k
Total recorded model tokens	674.67k	687.12k	699.74k
Target latency (s)	0	12.22	9.17
Downstream wall time (s)	241.38	234.88	240.95
Post-target completion time (s)	242.18	247.91	250.89

Table 17: Per-instance SWE-bench responses on the final 134-point intervention cohort. Net tokens and time saving use $B - T_{\text{span}}$; strict success and local equivalence are run-level rates.

Instance	Spans	$I/B/T$ strict (%)	T eq. (%)	Net tok. (k)	Time save (s)
Django-10097	31	96.8 / 98.1 / 97.7	9.7	-4.28	-41.71
Django-10880	37	100 / 100 / 100	10.8	-36.43	-4.85
Django-10914	40	100 / 100 / 100	12.5	-7.49	$+13.63$
Django-13933	26	99.6 / 100 / 100	0.0	$+3.47$	$+20.29$

Table 18: Representative SWE-bench downstream changes for $T_{\text{span}} - B$ over the final 134-point SWE-bench cohort. Intervals are the existing 2,000-replicate span bootstrap and are descriptive within the four measured Django instances.

Metric	$\Delta(T - B)$ [95% CI]	Interpretation
ToolSeqJaccard	-0.041 $[-0.072, -0.014]$	less baseline-path overlap
NewBranchRate	$+0.014$ $[+0.005, +0.025]$	more new exploration
BatchDecisionRate	-0.026 $[-0.042, -0.012]$	fewer batched decisions
ObserveChaining	-0.019 $[-0.031, -0.008]$	altered observation rhythm
SameToolPersistence	-0.032 $[-0.063, -0.005]$	less repeated tool-class use
ParamMatchRate	-0.068 $[-0.092, -0.042]$	changed command arguments
Downstream LLM rounds	$+0.790$ $[+0.032, +1.539]$	more continuation work

Measured quality response. Table 2 summarizes cohort-level treatment quality, local agreement, and resource response. The original AppWorld point contrasts are near or above zero, the disjoint re-collected 84-span screen is -5.0 points, LoCoMo is -10.6 points, and the final SWE-bench cohort is approximately -0.1 point. Exact success denominators and uncertainty details appear in Appendix G.

Local mismatch and final outcome. Figure 3 shows that local agreement and final success can be almost completely decoupled. In SWE-bench, treatment reproduces the original target command in only 120/1340 runs (9.0%), yet 1333/1340 treatment runs (99.5%) pass the official evaluator; 1213/1220 locally non-equivalent treatment runs still pass. The workflow is not simply replaying the same path: Table 18 shows significant changes in command-sequence overlap, new-branch exploration, observation rhythm, parameter matching, and downstream LLM rounds. The other five cohorts show the same qualitative local/final mismatch at less extreme magnitudes. For SWE-bench, the adapter uses official task success plus repository/trajectory diagnostics rather than the transactional-state collateral predicate used by AppWorld, so collateral evidence is reported separately.

Signed token and time response. Figure 4 and Table 2 show that resource and quality coordinates move independently. Direct target-token saving is positive in every cohort, but continuation compensation changes the net sign. LoCoMo has a -10.6-point quality contrast while using 2.19k fewer model tokens and finishing 0.96 s sooner on average. SWE-bench exhibits the complementary pattern: strict success is essentially unchanged, yet the 0.61k target-token saving is outweighed by 13.22k additional downstream tokens, leaving a mean net token saving of -12.61k.

The time decomposition tells the same story. On SWE-bench the candidate target call takes 9.17 s versus 12.22 s under baseline resampling, a roughly 25% reduction in the point estimate of target latency. The treatment continuation then spends 6.07 s more downstream on average, so post-target completion time is 250.89 s for treatment versus 247.91 s for baseline, i.e., a mean saving of -2.98 s. The corresponding interval spans zero. Target-level efficiency therefore does not imply workflow-level speedup. Detailed target/continuation accounting and uncertainty appear in Appendix G.

Full run-linked and control-flow rows remain in the analysis artifact; the appendix reports cohort-level summaries and selected conditional analyses.

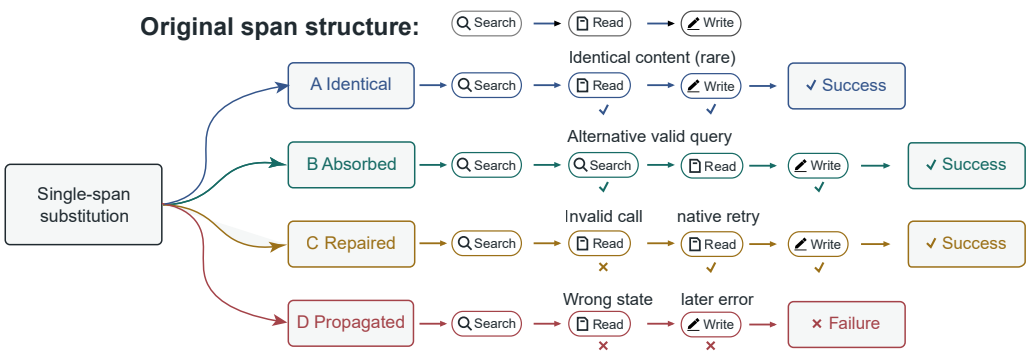


Figure 6: A local substitution can be absorbed, repaired, or propagated through the workflow. The paths are conceptual examples; final quality, collateral effects, and resource use are measured separately.

H Conditional Response Analysis

Phase 3 derives an analytical schema only after raw trace collection. For an analyzed span,

$$\phi_s = (\text{node pattern}, \text{control role}, \text{subtask}, \text{mechanism tags}, \text{verifier}, \text{context}),$$

with task, agent/architecture, and deployment conditions stored separately.

Feature timing. Pre-decision conditions may include declared node/control information known before generation, output contract, prefix-observable first use, invocation index or relative progress when the denominator is known, current context/memory state, and prefix tool history. Retrospective descriptors include realized single/multi-call shape, actual read/write effects, repair/retry, first divergence, and propagation. Outcomes include final quality, collateral state, tokens, and time. Retrospective descriptors and realized target output cannot be used as same-invocation routing features.

Table 19: Final conditional response atlas: inventory of reported descriptor families. Entries reorganize existing evidence, not new subgroup tests. Pre denotes information available before the target invocation; retro denotes retrospective information. Favorable/unfavorable describe the indicated response coordinate and its reported point estimate, not joint offloading eligibility. Figures and detailed tables retain the original uncertainty scopes.

Descriptor / condition	Timing	Population	Response	Observed evidence	Status
Role / mechanism interactions	Pre if declared; retro if realized	AppWorld	Quality / tokens	Role responses differ by agent; single-call/first-use and write/read-only first-use strata have different response directions.	Heterogeneous
Native target size	Retro	AppWorld screen / Tau2	Direct token saving	Spearman $\rho = 0.668$ versus 0.068; the size association varies across populations.	Heterogeneous
Declared read-only capability	Pre	LoCoMo	Quality	All 257 spans are read-only; aggregate strict-success contrast is -10.6 pp.	Unfavorable
Declared read-only capability	Pre	LoCoMo	Tokens / time	Aggregate means save 2.19k tokens and 0.96 s despite the quality loss; Table 14 gives intervals.	Favorable means
Execution phase	Pre if exposed	AppWorld-84 / LoCoMo	Quality / tokens	Initial, execution, and terminal strata have different signed responses (Table 22).	Heterogeneous
Low prefix progress	Pre	SWE-bench	Tokens / quality	Low prefix-message count: -78.0k net token saving; treatment strict success 0.98.	Unfavorable tokens

Table 19 continued: conditional response atlas

Descriptor / condition	Timing	Population	Response	Observed evidence	Status
Middle / high prefix progress	Pre	SWE-bench	Tokens / quality	Net token means $+36.6$k / $+3.1$k; treatment strict success 1.00 / 1.00.	Favorable means
Normalized final-trace position	Retro	SWE-bench	Tokens / time	Early/middle/late net tokens -83.8k / $+28.9$k / $+13.4$k; time intervals are wide (Table 23).	Heterogeneous
Prefix tool-call count	Pre	SWE-bench	Token response	Correlation with prefix-message count $\rho = 0.999$; does not supply independent progress evidence.	Redundant
Recent command repetition	Pre	SWE-bench	Resource response	112/134 spans have high repetition; correlation with progress $\rho = 0.74$.	Redundant
Prior nonzero return code	Pre	SWE-bench	Quality	Present in 98/134 prefixes; final quality does not discriminate the groups.	Non-discriminative
Previous-result length	Pre	SWE-bench	Quality	No monotonic quality pattern across length terciles.	Non-discriminative
Long-tutorial first-tool condition	Pre	SWE-bench	No response estimate	Support $n = 0$ in this adapter; no effect is inferred.	Unsupported

Table 20: Feature-timing discipline for conditional analysis. Pre means available before target generation; normalized final-trace position is retrospective unless the total horizon is independently known.

Descriptor	Timing	Availability and analysis use
Absolute invocation index	Pre	recorded for AppWorld, LoCoMo, and SWE-bench
Declared node/control role	Pre if exposed	used by AppWorld and LoCoMo; mini-SWE-agent has no shared role taxonomy
Prefix message/tool history	Pre	recorded across adapters; SWE uses it for progress/repetition analysis
Current memory/context state	Pre	agent context in AppWorld, memory + context in LoCoMo, repository/context in SWE-bench
Prefix progress count	Pre	prefix-observable; SWE uses `prompt_msgs` and tool-call count
Relative k/final steps	Retrospective unless horizon known	analyzed for SWE as normalized position; descriptive rather than inherently online-known
Actual read/write effect	Retrospective	mechanism descriptor, not a same-invocation selector
Repair/retry/first divergence	Retrospective	propagation descriptor across supported adapters
Local equivalence	Outcome	measured after the target output is realized
Realized target tokens/time	Outcome	measured resource response, not a pre-decision feature

Observed regimes require decision evidence. A prospective profile may use only information known before target generation. A single-call rule therefore requires contract-fixed cardinality; first-use must be computed from the prefix; declared write capability must replace realized writes; and a known phase or used-budget fraction must replace eventual-length position. Realized tokens, repair, tool outcomes, and downstream behavior cannot route the same invocation. The joint bounds in Eq. (4) remain the deployment gate.

Table 21: Audit of SWE-bench conditional descriptors on the measured SWE-bench cohort. Weak, redundant, and absent signals are retained rather than reporting only favorable strata.

Descriptor	Measured evidence	Interpretation
Prefix messages (`prompt_msgs`)	low/mid/high net saving -78.0k / $+36.6$k / $+3.1$k; T success 0.98 / 1.00 / 1.00	usable pre-decision progress signal in this workload
Relative k/final steps	early/mid/late net saving -83.8k / $+28.9$k / $+13.4$k	clean descriptive normalization; final denominator is not inherently online-known
Recent command-class repetition	112/134 spans are in the high-repetition group; correlation with progress $\rho = 0.74$	observation only; limited incremental separation
Prefix tool-call count	correlation with `prompt_msgs` $\rho = 0.999$	merge with progress rather than count as independent evidence
Prior nonzero return code	98/134 prefixes contain one, but quality does not discriminate the groups	drop as selector; many codes reflect grep/test noise
Previous-result length	no monotonic quality pattern across terciles	drop from headline profiles
Long-tutorial first-tool position	support $n = 0$	AppWorld-style condition absent in mini-SWE-agent

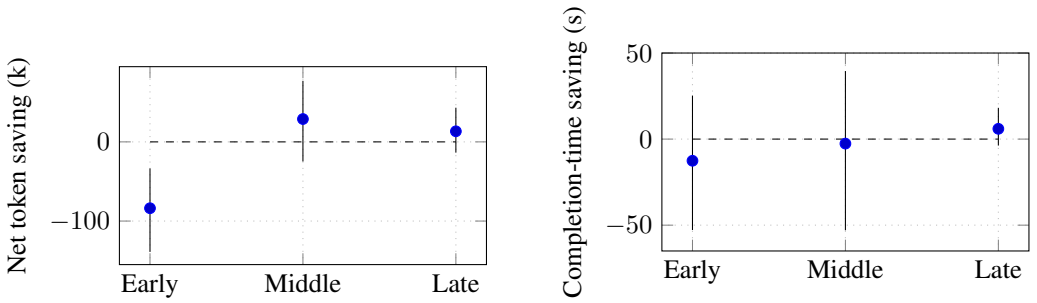
Population context. The 84-span Simplified screen comes from newly collected traces and is disjoint from the earlier Simplified screen/follow-up. LoCoMo contributes 257 QA-stage spans across 95 questions, including 162 tool-argument-generation and 95 final-synthesis spans; all are declared read-only. Cross-agent and cross-workload comparisons therefore describe recurrence or heterogeneity under different sampled populations, not controlled architecture or workload causal effects.

AppWorld descriptor evidence. Role-conditioned outcomes differ across agents. Native target-token count has checkpoint-level Spearman correlation 0.668 with DirectSaving in the 48-point Simplified screen and 0.068 in the 44-point Tau2 screen. This retrospective size association does not carry uniformly across populations. Existing interaction strata include single-call/first-use and opposing write/read-only first-use quality directions. Realized call cardinality and actual state effects are retrospective unless the contract or declared capability fixes them before generation. These interaction profiles describe response heterogeneity; subgroup support and feature semantics differ across adapters.

Table 22: Descriptive phase-conditioned responses in the two new cohorts. ΔQ is $T_{\text{span}} - B$ strict-success difference; net tokens are $B - T_{\text{span}}$.

Workload/cohort	Phase	Spans	ΔQ (pp)	Net tokens (k)
AppWorld expanded	initial	12	$+0.8$	-26.30
	execution	60	-6.5	$+52.73$
	terminal	12	-3.3	-37.54
LoCoMo	initial	95	-1.8	-0.17
	execution	67	-24.5	$+8.27$
	terminal	95	-9.7	$+0.25$

Cross-workload phase evidence. The labels are not causal treatments and do not define a transferable allowlist. In particular, LoCoMo's measured spans are all declared read-only, yet the aggregate quality contrast is negative; state-effect capability alone is insufficient for prospective selection.



(a) Token response (b) Time response

Figure 7: SWE-bench response by normalized intervention position (k/final task steps). Early substitutions incur a clear token penalty, while middle/late point estimates are positive; wall-clock response moves in the same broad direction but remains uncertain. Final task length is a retrospective normalization unless the horizon is known independently. Error bars use the existing 2,000-replicate span bootstrap and do not establish task-level generalization beyond the four measured instances.

SWE-bench progress and position. Table 23 reports a normalized position stratification of the SWE-bench traces. Because total final-trace length is not inherently known at an online decision point, relative k/final steps is descriptive unless the horizon is supplied independently. Early positions preserve quality at the aggregate resolution but incur a large token penalty; middle/late point estimates are token-positive. The pre-decision prefix-progress measure, `prompt_msgs`, shows a similar token pattern. Wall-clock response moves in the same broad direction but has wide subgroup intervals.

Table 23: SWE-bench response by relative intervention position. Net token and time values are $B - T_{\text{span}}$; intervals use the existing 2,000-replicate span bootstrap.

Position	Spans	ΔQ (pp)	Net tokens (k)	Time saving (s)
Early (k/steps $\leq 1/3$)	43	$+0.0$	-83.79 $[-138.83, -33.60]$	-12.59 $[-52.82, +25.29]$
Middle	45	-0.2	$+28.86$ $[-24.73, +77.13]$	-2.63 $[-52.96, +39.52]$
Late (k/steps $\geq 2/3$)	46	$+0.0$	$+13.36$ $[-13.57, +43.17]$	$+5.97$ $[-3.75, +18.17]$

Prefix message count gives a similar pattern: the low-progress group averages -78.0k net token saving, while middle/high progress point estimates are positive. Recent command-class repetition correlates strongly with progress and is therefore retained as an observation rather than an independent rule. The AppWorld-style long-tutorial first-tool-call condition does not occur in this SWE adapter. These negative/collinear findings are part of the atlas because descriptor usefulness is workload-dependent.

I Stability and Cross-Population Evidence

The selected 18-checkpoint AppWorld follow-up measures repeated-checkpoint stability, not prospective transfer. The five-repeat screening and fifteen-repeat follow-up contrasts have Pearson correlation -0.302, Spearman correlation -0.356, mean absolute change 47.4 percentage points, and four direct sign reversals. Selection on screening extremes makes regression to the mean a competing explanation.

Screening versus follow-up. Figure 5 plots the strict-success contrast at the same 18 checkpoints; the wide spread and four sign reversals support the population-conditioned interpretation above.

The disjoint AppWorld-84 cohort, LoCoMo, and SWE-bench provide cross-population evidence, while the selected follow-up shows limited repeated-checkpoint stability. These measurements show limited repeated-checkpoint stability and cross-population heterogeneity, supporting population-conditioned interpretation. They do not constitute a held-out predictor test: selection on noisy screening estimates and adaptive reuse limit the transfer interpretation (Cawley & Talbot, 2010; Dwork et al., 2015).

J Qualitative Cases

Two measured AppWorld follow-up checkpoints illustrate why local and final outcomes cannot be collapsed. At `fac291d_1`, invocation 5 (final synthesis), all 15 treatment outputs fail the recorded parser, local verifier, and stored equivalence checks, yet all 15 pass the final task evaluator. This can occur when task state already satisfies the evaluator or when the malformed target output is irrelevant to the final predicate; without additional state evidence it is not a claim that malformed outputs are generally safe.

At `4ec8de5_2`, invocation 0, stored equivalence passes in all 15 treatment runs while strict success is 12/15. Thus local equivalence is not sufficient for final success either. These cases supplement, rather than replace, the unconditional cohort comparisons and do not by themselves identify a causal repair mechanism.

LoCoMo: a cheaper read-only substitution can still destroy answer quality. For `locomo:conv-26:q012`, invocation 1 is an execution-phase, read-only `search_memory` argument-generation span. All ten I and all ten B continuations succeed. The treatment produces no memory-search tool call in any of its ten runs, fails local equivalence in all ten, and directly states that the requested birthday information is unavailable; all ten final treatment evaluations fail, whereas the evaluator's gold answer is *10 years ago*. Despite this quality failure, mean token saving relative to B is $+39.61$k and mean completion-time saving is $+14.44$ s. This case makes the LoCoMo aggregate lesson concrete: a read-only label and favorable resource response do not imply quality-safe substitution.

SWE-bench: early mismatch can preserve success but trigger large compensation. For `django__django-10097`, intervention $k = 7$ of 32 is in the early portion of the trajectory. Both B and T_{span} pass all ten official SWE-bench evaluations, while treatment local command equivalence is 0/10. The candidate saves only about 0.10k target tokens but induces about 334.26k additional downstream tokens, giving mean net token saving -334.16k. Post-target completion time is also 223.73 s longer under treatment on average. Final success therefore hides a large downstream detour.

SWE-bench: a later mismatch can preserve success and reduce continuation work. On the same instance, $k = 24$ of 32 is late in the trajectory. Again both B and T_{span} pass 10/10 runs and treatment local equivalence is 0/10. Here the response reverses: mean net token saving is $+281.73$k and completion-time saving is $+245.21$ s, driven primarily by about 278.73k fewer downstream tokens under treatment. These two within-task examples are descriptive extremes rather than causal evidence, but they illustrate why IMPACT models offloadability as a conditional workflow response rather than a property of command agreement alone.